**Abstractive Text Summarization**

***A Report submitted***

***in partial fulfillment for the Degree of***

***Bachelor of Technology***

***in***

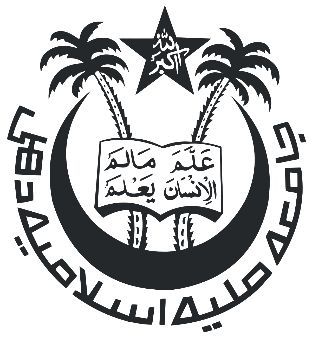
***Computer Engineering***

***by***

***Habiburrahman 17BCS071***

***Khan Mohd Arquam 17BCS008***

***Saman Rashid 17BCS094***



***Department of Computer Engineering***

***Faculty Engineering & Technology,***

***Jamia Millia Islamia New Delhi –110025***

# CERTIFICATE

This is to certify that the project report entitled **Abstractive Text Summarization** submitted by **Khan Mohd Arquam, Habibur Rahman, Saman Rashid** to the Department of Computer Engineering, F/O Engineering & Technology, Jamia Millia Islamia New Delhi – 110025 in partial fulfillment for the award of the degree of B. Tech in (Computer Engineering) is a bona fide record of project work carried out by him/her under my/our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree.

<Signature>

Mr. Faiyaz Ahmad,

Supervisor,

Department of Computer Engineering

New Delhi Counter signature of HOD with seal

July. 2021

# DECLARATION

I declare that this project report titled Abstractive Text Summarization submitted in partial fulfillment of the degree of B. Tech in (Computer Engineering) is a record of original work carried out by me under the supervision of Dr. Sarfaraz Masood and has not formed the basis for the award of any other degree, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

<Signature> <Signature> <Signature>

Khan Mohd Arquam, Habibur Rahman, Saman Rashid,

17BCS008 17BCS071 17BCS094

New Delhi-110025

07/07/2021

# ACKNOWLEDGMENTS

We take this opportunity to thank Dr. Tanvir Ahmad, Dr. Bashir Alam, Dr. Sarfaraz Masood, Mr, Faiyaz Ahmad and other faculty members of the Department of Computer Engineering, Jamia Millia Islamia who helped in preparing the guidelines

.

We extend our sincere thanks to all members Department of Computer Engineering family including the teaching and non-teaching staff and also fellow students for the completion of this document on the project report format guidelines.

Khan Mohd Arquam Habibur Rahman Saman Rashid

# ABSTRACT

For the challenging task of abstractive text summarization, sequence-to-sequence RNN LSTM models have provided a different, efficient and novel approach. Thus, unlike extractive summarizers, abstractive summarizers not only select and rearrange text from source article. Among abstractive summarizers, almost all of the previous work has been done on sequence-to-sequence models. These models have two major limitations: First, they often write wrong and ambiguous factual details, and Second, they quite often repeat sentences/words/phrases. In this project, we have used a sophisticated architecture that uses the regular sequence-to-sequence attention-based model in two different ways. We use a pointer-generator network which enables copying of original words from the original document with the help of pointing. This technique allows us to generate more accurate and correct factual information without compromising the ability to generate new words from the vocabulary. Apart from Pointer-Generator network, we also use a coverage vector to maintain a record words/phrases that have already been used in the summary. This in turn allows us to tackle the problem of repetition. We train and test this model on the massively popular CNN/Daily Mail Dataset and show that it generates ROUGE scores comparable to the best available models.

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Description** | **Page No.** |
| 1 | An illustration of the extractive summarization | 9 |
| 2 | An illustration of the abstractive summarization | 10 |
| 3 | Switching generator/pointer model as proposed by Nallapati et al, 2016 | 13 |
| 4 | Neural Network based Coverage model from Tu et al., 2016 | 14 |
| 5 | Pointer-generator network architecture | 23 |
| 6 | Main graph of our model | 30 |
| 7 | Auxiliary Nodes of our model | 30 |
| 8 | Training Loss graph | 31 |
| 9 | Validation loss graph | 31 |

# LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Description** | **Page No.** |
| 1 | A comparative study on abstractive summarization | 16 |
| 2 | ROUGE results | 33 |
| 3 | Comparison of our model with recent work | 34 |

# ABBREVIATIONS/NOMENCLATURE/ NOTATIONS

|  |  |
| --- | --- |
| **Abbreviation /Nomenclature/ Notation** | **Description** |
| LSTM | Long Short Term Memory |
| OOV | Out of Vocabulary |
| RNN | Recurrent Neural Network |
| NATS | Neural Abstractive Text Summarizer |
| NLP | Natural Language Processing |
| ROUGE | Recall-Oriented Understudy for Gisting Evaluation |
| METEOR | Metric for Evaluation of Translation with Explicit ORdering |

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Description | Page No. |
| CERTIFICATE | 1 |
| DECLARATION | 2 |
| ACKNOWLEDGEMENTS | 3 |
| ABSTRACT | 4 |
| LIST OF FIGURES | 5 |
| LIST OF TABLES | 6 |
| ABBREVIATIONS/NOMENCALTURE/NOTATIONS | 6 |
| 1. INTRODUCTION | 8 |
| 1.1 Background | 9 |
| 1.1.1 What is text Summarization | 9 |
| 1.1.2 Extractive Summarization | 9 |
| 1.1.3 Abstractive Summarization | 9 |
| 2. LITERATUE REVIEW | 12 |
| 3. PROBLEM STATEMENT | 18 |
| 3.1 Inaccurate Factual Details | 19 |
| 3.2 Repetition of Sentences | 19 |
| 4. PROPOSED METHOD AND EXPLANATION | 20 |
| 4.1 Sequence-to-Sequence Attention Based Model | 21 |
| 4.2 Pointer Generator Network | 22 |
| 4.3 Coverage Mechanism | 24 |
| 5. IMPLEMENTATION AND EXPERIMENTAL RESULTS | 26 |
| 5.1 Dataset Description | 27 |
| 5.1.1 Dataset sample | 27 |
| 5.1.2 Dataset Preparation | 28 |
| 5.2 Creating our Model | 29 |
| 5.3 Experimental Results | 31 |
| 5.4 Evaluation Metrics | 32 |
| 5.5 Test Results | 33 |
| 5.6 Sample Summary | 34 |
| 6. CONCLUSION AND SUGGESTIONS | 36 |
| 6.1 Conclusion | 37 |
| 6.2 Suggestions for Future Work | 37 |
| REFERENCES | 38 |

# Chapter 1

# INTRODUCTION

# 1.1 Background

## 

## **1.1.1 What is text Summarization**

The task of condensing a piece of text/article to a more concise version containing the important and requisite information from the original text.

There are two approaches to summarization:

* Extractive Summarization
* Abstractive Summarization

## **1.1.2 Extractive Summarization**

Extractive methods generate the summary exclusively from the original text (often whole sentences) directly copied from the original text/article, the extractive method is simpler, since copying large amount of text from the source text often results in accurate grammar and also less misinformation.

**Summary**

Sentence 2

Sentence 4

**Text**

Sentence 1

Sentence 2

Sentence 3

Sentence 4

Extractive Summarizer

**Fig. 1: An illustration of the extractive summarization**

## **1.1.3 Abstractive Summarization**

The abstractive methods of summarization is used to generate new words and phrases that are not present in the original document. Hence, this is considered much similar to a human written summary.

Much complex abilities that are very important to achieve human-like summarization, such as generalization, the addition of context knowledge, or paraphrasing, are possible only in the abstractive method.

**Summary**

New Sentences

Abstractive Summarizer

**Text**

Sentence 1

Sentence 2

Sentence 3

Sentence 4

**Fig. 2: An illustration of the abstractive summarization**

Due to the difficulty of abstractive summarization, the great majority of past work has been extractive (Kupiec et al., 1995; Paice, 1990; Saggion and Poibeau, 2013). However, the recent success of sequence-to-sequence models (Sutskeve el al., 2014) in which recurrent neural networks (RNNs) both read and freely generate text, has made abstractive summarization viable (Chopra et al., 2016; Nallapati et al., 2016; Rush et al., 2015; Zeng et al., 2016). Though these systems are promising, they exhibit undesirable behavior such as inaccurately reproducing factual details, an inability to deal with out-of-vocabulary (OOV) words, and repeating themselves.

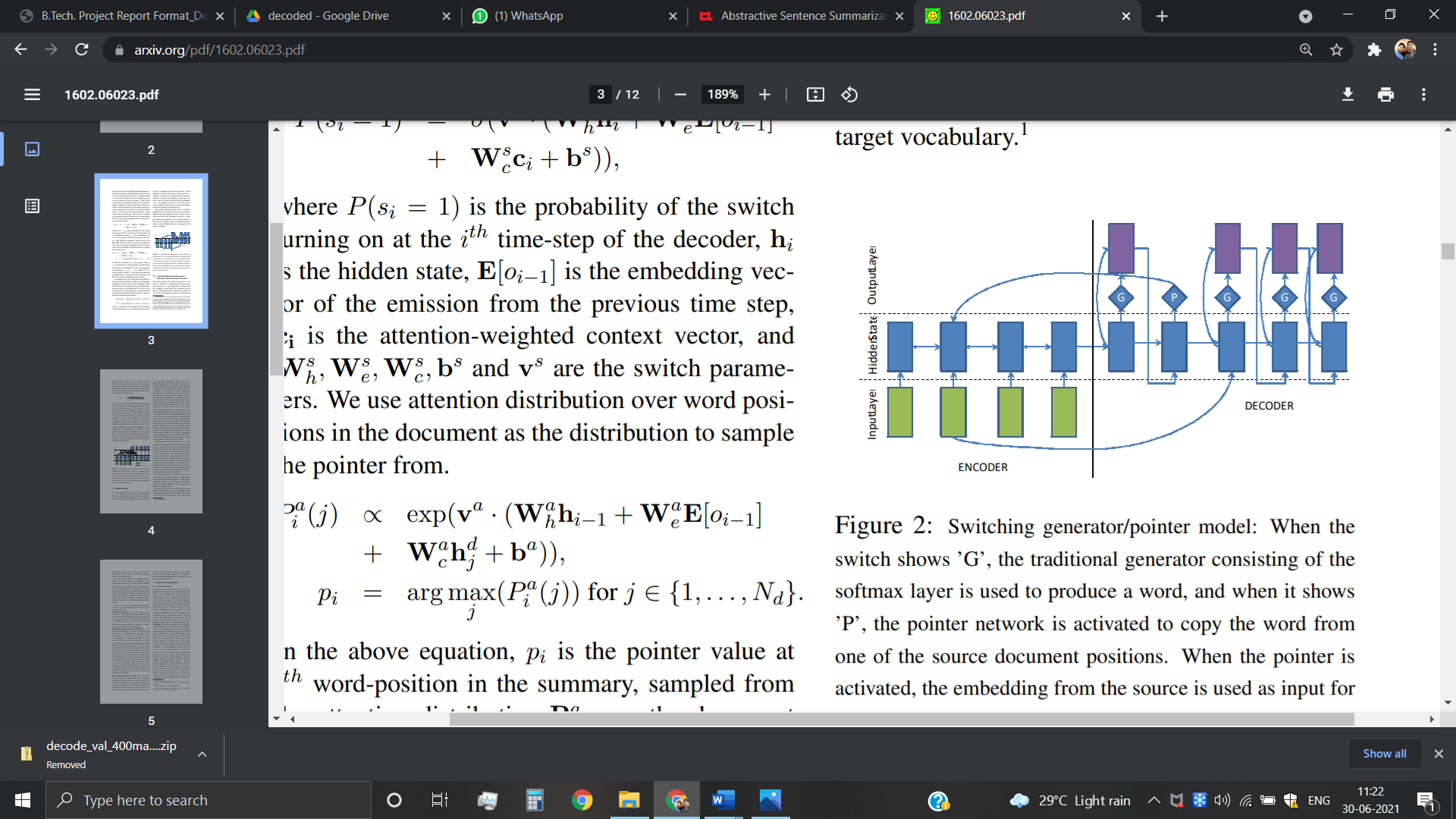
In this report we present an architecture that addresses these three issues in the context of multi-sentence summaries. While most recent abstractive work has focused on headline generation tasks (reducing one or two sentences to a single headline), we believe that longer-text summarization is both more challenging (requiring higher levels of abstraction while avoiding repetition) and ultimately more useful. Therefore, we apply our model to the recently-introduced CNN/ Daily Mail dataset (Hermann et al., 2015; Nallapati et al., 2016), which contains news articles (39 sentences on average) paired with multi-sentence summaries, and show that we outperform the state -f-the-art abstractive system by at least 2 ROUGE points.

Our hybrid pointer-generator network facilitates copying words from the source text via pointing (Vinyals et al., 2015), which improves accuracy and handling of OOV words, while retaining the ability to generate new words. The network, which can be viewed as a balance between extractive and abstractive approaches, is similar to Gu et al.’s (2016) Copy-Net and Miao and Blunsom’s (2016) Forced-Attention Sentence Compression, that were applied to short-text summarization. We propose a novel variant of the coverage vector (Tu et al., 2016) from Neural Machine Translation, which we use to track and control coverage of the source document. We show that coverage is remarkably effective for eliminating repetition.

# Chapter 2

# LITERATURE REVIEW

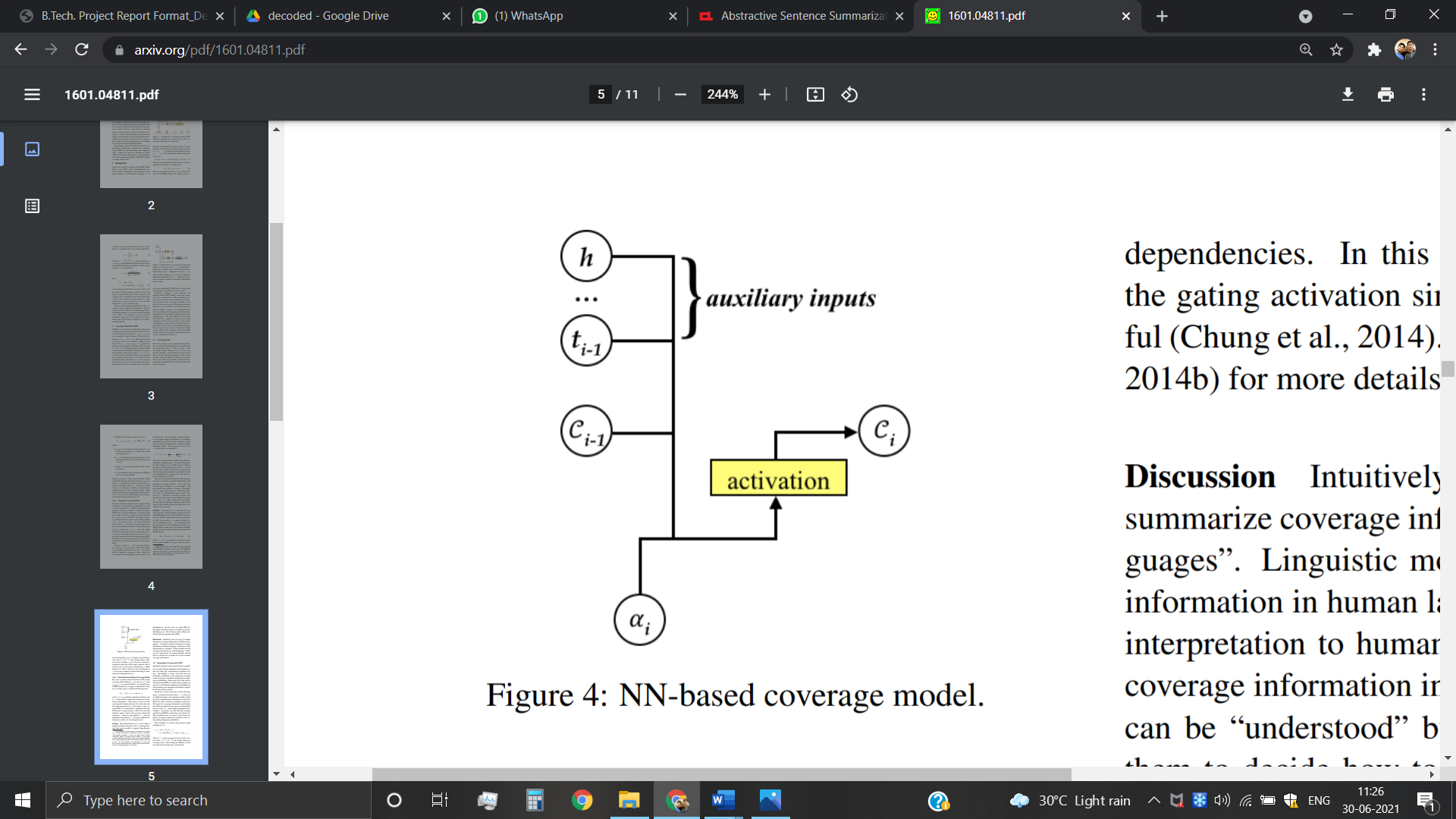
Since the abstractive method of summarization is quite complex and difficult to achieve, the majority of the past work has been based on the extractive method of summarization (Kupiec et al., 1995; Paice, 1990; Saggion and Poibeau, 2013). But however, the recent success of sequence-to-sequence models (Sutskever el at 2014) , wherein recurrent neural networks (RNNs) have the ability to both read and generate summaries on their own, has made the seemingly impossible task of abstractive summarization look possible (Chopra et al., 2016; Nallapati et al., 2016; Rush et al., 2015; Zeng et al., 2016). Even though these works are very good and should be acknowledged, they have their own flaws such as they inaccurately reproduce factual information, and are also uncapable of dealing with words not present in our vocabulary, and they often repeat words/phrases.



**Fig. 3: Switching generator/pointer model as proposed by Nallapati et al, 2016**

In this project we propose a much newer model that tries to solve these three major problems in the context of multi-sentence summaries. However, most recent abstractive work has been focused on generating headlines of news articles (summarizing one or two sentences to a single sentence), but here we take a different path which will be more useful and is a requirement today. Hence, we work our proposal to the CNN/ Daily Mail dataset (Hermann et al., 2015; Nallapati et al., 2016), which contains news articles along with reference summaries having multiple sentences.

The proposed pointer-generator model enables extraction of words from the original document (Vinyals et al., 2015), this helps in managing the problem of Out of Vocabulary words and also helps improve accuracy, at the same time it also keeps on its task of generating new and novel words. This network, can be seen as a combination of extractive and abstractive methods, hence it can be considered similar to Gu et al.’s (2016) CopyNet and Miao and Blunsom’s (2016) Forced-Attention Sentence Compression, that were applied to short-text summarization. A coverage vector (Tu et al., 2016) from Neural Machine Translation is also used, which will enable us to manage coverage of the original text. Hence, we show that coverage will reduce repetition.



**Fig. 4: Neural Network based Coverage model from Tu et al., 2016**

Firstly, as per the work done on Neural Sequence to Sequence attention models, these models have shown results that are quite promising in context of Abstractive Text Summarization. However, there are many issues that pertain. The generated summaries are often meaningless; hence many techniques have been tried to solve these issues.

The reinforcement learning based training procedures using intra-Attention that improves the model’s accuracy remarkably has been explored as well. The problems that plagued the area in detail, and the possible ways to improve those areas were also analyzed in previous works. Also, a novel architecture was proposed to solve the issues and problems in summary generation of longer text which were difficult to be captured with currently ongoing used models. With their deep learning approaches, they succeeded to a great extent and finally concluded that Deep Learning approach are quite promising and would be helpful in the near future to solve the abstractive text summarization issues in the near future.

However, the tough task of scalability and generalization process of multi- sentence summarization is plagued by problems with the evaluation metrics (ROUGE, METEOR) and lack of dataset availability.

Hence it was finally concluded that Deep Learning approach are much relevant and would be very helpful in further research in this domain and to tackle the challenge of abstractive text summarization.

Talking about other works, the usage of Sequence-to-Sequence model was shown and developed by Tian Shi et al. According to them, neural sequence-to-sequence model has become quite renowned in the past years. To handle future challenges in this domain of text summarization, many interesting techniques have been developed. This will improve the sequence-to-sequence models. Also, they believe that the majority of the techniques vary among the three categories: network structure, parameter inference and decoding/generation. At times, efficiency of a model becomes a concern during the training phase. In their paper publication, a detailed literature as well as technical survey on various sequence-to-sequence models for abstractive text summarization was presented.

Many of their models were primarily used and proposed for modelling of the language and task generation, such as machine translation, and were applied later to abstractive text summarization.

An open source library was also developed, named as Neural Abstractive Text Summarizer (NATS) toolkit, to enable the task of abstractive text summarization, during their survey. On the CNN/Daily Mail dataset a number of examinations were conducted to evaluate its effects of various different neural network components. Finally, two models implemented in NATS on two novel contributed datasets, such as, Newsroom and Bytecup were compared and confirmed. Hence, their effectiveness in their work was proved. Therefore, it was concluded that this application was quite successful for sequence-to-sequence models. Hence, neural abstractive text summarization has been a quite popular research area and is in demand from both the industry as well as the academics.

Further, going ahead to see something different, is the use of Feed-Forward network by Lu Yang, This focus of this research is to use the feed-forward neural network with an attention-based encoder in order to meet the problems of abstractive text summarization. They analyzed the attentive recurrent neural network and recurrent neural network encoder-decoder to check its effectiveness and power. These discussed models were mainly developed for solving tasks, such as summarization of newspaper publication and translation of machine context. They made assessment on model’s advantages using ROUGE and results of visual inspection by modifying and furthering these models to the issue of product review summarization. In the end, they discovered the outcomes to be favorable.

**Table 1: A comparative study on abstractive summarization**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year** | **References** | **Highlight** | **Model used** | **Dataset** | | **Outcome** |
| 2015 | Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Empirical Methods in Natural Language Processing*. | Reinforcement learning based training procedures using intra-Attention. | Neural Sequence to Sequence Model | The standard sentence summarization evaluation set is associated with the DUC-2003 and DUC2004 | | They presented a neural attention-based model for abstractive summarization, based on recent developments in neural machine translation. They combined this probabilistic model with a generation algorithm which produced accurate abstractive summaries. |
| 2016 | Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In Empirical Methods in Natural Language Processing. | Neural network-based encoder-decoder models are among recent attractive methodologies for tackling natural language generation tasks | AMR, 2 Attention-Based AMR Encoder | Data of the abstractive headline generation task described in Rush et al. (2015). | | The experimental results of headline generation benchmark data showed that our attention-based AMR encoder-decoder model successfully improved standard automatic evaluation measures of headline generation tasks, ROUGE-1, ROUGE-2, and ROUGEL |
| 2016 | Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *North American Chapter of the Association for Computational Linguistics*. | introduce a conditional recurrent neural network (RNN) which generates a summary of an input sentence. The conditioning is provided by a novel convolutional attention-based encoder which ensures that the decoder focuses on the appropriate input words at each step of generation. | Attentive Recurrent Architecture, Recurrent Decoder, Attentive Encoder | The Gigaword corpus (Graff et al., 2003; Napoles et al., 2012) | | Extended the state-of-the-art model for abstractive sentence summarization (Rush et al., 2015) to a recurrent neural network architecture. This model is a simplified version of the encoder-decoder framework for machine translation (Bahdanau et al., 2014). |
| 2016 | Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, C¸ aglar Gulc¸ehre, and Bing Xiang. 2016. Abstrac- tive text summarization using sequence-to-sequence RNNs and beyond. In *Computational Natural Lan- guage Learning*. | Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond | Encoder-Decoder RNN with Attention and Large Vocabulary Trick , Capturing Keywords using Feature-rich Encoder | | Gigaword Corpus | Proposed several novel models that address critical problems in summarization that are not adequately modeled by the basic architecture, such as modeling key-words, capturing the hierarchy of sentence-toword structure, and emitting words that are rare or unseen at training time |

# CHAPTER 3

# PROBLEM STATEMENT

## 

The task of abstractive text summarization faces two major problems:

## **3.1 Inaccurate Factual Details**

The summaries sometimes **emulate factual details incorrectly**. This is most common for infrequent or words beyond vocabulary. It is too difficult to copy a word ‘w’ from the source text with the sequence-to-sequence-with attention model. Essentially, the original word must somehow be recovered by the network after the information has passed through several layers of computation (including mapping w to its word Embedding). This is especially true if w has a poor word embedding (i.e. it is clustered with completely unrelated words) because it is a sparse/rare word that develops infrequently during training and therefore, then w is, from the understanding of the network, identical from many other words and hence impossible to duplicate. The network may still have difficulty reproducing the word even if w has a good word embedding. For example, RNN summarization systems often mistake and erroneously substitute a name with another name (e.g. Mary → Katie) or a city with another city (e.g. Kolkata → Chennai). This is because the word embeddings for e.g. names of women or names of Indian cities tend to cluster together, which may cause confusion when attempting to reproduce the original word.

In short, this seems like an unnecessarily difficult way to perform a simple operation – copying – that is a fundamental operation in summarization.

## **3.2 Repetition of Sentences**

The summaries sometimes **repeat a said statement** (e.g. *Austria beat Austria beat Austria beat…*)

These problems are not rare for RNNs in general. It is difficult to explain *why* the network exhibits any particular behavior, a well-known characteristic of deep learning.

For those who are interested, I offer the following conjectures- Repetition may be caused by the decoder’s *over-reliance on the decoder input (i.e. previous summary word)*, rather than storing longer-term information in the decoder state. A single repeated word commonly triggers an endless repetitive cycle, prove this. For example, a single substitution error *Austria beat****Austria*** leads to the catastrophic *Austria beat Austria beat Austria beat…*, and not the less-wrong *Austria beat Austria 2-0*.

# Chapter 4

# PROPOSED METHOD AND EXPLANATION

To tackle the mentioned problems, we propose a model consisting of the following:

(1) Sequence-to-Sequence attention based model

(1) Pointer-Generator model

(2) Coverage Mechanism.

## **4.1 Sequence-to-Sequence Attention Based Model**

A Sequence to Sequence model was developed first with reference to the model proposed by Nallapati et al. (2016). The encoder is fed the tokenized form of the article (wi being considered as a token) one at a time into the encoder. This in turn produces a sequence of encoder hidden states hi. It is important to note that the encoder is a single layer bidirectional LSTM. The decoder which is a single-layer unidirectional LSTM gets the word embedding vector of the last word. This word is the previous word of the reference summary during the training phase, and is the previous word given by the decoder during the testing phase. The decoder state becomes st .This process happens at each timestep ‘t’. We use the formula given by Bahdanau et al. (2015) to calculate the attention distribution,

**eti = vT tanh(Whhi +Wsst +battn) --------(1)**

**at= softmax(et ) -------(2)**

where v, Wh, Ws and battn are learnable parameters.

The attention distribution is nothing but a probability distribution over the words from the original document. It is this distribution that enables the decoder to look and find the words to be written in the summary. Next, the context vector h\*t is generated which is the weighted sum of the encoder hidden states produced with the help of the attention distribution,

**ht∗ = ∑i at hi -------(3)**

The context vector is actually fixed size representation of the source text that has been processed for this particular step. This vector is concatenated with the decoder state st and passed through two different unidirectional layers to generate the distribution over our vocabulary i.e. Pvocab ,

***P*vocab = softmax(*V,*(*V* [*st , ht∗*] + *b*) + b,) -------(4)**

where V, V, , b and b, are learnable parameters.

Pvocab is the distribution of probability over all the words present in our vocabulary. This distribution in turn helps us generate our final distribution from which we generate novel words w:

**P(w) = Pvocab(w) --------(5)**

During the training phase, the loss for timestep t is generated with the help of the following formula,

**Losst = −logP(w\*t ) ---------(6)**

where, w\*t is the target word.

The overall loss for the whole sequence is:

**loss = 1 ( ∑t = 0 T losst ) / T ---------(7)**

## **4.2 Pointer Generator Network**

We create a model which is a hybrid between our sequence-to-sequence model and a pointer network model similar to the one used in Vinyals et al., 2015. This enables the multitasking of both copying words from the source text with the help of pointing, as well as generation of novel words from our pre-defined vocabulary. In this pointer-generator model(as shown in Fig. 5), the attention distribution at and context vector ht\* are calculated. The probability of word generation, pgen ∈ [0,1], is also calculated for timestep t using the context vector ht\*, the decoder state st, and the decoder input xt :

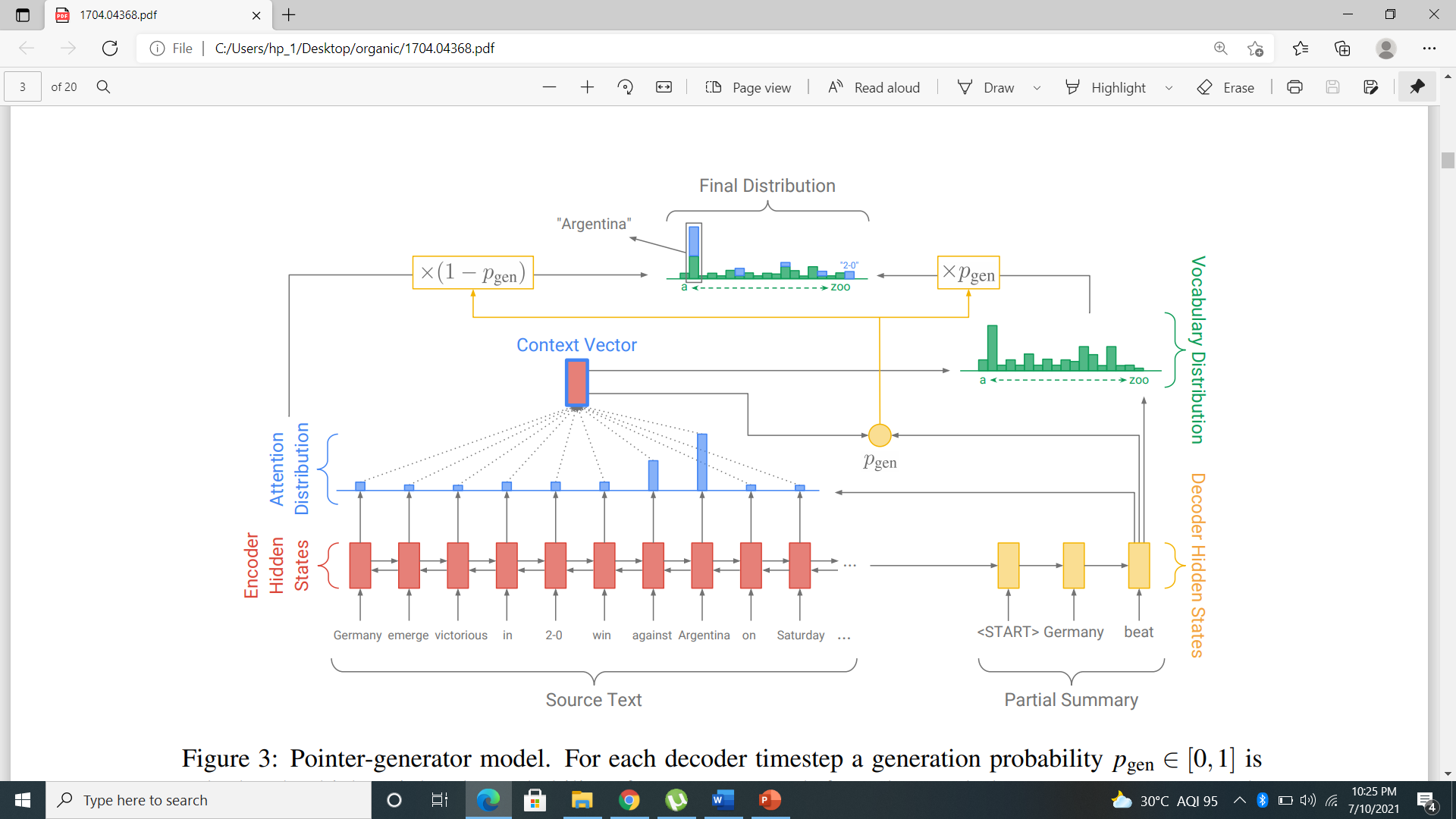
**pgen = σ(whT\* ht\* + wsT st + wxT xt + bptr) --------(8)**

where vectors wh\* , ws , wx and scalar bptr are learnable parameters and σ is the sigmoid function.

Next, the task of choosing whether to generate a novel word from our vocabulary by sampling from Pvocab, or whether to copy a word from the source text by sampling from the attention distribution that was calculated earlier (at) is done by using pgen as a soft switch. For every article, suppose the extended vocabulary denotes the concatenation of the vocabulary, and all words appearing in the original text.

We obtain the following probability distribution over the extended vocabulary:

**P(w) = pgen Pvocab(w) + (1− pgen) ---------(9)**



**Fig. 5: Pointer-generator network architecture**

Suppose w is a word that does not appear in the vocabulary (OOV), then Pvocab(w) will be zero. In a similar fashion, if w does not appear in the source document, then is zero. The ability to produce words that do not appear in the vocabulary (OOV) is one of the biggest advantages of pointer-generator models. In comparison the sequence-to-sequence models are limited to their vocabulary. The loss function is as described in equations (6) and (7), but with respect to our modified probability distribution P(w) given in equation (9).

## **4.2 Coverage Mechanism**

One of the most common problems associated with sequence to sequence models is the problem of repetition(Tu et al., 2016; Mi et al., 2016; Sankaran et al., 2016; Suzuki and Nagata, 2016). This problem worsens when the task is to generate summaries containing multiple sentences. To solve this major problem, we use the coverage model of Tu et al. (2016) along with our sequence-to-sequence pointer-generator model. In this model, a coverage vector ct , which is the sum of attention distributions over all previous decoder timesteps is maintained,

**ct= ∑t-1tt=0 at~ --------(10)**

Hence we can say that the coverage vector, ct, is a distribution over the original article words. Thus this vector represents the amount of coverage that those tokens have got from our attention mechanism till now.

Interestingly, c0 is a zero vector, since at the beginning of the task, no part of the original text has been covered. This vector is then inputted to the attention mechanism, thus equation (1) becomes,

**eti = vT tanh(Whhi +Wsst + wc ct+battn) ---------(11)**

where wc is a learnable parameter vector of same length as v.

Hence we can say that the attention mechanism’s decision of choosing where to look next uses information of its previous outputs. Hence the attention mechanism avoid looking at the same text again, thereby avoiding generation of repetitive text.

However a coverage loss is calculated to penalize the model to attending repeated locations,

**covlosst = ∑imin(ati, cit) ---------(12)**

Since we do not need uniform coverage as demanded by tasks other than summarization, only the overlap between every attention distribution and the coverage till now is penalized. This in turn prevents attention repetition. Lastly, the coverage loss and our loss function are added to generate a new loss function,

**Losst = −logP(wt\* ) + ∑i min(ait , cti ) ----------(13)**

# Chapter 5

# IMPLEMENTATION AND EXPERIMENTAL RESULTS

## **5.1 Dataset Description**

We use the immensely popular CNN/DailyMail dataset for our project. This dataset contains over 3,00,000 news articles along with their respective multi-sentence, human written summaries.

These news articles belong to two prominent news agencies:

1. CNN
2. Dailymail

Out of the approximate 3,00,000 news articles, we used around 10,000 articles for validation, 10,000 articles for testing and the rest (around 2,90,000) articles for the training phase. We also ensure that there is no overlap between any of these sets as this would result in overly promising results.

Also, since the number of Daily mail articles(2,19,000) is greater than the number of CNN articles(92,000), the data from both sets were divided proportionately in the aforementioned distribution (Testing, Validation, and Training).

**5.1.1 Dataset sample**

**News Article:**

(CNN) -- Andy Murray's first match since undergoing back surgery in September ended in a straight sets defeat to Jo-Wilfried Tsonga at an exhibition tournament in Abu Dhabi Thursday.

The reigning Wimbledon champion went down 7-5 6-3 to the Frenchman, who himself was plagued by injury at the back end of this year.

Murray, who has dropped to No.4 in the rankings, lacked sharpness after his layoff and was broken in the 12th game of the opening set to fall behind.

The British star has been training at his base in Florida to prepare for the upcoming season and looked set to even the match up when he gained an early break of service in the second set.

But Tsonga hit back with two breaks of his own to wrap up victory in 72 minutes at the Zayed Sports City complex.

"The courts here are very fast and you have to react quickly," said 26-year-old Murray.

"Jo was sharper than me today, he served very well.

"It's always good fun here. It's great preparation for the season as you have to play against the best in the world."

The organizers of the Mubadala World Tennis Championship have indeed attracted a stellar field with the top two ranked players, Rafael Nadal and Novak Djokovic, in the line-up.

David Ferrer of Spain won the opening match Thursday as he beat Stanislas Wawrinka of Switzerland 7-5 6-1 to set up a semifinal clash against compatriot Nadal.

Tsonga's win over Murray has earned him a match against Serbia's Djokovic, while Murray will gain much-needed match practice against Wawrinka in the fifth place playoff.

Murray, recently voted BBC Sports Personality of the Year back in the UK, became the first British man to win the Wimbledon title in 77 years when he triumphed at the All England Club back in July, but his season took a turn for the worse as he became troubled by a long-standing back problem.

**Reference summary:**

Andy Murray falls to Jo-Wilfried Tsonga at an exhibition tournament in Abu Dhabi.

Wimbledon champion Murray loses 7-5 6-3 to the hard-hitting Frenchman.

It was Murray's first match since undergoing back surgery in September.

David Ferrer defeats Stanislas Wawrinka in straight sets in the other match.

## **5.1.2 Dataset Preparation**

Before using the dataset we first prepare the data to be ready for fast and efficient processing. We begin with tokenizing the data, i.e., converting our data into the form of tokens. Since we are using Recurrent Neural Networks, we need sequential data in the form of tokens. For this task, we use Stanford NLP Tokenizer which is a software that has been developed by Stanford University for this very particular task. We also count the number of time each token appears in the dataset. With this we create a vocabulary file which lists the token(word) and its frequency over the dataset.

Next, we convert the tokenized files into .bin(binary) files. This is done to enable better efficiency and speedup the training process.

## **5.2 Creating our Model**

We create our model using the Tensorflow library. Tensorflow is a library in python which enables us to build machine learning models. To optimize our model, we use the Adagrad optimizer. The following are some of the hyperparameter values that we used,

Batch size= 16

Word embeddings dimension= 128

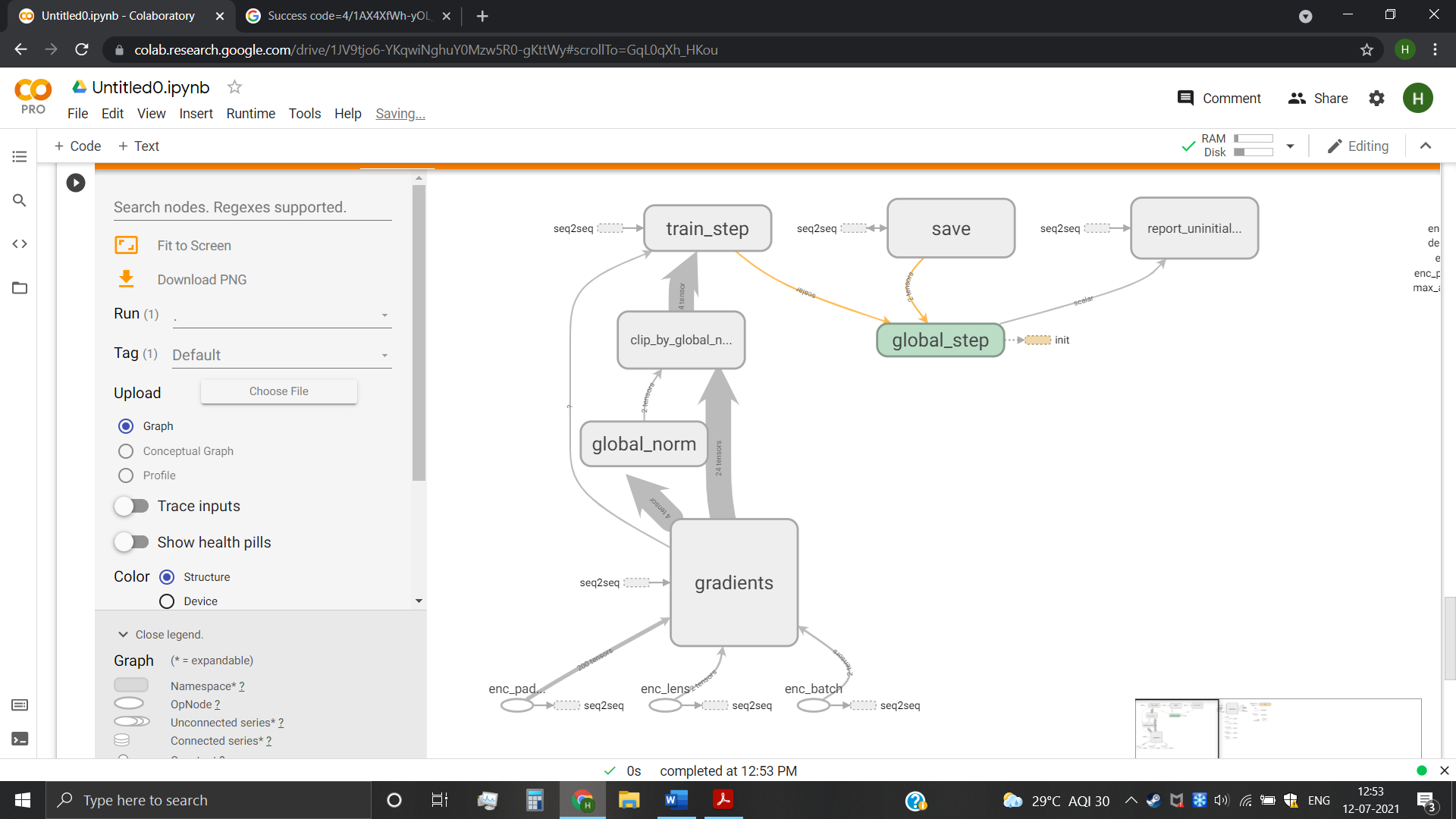
Vocabulary size= 50,000

Learning rate= 0.1

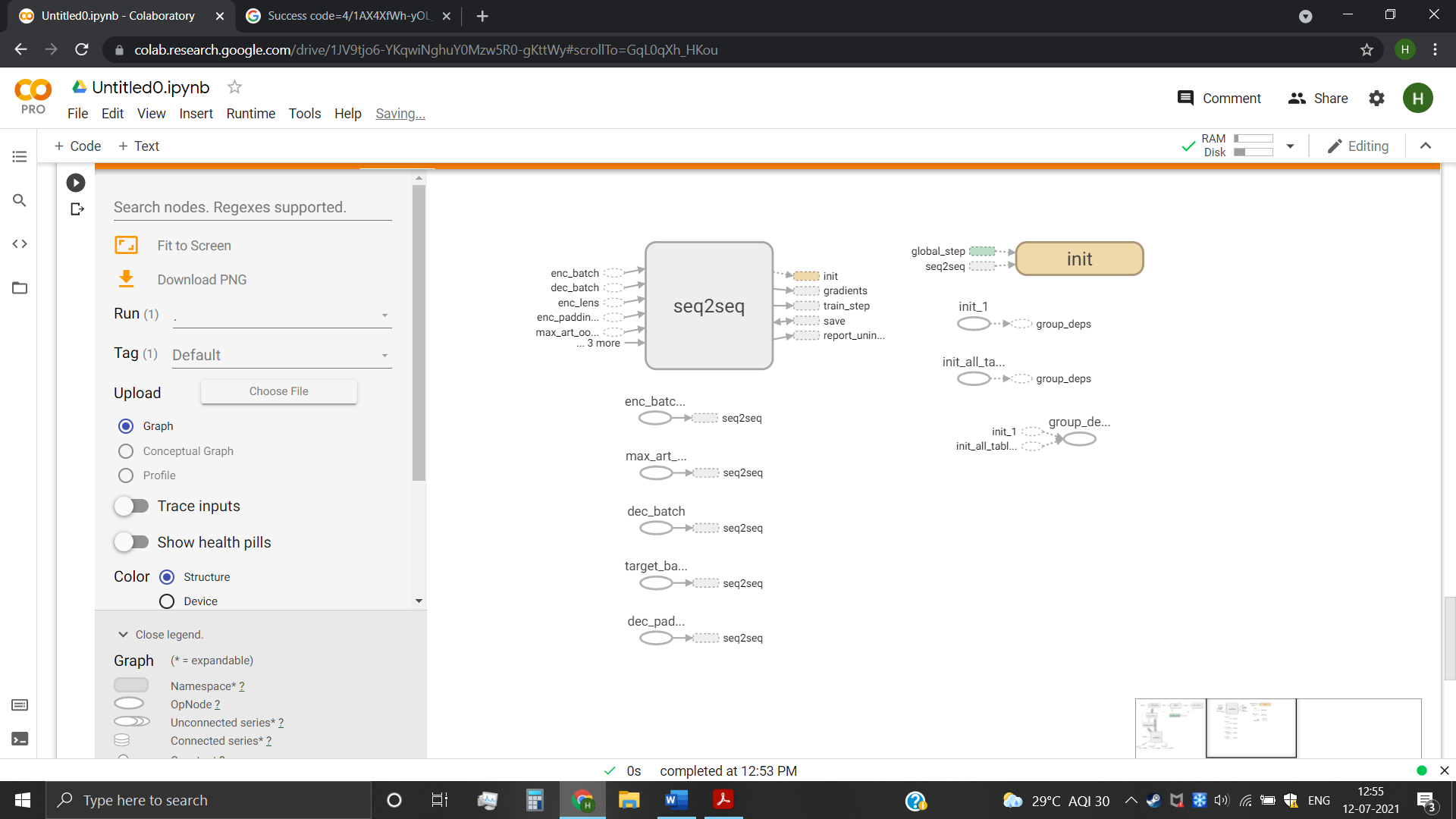
Maximum encoder steps= 400

Maximum decoder steps= 100

The structure of the graph of our model is shown in figure 6 and figure 7.



**Fig. 6: Main graph of our model**



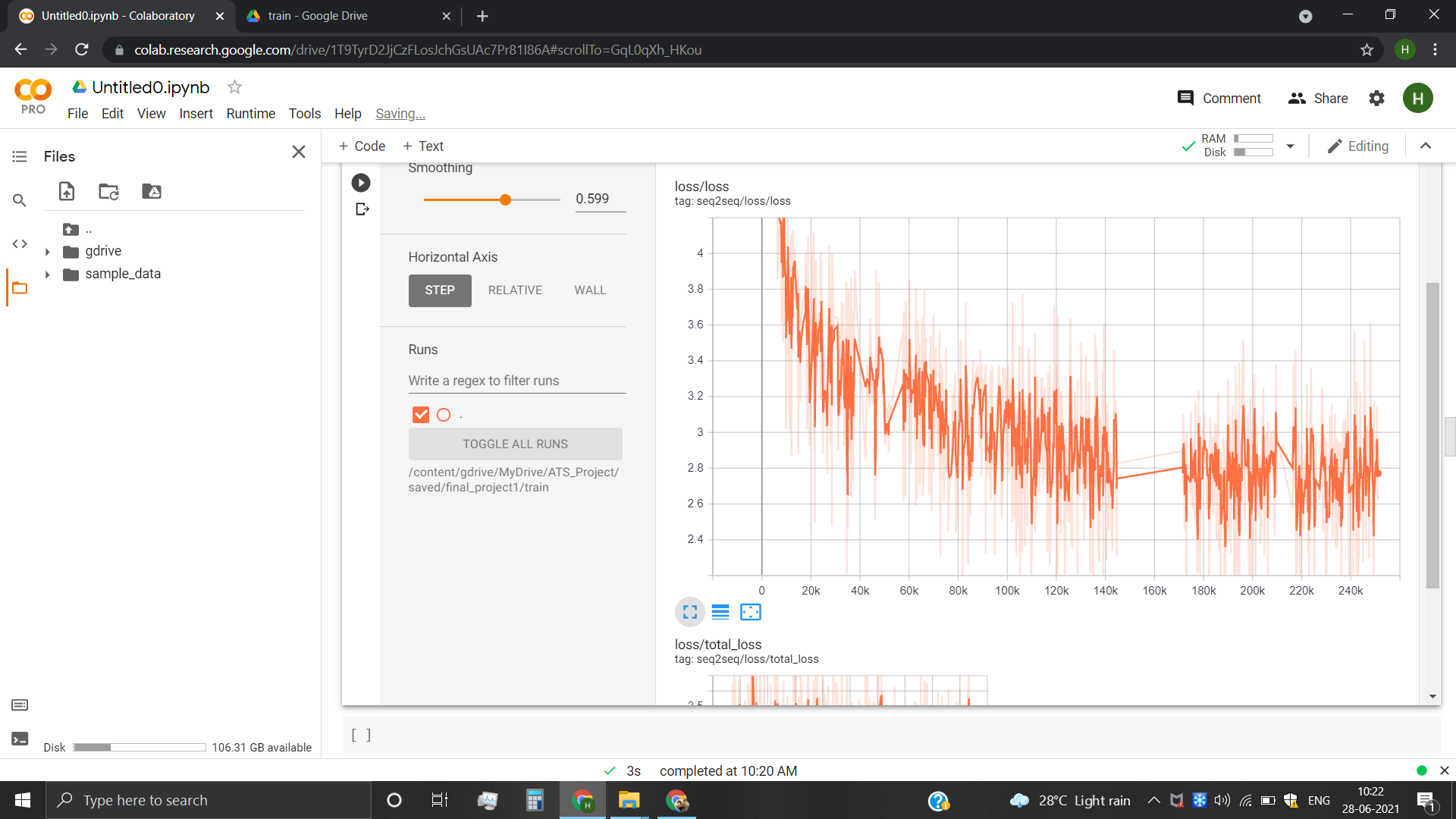
**Fig. 7: Auxiliary Nodes of our model**

# 5.3 Experimental Results

We train our model using Google Colaboratory’s environment on our training dataset (around 2,90,000) for our 2,50,000 iterations, which translates to around 15 epochs for our specified batch size.

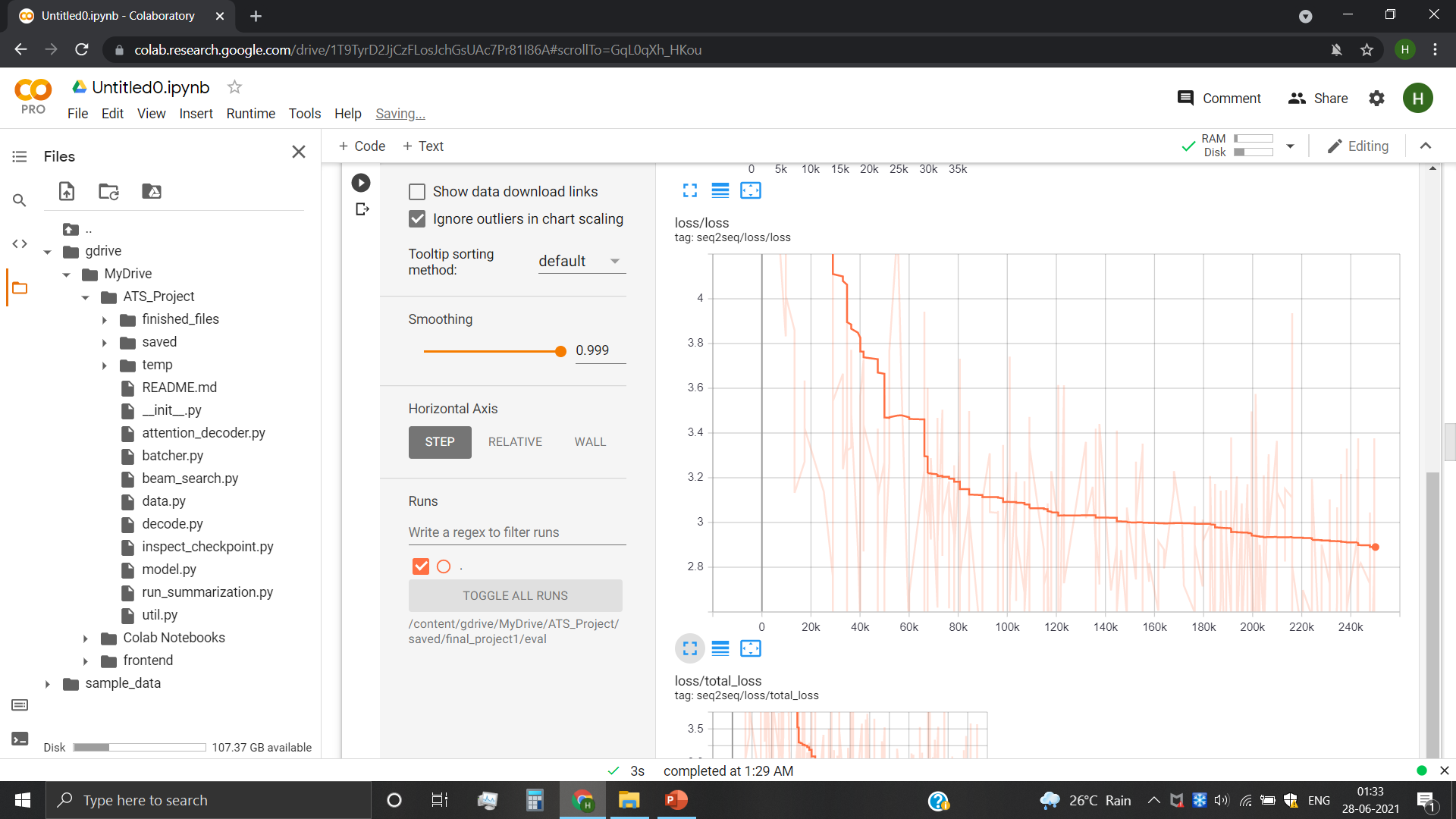
To check for overfitting, we also validate our model regularly using the validation dataset(around 10,000 articles)

The following are the visualization of the training and validation processes:



**Fig. 8: Training Loss graph**

As we can see from the training loss graph, the loss began at a value of around 10. The loss value quickly reduces to around 4 after 8000 iterations. Finally, after 250,000 iterations the value of loss goes as low as 2.5



**Fig. 9: Validation loss graph**

As we can see from the validation loss graph, the graph exhibits similar behavior to training loss graph. It begins with a loss value of around 10, and ends up with a loss value of around 2.9 after around 2,50,000. Hence, we can say that there is no case of overfitting.

## **5.4 Evaluation Metrics**

There are very few metrics available to evaluate the performance of text summarizers. The two most commonly used are ROUGE and METEOR. For our project, we used ROUGE scores for evaluating the performance of our model.

ROUGE is a set of metrics used for evaluating automatic text summarization and machine translation in NLP.

We used the following three metrics:

**1. ROUGE-1:** ROGUE-1 refers to the overlap of unigrams between the system and the reference summary.

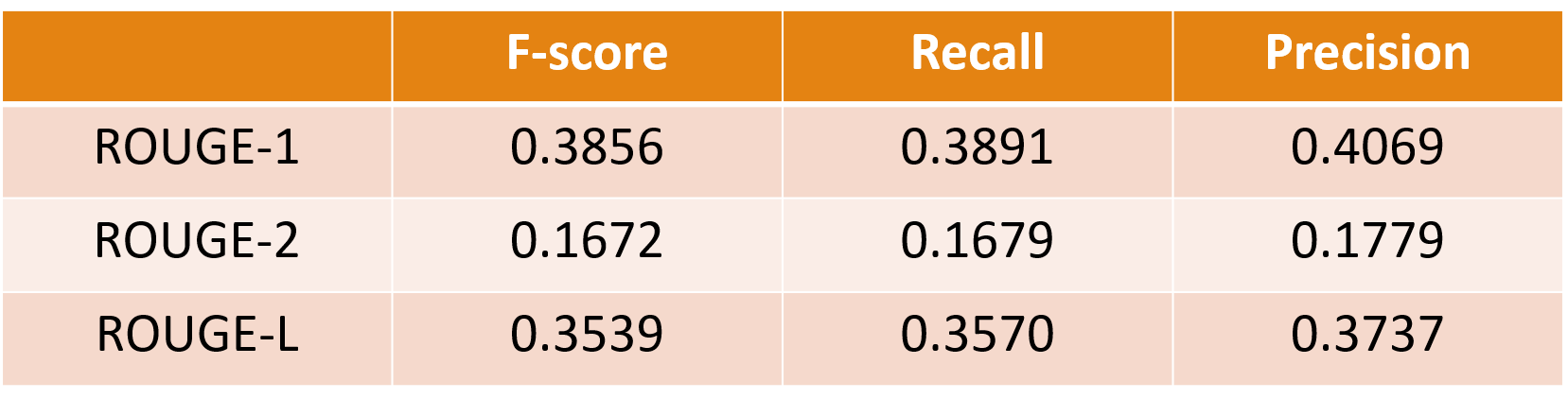
**2. ROUGE-2:** ROGUE-2 refers to the overlap of bigrams between the system and the reference summary.

**3. ROUGE-L:** ROGUE-L takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams.

However, it must be noted that neither of the evaluations metrics presently available is considered as an accurate metric for text summarization tasks, especially abstractive text summarization. These metrics depend upon comparison between the reference summary and generated summary, which may actually be an inaccurate measure.

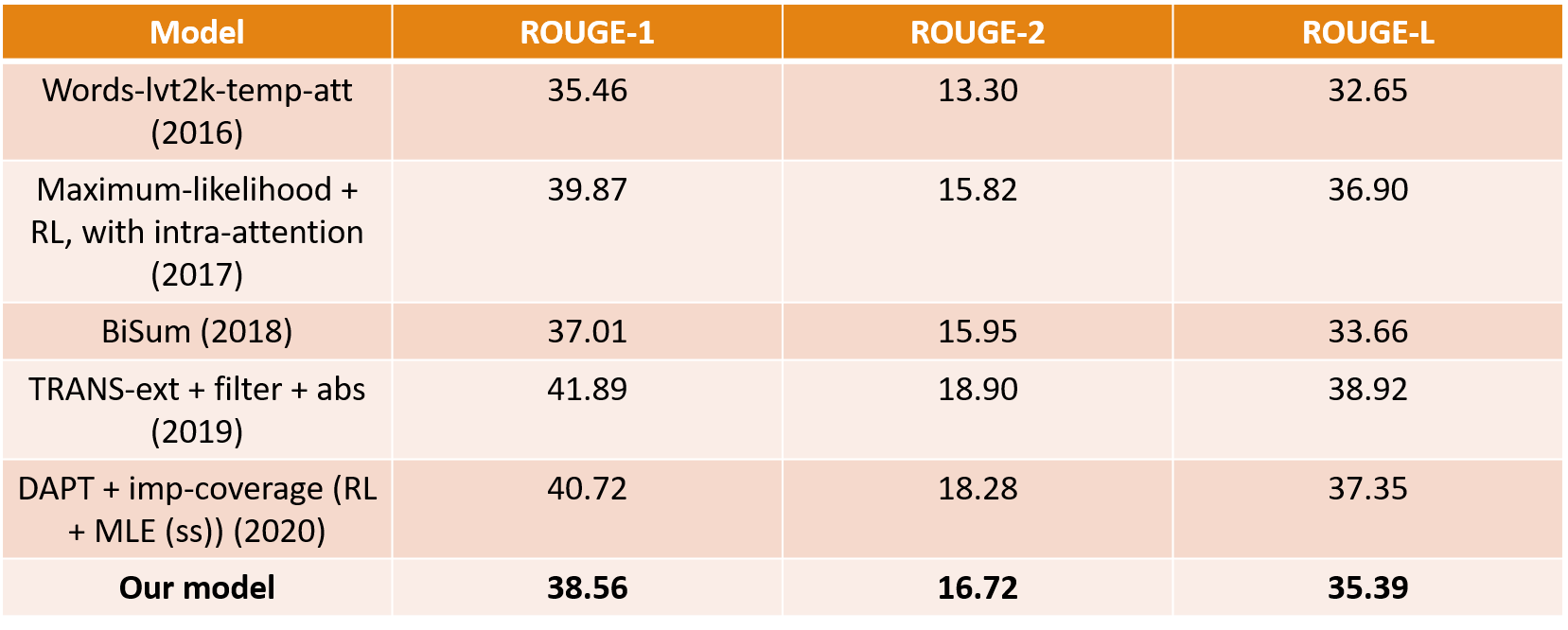
## **5.5 Test Results**

We test our model on the test dataset(around 10,000 articles). The summaries generated by our model are compared to the reference summaries of the dataset to generate the ROUGE results consisting of ROUGE-1, ROUGE-2, and ROUGE-L scores.



**Table 2: ROUGE results**

Comparing ROUGE results of our model with some of the recent work using different technologies,



**Table 3: Comparison of our model with recent work**

Here, we can see that our model yields results similar to some of the previous work in the field of abstractive text summarization. However, as mentioned earlier, ROUGE scores are not a perfect or accurate metric for testing a summarizer and thus we are unsure which of these models is the best.

## **5.6 Sample Summary**

**News Article:**

Former Vice President Walter Mondale was released from the Mayo Clinic on Saturday after being admitted with influenza, hospital spokeswoman Kelley Luckstein said.

"He's doing well. We treated him for flu and cold symptoms and he was released today," she said.

Mondale, 87, was diagnosed after he went to the hospital for a routine checkup following a fever, former President Jimmy Carter said Friday.

"He is in the bed right this moment, but looking forward to come back home," [Carter said](http://nobelpeaceprizeforum.org/live/) during a speech at a Nobel Peace Prize Forum in Minneapolis.

"He said tell everybody he is doing well."

Mondale underwent treatment at the Mayo Clinic in Rochester, Minnesota.

The 42nd vice president served under Carter between 1977 and 1981, and later ran for President, [but lost to Ronald Reagan.](http://www.cnn.com/2013/08/30/us/walter-mondale-fast-facts/) But not before he made history by naming a woman, U.S. Rep. Geraldine A. Ferraro of New York, as his running mate.

Before that, the former lawyer was a U.S. senator from Minnesota.

**Reference Summary:**

walter mondale was released from the mayo clinic on saturday , hospital spokeswoman said .

the former vice president , 87 , was treated for cold and flu symptoms .

**Summary generated by our model:**

the former vice president was released on saturday after being admitted with influenza .

the 87 - year - old was diagnosed with a routine checkup following a fever .

he was also on a nobel peace prize forum in minneapolis .

# Chapter 6

# CONCLUSION AND SUGGESTIONS

# 6.1 Conclusion

In this project we used a hybrid pointer- generator model to reduce the inaccuracies in generated summaries which were seen in previous models. We also used the coverage mechanism to avoid the problem of repetition of phrases and words.

Even though our model exhibits some abstractive abilities along with correct attention, achieving more abstraction without compromising inaccuracies is yet to be achieved.

# 6.2 Suggestions for Future Work

As already discussed, a better and more accurate model is still needed. The task will become a whole lot easier if we have a more detailed dataset containing multiple reference summaries for each article. This dataset will enable much better training and a better loss function thereby generating a more abstractive model. Also, ROUGE evaluation will yield more reliable results as it will now depend upon comparison with multiple reference summaries.

Also, another work that can be undertaken in the future is to first classify the news articles into different genre and then create a model for each genre. Since, writing styles vary greatly among different genres, this model may give us much better results.

# REFERENCES

# 

1. Abigail See, Peter J. Liu, Christopher D. Manning: Get To The Point: Summarization with Pointer-Generator Networks. (2017)
2. Alexander M Rush, Sumit Chopra, and Jason Weston: A neural attention model for abstractive sen- tence summarization. In *Empirical Methods in Natural Language Processing*. (2015)
3. Chujie Zheng, Kunpeng Zhang, Harry Jiannan Wang, Ling Fan: Topic-Aware Abstractive Text Summarization. (2020)
4. Dima Suleiman and Arafat Awajan: Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges. (2020)
5. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio: Neural machine translation by jointly learning to align and translate. In *International Con- ference on Learning Representations*. (2015)
6. Ekaterina Zolotareva, Tsegaye Misikir Tashu and Tomáš Horváth: Abstractive Text Summarization using Transfer Learning. (2020)
7. Ilya Sutskever, Oriol Vinyals, and Quoc V. Le: Sequence to Sequence Learning with Neural Networks. (2014)
8. Jiatao Gu, Zhengdong Lu, Hang Li, Victor O.K. Li: Incorporating Copying Mechanism in Sequence-to-Sequence Learning. (2016)
9. Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, Phil Blunsom: Teaching Machines to Read and Comprehend. (2015)
10. Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly: Pointer networks in Neural Information Processing Systems. (2015)
11. Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, Bing Xiang: Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond (2016)
12. Ramesh Nallapati, Feifei Zhai, and Bowen Zhou: SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In *Association for the Advancement of Artificial Intelligence*. (2017)
13. Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata: Neural headline generation on abstract meaning representation. In Empirical Methods in Natural Language Processing. (2016)
14. Sumit Chopra, Michael Auli, and Alexander M Rush: Abstractive sentence summarization with at- tentive recurrent neural networks: In *North American Chapter of the Association for Computational Linguistics*. (2016)
15. Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, Chandan K. Reddy: Neural Abstractive Text Summarization with Sequence-to-Sequence Models. (2018)
16. Tiezheng Yu, Zihan Liu, Pascale Fung: “AdaptSum” Towards Low-Resource Domain Adaptation for Abstractive Summarization. (2021)
17. Vrinda Vasavada, Alexandre Bucquet: Just News It: Abstractive Text Summarization with a Pointer-Generator Transformer. (2019)
18. Xin Wan, Chen Li, Ruijia Wang, Ding Xiao, and Chuan Shi: Abstractive Document Summarization via Bidirectional Decoder. (2018)
19. Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, Hang Li: Modeling Coverage for Neural Machine Translation. (2016)
20. Dataset source: [www.tensorflow.org/datasets/catalog/cnn\_dailymail](http://www.tensorflow.org/datasets/catalog/cnn_dailymail)