

CNN News Summarization

Pramodini Karwande : Project Report R3 : 27TH Nov 2022

Abstract

We present CNN NEWSROOM, a summarization dataset of 92580 articles and highlights written by authors and editors in newsroom. Extracted from search and social media metadata between 1998 and 2022, these high-quality summaries demonstrate high diversity of summarization styles. In particular, the summaries combine abstractive and extractive strategies, borrowing words and phrases from articles at varying rates. This paper presents a method of achieving text summaries accurately using deep learning methods. We analyze the extraction strategies used in summaries against other datasets to quantify the diversity and difficulty of our new data, and train existing methods on the data to evaluate its utility and challenges. We also give a comparison of models using different dropout rates and attention commonly used for training, validation, and testing. We further analyze the performance of several typical abstractive summarization systems on common datasets. Finally, we highlight some open challenges in the abstractive summarization task and outline some future research trends. We hope that these explorations will provide researchers with new insights into DL-based abstractive summarization.

Key Words: *Text Summarization, Deep Learning, Long Short Term Memory, Sequence2Sequence model, Attention Model, Natural Language Processing, Abstractive summary, Extractive summary*

I. Problem Statement

“I don’t want to read lengthy report, just give me a summary of the results”. I have often found myself in this situation – both in college as well as my professional life. We prepare a comprehensive report and the teacher/senior professionals only has time to read the summary. Sound Familiar? This made me think about reading news paper for the people having time crunch in day-to-day life.

As we all know, newspaper carry the news of the world, provide information and general knowledge as well. Newspaper provide news about country’s economic situation, sports, games, entertainment, trade and commerce. Reading newspaper, on paper or on internet, is a part of modern life to stay updated in today’s world. This way, news data is getting generated everyday. As a result, volume of news data is expanding day to day exponentially and enormously, covering billions of archived news articles, we can say documents. These documents are also interconnected with each other with numerous sources and contains lot of information which is useful to take into account while taking decisions or understanding the situation to follow any topic such as financial crisis, inflation, market trends, political decisions, asking any opinions related to sports and many more.... Making such big decisions, analyzing world event and decisions taken based on it, and finding about the consequences of decisions becomes more difficult for the businesses, citizens, and people(who do not have access to government documents but should be making decisions for their political parties).

Another approach about the professional. Professionals in any sector depend on access to accurate and complete knowledge to make well-informed decisions. Here, professionals is directed to the lawyers, compliance directors, politicians, heads of large firms or organizations, students achieving political degree and journalists. To take any right decision or knowledge, its important to have the complete knowledge of all interconnected topics and their sources as well. Missing of any important piece from crucial information can cause to make wrong decisions or spread the incorrect information, message in the society.

Let's discuss not only for professionals, but also for normal citizens. People has to read longer news to get complete updates for the topic. People gets educated about specific topics such as bitcoins, how that can be change lives or buying hybrid/electrical cars instead of cars run on gas, investing in the housing market. But sometimes people gets busy on multiple tasks and did not get that much time to read the longer articles.

This is where the awesome concept of Text Summarization from Natural Language Processing really comes into picture. It solves the one issue to get a quick summary of a documents or news! I believe, This will help indirectly to all of the news paper readers. Every reader, having time crunch to read or not, will like to read summary first and if interested go for a detailed news.

II. Introduction

Text summarization is becoming very important and popular research topic now a days. Text summarization is a way to condense the large amounts of information into a concise form by selection of important information pieces and discarding unimportant information. Text summarization is a process of producing brief and concise summary by capturing the vital information and the comprehensive meaning. Text summarization is achieved by natural language processing techniques. Earlier algorithms like page rank fail to generate new sentences which are not in the document like humans. This is where Deep Learning helped. The use of deep learning builds an efficient and fast model for text summarization. The use of deep learning methods helps us generate summaries which can be formed with new phrases and sentences, and which are grammatically correct.

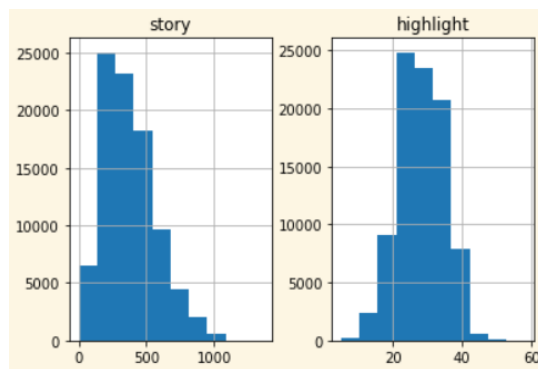
Text Summarization is broadly classified into two types:

- i. **Abstractive Summarization** : Abstractive Summarization is a more advanced method of Text Summarization technique. This approach is to identify the important sections, interpret the context and reproduce in a new way. This ensures that the core information is conveyed through the shortest text possible. Note that here, new sentences in summary are generated, not just extracted from original text.
- ii. **Extractive Summarization** : This is the traditional method developed first. This is in contrast to the Abstractive Summarization approach. The main objective is to identify the significant sentences of the text, extract from the original text and add them to the summary. Note that the summary obtained contains exact sentences from the original text.

We implement the abstractive method using the deep learning technique, Long Short Term Memory which is a type of Recurrent Neural Network Algorithms.

III. Data

Every ML project starts with data! The data used is the CNN-Dailymail dataset[1]. For this project, we are specifically using CNN stories dataset. This data contains ~90K stories. It has two features: article and highlights. The article includes the document that is to be summarized. It is the news article. Highlights are the headlines of the corresponding news which are used as summaries. Below is the distribution of word counts of story i.e. article and highlight.



IV. Methodology

The first section, Data Preprocessing, shows the methods we have used to preprocess the dataset. The second Section displays the system Architecture.

Data Preprocessing :

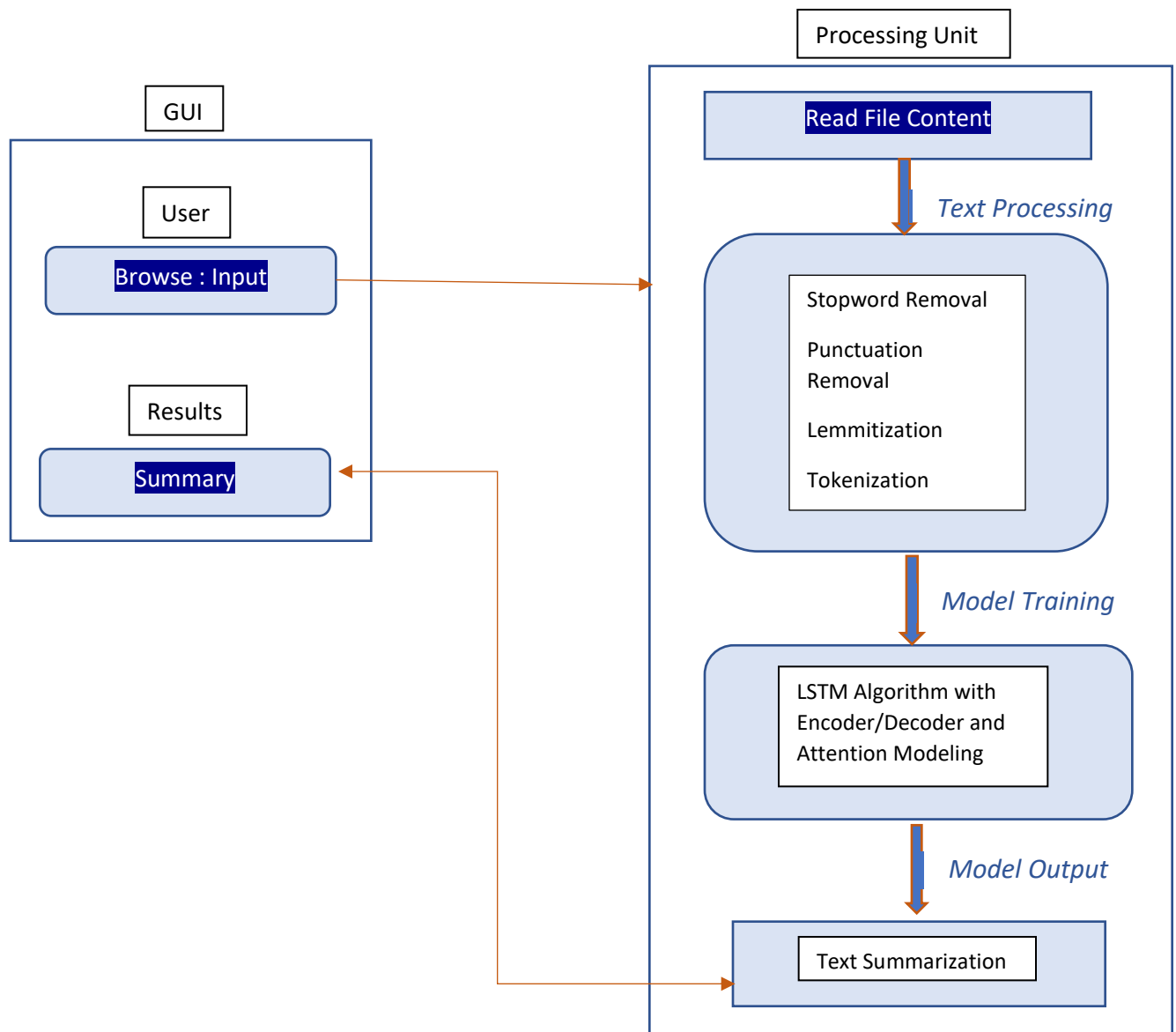
Data preprocessing is a vital step in machine learning. A model can be as good as the dataset used to train it. In NLP tasks, there are multiple different techniques to remove noise from the text to achieve better performance. But there is no “One-fits-all” methods to clean the data. It all depends on how the training data is available. Sometimes data don’t required to preprocess. In such case, available data may give almost the same results as preprocessed data. Whereas some models may work very well only with slight alterations of the texts, while others may profit from a combination of different methods. Data Preprocessing step contains removing stopwords and special characters. As we know, not all words are equally useful. When analyzing these words , we observed that these are predominately function words (like a, this, that, the) used to give a sentence structure and only convey little information. Lexical words such as nouns, verbs, adjectives and adverbs, on the other hand, contain the most information. Machine learning models do not contain this information so we cleaned up the data by adding new additional stopwords like ‘CNN’ in the default stopwords dictionary from NLTK and removing stopwords and special characters. This is not always the same case though. Some of the stopwords are required to retain based on the project task. To remove stopwords, we used NLTK English word library. Most frequently used technique is stemming and lemmatization technique. Stemming reduces words to their root form. It lacks to retain meaning of the word. Whereas Lemmatization reduces words to their

dictionary form and can be used in combination with word embeddings, as the meaning of the word is not lost. For the larger datasets, using lemmatization will consume server memory and computation time where as stemming will be more faster than lemmatization. For the stemming, we have used most popular technique as PorterStemmer and for lemmatization as WordNetLemmatizer. PorterStemmer is known for its speed and simplicity. WordNetLemmatizer is a built-in English library from WordNet. Based on the project requirement and training data, we have used lemmatization.

Basically, we perform the below preprocessing tasks for our data:

- Convert everything to lowercase
- Eliminate punctuations and special characters
- Remove stopwords
- Apply lemmatization

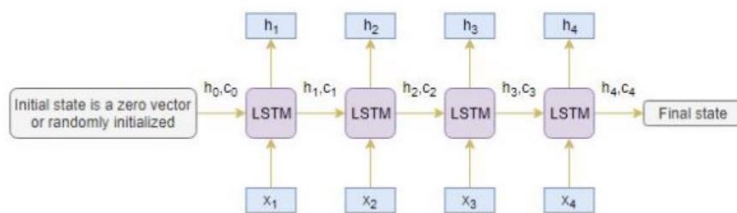
System Architecture:



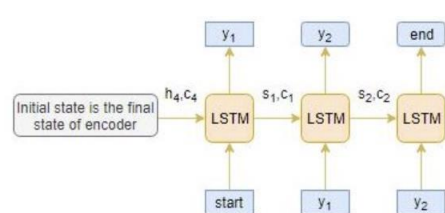
V. Model

We have used sequence to sequence model. Sequence-to-sequence learning is a training model that can convert sequences of one input domain into the sequences of another output domain. It is generally used when the input and output of a model can be of variable lengths. It is a method of encoder-decoder based machine translation that maps an input of sequence to an output of sequence with a tag and attention value. We have implemented idea is to use three LSTMs that worked together with a special token and try to predict the next state sequence from previous sequence.

- i. **Encoder** : An encoder is an LSTM network which reads the entire input sequence. At each time step, one word from the input sequence is fed into the encoder. It then processes the input at each time step and captures the context and the key information related to the input sequence. It takes each word of input(x) and generates the hidden state output (h) and the cell state which is an internal state(c). The hidden state(h_i) and cell state(c_i) of the last time step are the internal representation of the complete input sequence which will be use to initialize the decoder. Please refer Fig[1].
- ii. **Decoder** : The decoder is also an LSTM network. It reads the entire internal representation generated by the encoder one word at a time step. It then predicts the same sequence offset by one time step. The decoder is trained to predict the next word in the output sequence given the previous word based on the contextual memory stored by the LSTM architecture. Two special tokens sostok and eostok are added at the beginning and at the end of the target sequence before feeding it to the decoder. We start predicting the target sequence by passing one word at a time. The first word of output of the decoder is always token. The end of the output sequence is represented by token. Please refer Fig[2].



Fig[1]



Fig[2]

- iii. **Encoder-Decoder Inference** : LSTM prefers encoder-decoder components as they are capable of capturing long term dependencies by overcoming the issue of vanishing gradient. In the inference phase, model encode the entire input sequence and initialize decoder with internal start of encoder. In our model, we pass sostok as start of the input sequence which is first input

to the decoder. Then model run decoder with internal states and output is the probability of next word. This higher probability word is the input to the decoder in next step and update the internal states accordingly. This continues till our model hits the end token, i.e. `eostok`. Since decoder is looking for entire input sequence to work on prediction, this is not efficient for the long sequences and we are having long articles to get the summary out of it. Our news articles are long and descriptive. To summarize these articles, attention mechanism comes into play. It aims to predict a word by looking at some specific parts of the input sequence instead of using whole long sequences.

- iv. Attention Layer : A Sequence to Sequence model with an attention mechanism consists of encoder, decoder and an attention layer. Attention mechanism is used to secure individual parts of the input which are more important at that particular time. It can be implemented by taking inputs from each time steps and giving weightage to time steps. The weightage depends on the contextual importance of that particular time step. It helps pay attention to the most relevant parts of the input data sequence so that the decoder can optimally generate the next word in the output sequence.

Attention mechanism works as below to get summary out of article.

- a. Calculate Alignment Score

$$e_{ij} = \text{score}(s_i, h_j)$$

Where,

h_j = encoder outputs the hidden state for every time step j in the source sequence

s_i = decoder outputs the hidden state for every time step i in the target sequence.

e_{ij} = alignment score for the target timestep i and source time step j

- b. Get Attention Weights by normalizing alignment score using softmax function.

$$a_{ij} = \frac{e^{e_{ij}}}{\sum_{k=1}^{T_x} e^{e_{ik}}}$$

- c. Context Vector : linear sum of the products of the attention weights and hidden start of encoder.

$$C_i = \sum_{j=1}^{T_x} a_{ij} h_j$$

- d. Attended Hidden Vector : Concatenation of context vector and target hidden state of the decoder at timestep i

$$s_i = \text{concatenate}([s_i; C_i])$$

- e. Prediction of target word :

$$y_i = \text{dense}(s_i)$$

VI. Evaluation and Results

For Evaluation outcomes, we ran the models by using optimizers rmsprop and adam optimizer. Also we used without regularization and also with regularization as dropout rate. We have used metric as `sparse_categorical_crossentropy` to compare the models. The purpose of loss function is to compute the quantity that a model should seek to minimize during training. We have multilabel classification problem. For LSTM model with output shape of (None, 1000), the conventional way is to have the target outputs converted to the one-hot encoded array to match with the output shape, however, with the help of the `sparse_categorical_crossentropy` loss function, we can skip that step and keep the integers as targets.

Below Table 1 denotes model_dropout has minimum validation loss as 0.6211 which will be our best model among others. This model has been trained using optimizer as rmsprop, regularization dropout as 0.4, epoch 20 and batch size as 50.

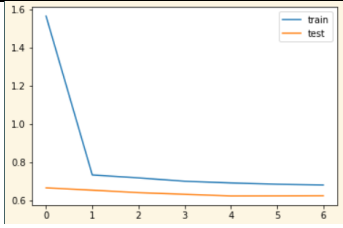
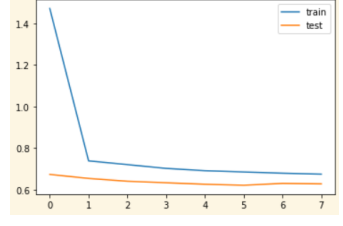
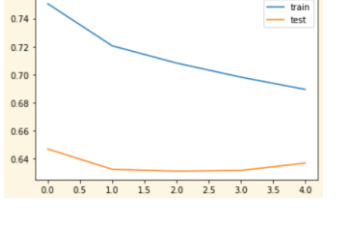
Model	Parameters	Validation loss	Train-Test Loss
model	optimizer = rmsprop, Regularization = No Epoch = 20, Batch size = 50	0.6229	
model_dropout	optimizer = rmsprop, Regularization = Dropout rate (0.4) Epoch = 20, Batch size = 50	0.6211	
model_dropout_adam	optimizer = adam, Regularization = Dropout rate (0.2) Epoch = 20, Batch size = 50	0.6309	

Table 1

Based on Encoder-Decoder, we got below results:

Story: washington accusation irs add one first opinion nonpartisan official agency went outside law conclusion tuesday official responsible managing historically important u record came congressional hearing republican also tried spotlight irs scandal include white house david head national archive record administration told house panel irs follow law failed tell agency loss email belonging former irs official lerner lerner email particularly senior manager far connected irs targeting tea party political group lerner retired last year refused testify congress constitutional right house charged contempt result lack testimony frustrated republican investigator week ago irs revealed lerner hard drive crashed 2011 destroying thousand email document agency insists missing email checking account irs employee lerner republican technical aspect crash well whether irs followed federal record keeping law better system electronic information law also require agency archive loss official record tuesday hearing house oversight committee followed contentious session monday night panel agency commissioner john questioning panel tuesday told lawmaker lerner email certainly federal record either temporary permanent said irs yet report loss hard drive email agency break law rep tim michigan republican asked im lawyer answered safely assume broke law followed follow law concluded white house official irs maintained realize full extent hard drive crash late april early may testified irs waited tell congress trying information could first hearing included testimony white house attorney jennifer oconnor worked irs may november 2013 oversight committee chairman issa oconnor white house initially said left irs agency knew hard drive crash could offer meaningful testimony agreed appear following set immediate tension california republican opened questioning accusing oconnor hostile witness demanding quick yes answer im definitely hostile answered almost voice eager cooperate ultimately issa say term hostile white house shed light lerner hard drive crash oconnor give one detailed public description yet irs initially responded congressional investigation tea party targeting process laid process find collect ten thousand email agency first irs material protected careful encryption oconnor said need would run search term congressional committee identified material done material would move review tool oconnor said acting irs commissioner dan directing agency turn document fully quickly possible response congressional request arrived oconnor saw irs resource place operation needed add people irs never anything like oconnor told panel didnt staff place kind document review production added agency also add significant server capacity handle request think record irs reflects hard work produce document oconnor concluded irs blasted lost hard drive apology agency chief key question controversy

Original summary: house committee another controversy national official say tell agency lost lost two year worth former

Predicted summary: state belief several georgia belief several iran nation black black black black response black response black response black response black response black response black response black response black response black response black response black round actress round round witness witness black black response response black response black response black response black response includes response 15 al black response black response response night call night call korean korean korean korean korean korean korean minority lawyer lawyer act act try iran increase twice didnt twice twice thousand twice remain source source israel girl north round round pilot thousand israel israel football includes response response pilot city eight storm document kill kill bridge bridge bridge recently bridge recently bridge recently problem ban issue strike strike daniel response response tuesday chris church movie church movie movie church movie really

VII. Conclusion and Future Work

Our learning doesn't stop here. We have many aspects that can implement to improve our implementation to get desired results.

- News data is huge. So we should use more computational power resources to work on such models or should be able to upload data into colab to work on GPU/TPU to train better our models.
- Experiment implementation using Bi-Directional LSTM which is capable of capturing the context from both the directions and results in a better context vector.
- Use beam search strategy for decoding the sequence to see if that produces better results.
- Evaluate performance on different metrics like ROUGH or BLEU score
- Try implementing pointer-generator networks or pretrained BART model
- Try with different regularizations or optimizers at varying rates.

The increasing growth of the Internet has made a huge amount of information available. It is difficult for humans to summarize large amounts of text. Thus, there is an immense need for automatic summarization tools and techniques based on niche topics. The International Data Corporation (IDC) projects that the total amount of digital data circulating annually around the world would sprout from 4.4 zettabytes in 2013 to hit 180 zettabytes in 2025. That's a huge amount of data circulating in the digital world. There is a need for algorithms which can be used to automatically shorten the amount of data with accurate summaries that capture the essence of the intended messages. Furthermore, applying text summarization reduces reading time and accelerates the process of researching for information playing a major role in the current era of rapid development and digitalization. Humans are generally quite good at this task as we have the capacity to understand the meaning of a text document and extract salient features to summarize the documents using our own words. However, automatic methods for text summarization are crucial in today's world where there is an over-abundance of data and lack of manpower as well as time to interpret the data.

References :

[1] <https://cs.nyu.edu/~kcho/DMQA/>

[2] Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., & Du, Q. (2018). A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. arXiv preprint arXiv:1805.03616.

[3] Long Short-Term Memory (Sepp Hochreiter and Jürgen Schmidhuber), In Neural Computation, volume 9, 1997.

[4] <https://arxiv.org/pdf/1706.03762.pdf>

VIII. Implementation and technical challenges

I looked at current text summary approaches before developing the text summing tool. The field of text summarization in Natural Language Processing is still in its infancy (NLP). Deep learning, a form of data analysis, has given legislature solutions in widely accepted Natural language processing tasks like Name Item Recognition (NER), Part of Sound (POS) tracking, and text. Abstractive literary summation are the foremost common techniques to sentiment analysis. So I decided to implement sequence to sequence model and deep dive into it.

Implementation :

All the dataset was read. Duplicates and NA values were then dropped. The data was cleaned using typical text cleaning operations. The text was cleaned by removing HTML tags, , any parenthesis text was removed, stopwords were removed. The same was done to clean the summaries present in both datasets. Same text preprocessing was applied to the news dataset. Then the start and end tokens were added to the cleaned summary. The text lengths are analyzed to get the maximum length of the sotry and highlight. The final data frame was created to contain that data only with clean story and clean highlight below or equal to the set maximum. The data was split into train and test set with 0.08 in train and 0.02 in a test. The story and highlight word sequences were converted into integer sequences using tokenizers and topmost common words. The Encoder model consisting of three LSTM layers stacked on top of each other was made and the Decoder was initialized with encoder states. A dense layer with softmax activation was added at the end. This was the setting up of the training phase for both Encoder and Decoder. The model was compiled using sparse categorical cross-entropy as the loss function. Early stopping was used to stop training the model if validation loss started increasing for highlight, the model stopped training at 50 epochs and for news, only ~7 epochs were used due to time and machine power constraints. The encoder and decoder inference phase was set up, encoder inputs and outputs from training were supplied as inputs to inference . The decoder was set up in the inference phase and to predict the next word in the sequence, initial states were set to the states from the previous time step. An inference function to decode input sequence was created which creates target sequence until end token is reached or max summary length is reached. Then the summaries were generated for the test set.

Technical Challenges:

The major challenge for this project is data. Data is huge. So I took a very small amount of data as sample and worked on it. Even though models took 1.5 days for each model to run. After each model run, due to utilizing all the processor power, anaconda was getting freeze and kernel got dead. So to run another model, I had to run all the steps from beginning, text loading, cleaning, tokenizing, till loading attention model. So this was the additional time taking task to run any model.

Attention class was not getting imported so I had to create the class locally on this project to use attention layers.

Using libraries like numpy, scipy and scikit was a challenge. Since underlying libraries needs specific versions that goes with. So implementation was blocked due to these prerequisites.

To resolve issue of computational power and libraries versions, I decided to work with google colab. But couldn't get the data to be transfer since google drive was hanging my computer.

IX. The limitations of problem formulation, hypotheses, methods, and findings

For CNN News Text Summarization project, Understanding the problem statement was clear but going through many algorithms, understanding concepts which are heavily math oriented and implementing and evaluating results them was a challenging part. Time is constraint to apply all the knowledge which we have learned. I still feel the awesome concepts which I have learned during NLP and Neural Network, Machine Learning and other statistical concepts, how they goes with the implementation. I am eager to analyze the concepts of dependency parsing, normalization, n-gram, Bayes theorem, semantic roles, coreference if these concepts puts more light on my existing project to get better outcomes.

I believe pretrained model, even though they are trained on a large dataset having a large corpus, those models have limitation to use for only on certain types of data. Like, law sector or even educational sector will have words more frequently like teacher, student, homework, classwork. So we should be using the customized models that goes specifically with such data. So for my project as well, I feel I should train model based on News Data. Due to limited computational power, I couldn't able to use the available dataset and used only smaller part.

Also professionals working on such complex problems, even though it looks simple, should have recent knowledge, be able to work with no time constraint and be able to utilize all the earlier knowledge is very important.

Using powerful resources for such big dataset , having all the knowledge in right direction, getting resources like some of the publications were not accessible from ieeexplore.ieee.org (which is a good resource to understand the concepts, finding methods and come up with the findings) and utilizing the knowledge I have is the major limitation for this project.

On other nodes, I enjoyed to apply and learn knowledge of embedding, sequence to sequence model, attention model, Tokenizing, Text cleaning and understanding approaches. Since this is the beginning step for me, I feel little comfortable to deep dive into the concepts to resolve or attempt any challenges society is facing.