



TEXT SUMMARIZATION ON COVID 19 NEWS

MR. CHATCHAWARN LIMPLOYPIPAT


A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(COMPUTER ENGINEERING)
FACULTY OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI
2021

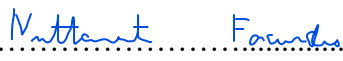
Text Summarization on Covid 19 News

MR Chatchawarn Limploypipat B.Science. (Computer Science)


A Thesis Submitted in Partial Fulfillment
of the Requirements for
The Degree of Master of Science (Computer Engineering)
Faculty of Engineering
King Mongkut's University of Technology Thonburi
2021


Thesis Committee


..... Chairman of Thesis Committee
(Assoc. Prof. Charnchai Pluempittiwiriyawej, Ph.D.)


..... Member and Thesis Advisor
(Asst. Prof. Nuttanart Facundes, Ph.D.)


..... Member
(Asst. Prof. Jumpol Polvichai, Ph.D.)


..... Member
(Assoc. Prof. Naruemon Wattanapongsakorn, Ph.D.)


..... Member
(Assoc. Prof. Jonathan H. Chan, Ph.D.)

Thesis Title	Text Summarization on Covid 19 News
Thesis Credits	12
Candidate	Mr. Chatchawarn Limploypipat
Thesis Advisor	Asst. Prof. Dr. Nuttanart Facundes
Program	Master of Science
Field of Study	Computer Engineering
Department	Computer Engineering
Faculty	Engineering
Academic Year	2021

Abstract

Since the spread of Coronavirus disease or Covid-19 at the end of 2019, there has been an extensive amount of news about Covid-19, and it takes a long time for humans to read the news, process it and retrieve important information from it. Therefore, automatic text summarization is necessary in this matter as it can help us process information faster and use it to make better decisions. Currently, there are two main approaches to automatic text summarization: extractive and abstractive. Extractive text summarization is conducted by identifying important parts of the text and extracting a subset of sentences from the original text. Abstractive text summarization is closer to the human method as it is the reproduction or rephrasing based on interpretation and understanding of the text using natural language processing techniques. In this research, we applied an abstractive summarization method on a specific dataset, namely, Canadian Broadcasting Corporation (CBC) news dataset about Covid-19 news. Data augmentation was also exploited in the pre-processing part to be an example case of working with data that are not perfect or diverse enough. Two Long Short Term Memory (LSTM) models, with and without data augmentation, were used to generate summaries. The resulting summaries were analyzed and compared with related work using Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics.

Keywords: abstractive text summarization/ coronavirus/ COVID-19/ natural language processing

หัวข้อวิทยานิพนธ์ การประมวลผลสรุปข้อความจากข่าวโรคโควิด19

หน่วยกิต 12

ผู้เขียน นายชัชวาลย์ ลิมพลอยพิพัฒน์

อาจารย์ที่ปรึกษา ผศ.ดร.ณัฐนาถ ฟาคุนเค็ช

หลักสูตร วิทยาศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะ วิศวกรรมศาสตร์

ปีการศึกษา 2564

บทคัดย่อ

เนื่องด้วยการระบาดของโรคโคโรนาไวรัสหรือเรียกอีกอย่างว่าโรคโควิด 19 ที่เกิดขึ้นตั้งแต่ปีพ.ศ. 2562 หลังจากนั้น ได้มีการนำเสนอข่าวสารข้อมูลต่างๆ เกี่ยวกับโรคโควิด 19 เป็นจำนวนมาก ทำให้ผู้ที่รับข่าวสาร ต้องใช้เวลาในการประมวลผลของข้อมูลข่าวสารเหล่านี้เพื่อที่จะเข้าใจในเนื้อหาและข้อมูลที่ต้องการ ดังนั้นการใช้ระบบสรุปข้อความอัตโนมัติมีความสำคัญในการช่วยให้ผู้อ่านสามารถประมวลผลข้อมูลได้เร็วขึ้นและทำให้มีการตัดสินใจที่ดีขึ้น ระบบสรุปข้อความอัตโนมัติมี 2 ประเภทนั่นคือ 1.การแบ่งข่าวออกเป็นข้อความย่อยๆ จากนั้นหาข้อความที่สำคัญจากบทความเหล่านั้นนำมาเป็นบทสรุป เรียกว่า Extractive text summarization 2.วิธีนี้เป็นวิธีที่ใกล้เคียงกับวิธีของมนุษย์ที่สุด ซึ่งจะเป็นการนำเอาข่าวมาประมวลผลข้อมูลใหม่ แล้วทำการสร้างคำใหม่ๆ โดยไม่ได้นำเอาคำศัพท์ที่มาจากข่าวที่เอามาประมวลผลโดยใช้วิธีการประมวลผลภาษาธรรมชาติ หรือที่เรียกว่า Abstractive text summarization ในการวิจัยครั้งนี้เราได้้นำเอาวิธี Abstractive text summarization มาสรุปข่าวจากฐานข้อมูลของ Canadian Broadcasting Corporation (CBC) ที่เกี่ยวกับโรคโควิด 19 แล้วยังมีการใช้ Data augmentation ในช่วงก่อนทำการจัดการข้อความประมวลผล มาเป็นตัวอย่างในกรณีที่ข้อมูลมีไม่มากพอ อีกทั้งได้ทำการแบ่ง LSTM โมเดล ที่ใช้ในการสรุปข่าว ออกเป็นสองโมเดลเพื่อเปรียบเทียบระหว่างโมเดลปกติ กับโมเดลที่ใช้ Data augmentation และโมเดลจากงานที่ใกล้เคียงในการสรุปข้อความข่าว โดยใช้เกณฑ์จาก Recall-Oriented Understudy for Gisting Evaluation (ROUGE) ในการประเมิน

คำสำคัญ: การขอลความ/ โรคโคโรน่าไวรัส/ การประมวผลภาษาธรรมชาติ

ACKNOWLEDGEMENTS

I would like to thank my advisor Asst. Prof. Nuttanart Facundes who gave me the advice and guideline for this thesis. Also, I would like to thank Asst. Prof. Jumpol Polvichai, Assoc. Prof. Naruemon Wattanapongsakorn, Assoc. Prof. Charnchai Pluempittiwiriyawej, and Assoc. Prof. Jonathan H Chan for being on my committee. Moreover, I would like to thank all my teachers and classmates for being nice and helpful to me.

CONTENTS

	PAGE
ENGLISH ABSTRACT	ii
THAI ABSTRACT	iv
ACKNOWLEDGEMENTS	vi
CONTENTS	vii
LIST OF TABLES	x
LIST OF FIGURES	xii
LIST OF SYMBOLS	xiv
 CHAPTER	
1. INTRODUCTION	1
1.1 Statement of Problem	1
1.2 Objectives	1
1.3 Scopes	2
1.4 Expected Benefit	2
1.5 Thesis Organization	2
 2. RELATED LITERATURE	3
2.1 Text Summarization	3
2.2 Recurrent Neural Network	4
2.3 Encoder-Decoder Architecture and Transformers	6

CONTENTS

	PAGE
2.4 Attention Mechanism	7
2.5 Data Augmentation	9
2.6 Recall-Oriented Understudy for Gisting Evaluation (ROUGE) Framework	10
 3. METHODOLOGY	 12
3.1 Overview of the Methodology	12
3.2 Data Collection	12
3.3 Text Preprocessing	14
3.4 Data Augmentation	15
3.5 Summary Generation	16
3.6 Evaluation of the Result	18
 4. EXPERIMENTAL RESULTS	 19
4.1 Data Augmentation	19
4.2 Model Training	20
4.3 Summary Generation and Comparison	25
4.4 ROUGE Framework	27
4.5 Comparative with MDGT	28
 5. CONCLUSION AND FUTURE WORK	 30
 REFERENCES	 31

CONTENTS

	PAGE
APPENDIX	35
A The Loss Score	35
B The Time per Iteration	38
C The Sample of ROUGE Score	41
CURRICULUM VITAE	47

LIST OF TABLES

TABLE		PAGE
2.1	Examples of Rule-base augmentation	9
2.2	Examples of MixUp augmentation	10
3.1	Data augmentation generates a new sentence from the original dataset	16
4.1	Example of random insertion compare to the original sentence	19
4.2	Example of random swapping compare to the original sentence	19
4.3	Example of random deletion compare to the original sentence	19
4.4	The model parameter setting	20
4.5	The model summary without the dataset that use the data augmentation	21
4.6	The model summary of the dataset that uses the data augmentation	22
4.7	Generated summary compared with human summary from the original text	27
4.8	The ROUGE scores for the model with data augmentation	27
4.9	The ROUGE scores for the model without data augmentation	28
4.10	The comparison between the MTDGT model result and the model result	28
A.1	The loss score for the modal	35
B.1	The time per each iteration for each model	38
C.1	ROUGE scores for the dataset without data augmentation	41

LIST OF TABLES

TABLE		PAGE
C.2	ROUGE scores for the dataset with data augmentation	44

LIST OF FIGURES

FIGURE	PAGE
2.1 Abstractive text summarization	4
2.2 A recurrent neural network	5
2.3 An unfolded recurrent neural network	5
2.4 The repeating module in a LSTM contains four interacting layers	6
2.5 The Encoder-Decoder architecture	7
2.6 The attention mechanism focusses a word in the sentence	7
2.7 The attention mechanism for the text summary	8
2.8 The architect of Bahdannau's attention	9
3.1 CBC News articles about COVID-19	13
3.2 Word cloud in January, 2020	13
3.3 Word cloud on March to April, 2020	14
3.4 Text preprocessing diagram	14
3.5 Stop word removal	15
3.6 Stemming method	15
3.7 The model diagram for creating summary	17
3.8 The summarized model layer flows	18
4.1 The model architecture	21
4.2 The Loss score for a model that does not use the data augmentation	23
4.3 The Loss score for a model that does use the data augmentation	24
4.4 The time for each iteration of model that does not use the data augmentation	25

LIST OF FIGURES

FIGURE		PAGE
4.5	The time for each iteration of model that does not use the data augmentation	25
4.6	The original text input	26

LIST OF SYMBOLS**SYMBOL****UNIT**

N_{single}	Number of ROUGE metrics
r_i	the sentences in the reference document

CHAPTER 1 INTRODUCTION

Natural Language Processing (NLP) is the computer engineering of machines to process and understand human languages in order to perform NLP tasks. It is the core of many applications nowadays including machine translation, natural language understanding, information extraction, text classification and text summarization. As for automatic text summarization, it is the task of generating concise summaries from voluminous texts for easy understanding and further information processing. For automatic text summarization which is the focus of this research, there are two main methods: Extractive and Abstractive text summarization. Extractive text summarization involves the selection of important contents of a text document. Abstractive text summarization applies rules that generate a summary not based on the words in the original document but based on its content. In this research, we will focus on the Abstractive text summarization which we consider to be close to how humans make summaries.

The domain of our research is on the news about Coronavirus or Covid-19. Covid-19 is a disease that originated in Wuhan, China in December 2019. Since then, it became a global pandemic and people want to have knowledge about its symptoms, preventive measures, vaccines, etc. and they follow the news for such knowledge or information. However, there are many news articles to process, which are too long and include many technical words that make them hard to be understood. Automatic text summarization is, thus, useful and can contribute to information processing and decision making.

1.1 Statement of Problem

This research focuses on automatic text summarization using the abstractive method, on the dataset of coronavirus news. The research aims at answering the following questions.

- 1.1.1. How to apply the abstractive summarization method to automatically summarize the dataset of coronavirus news?
- 1.1.2. How to apply the data augmentation in text preprocessing?
- 1.1.3. How to evaluate the results of the summarized texts?
- 1.1.4. How to apply NLP metrics for evaluation?

1.2 Objectives

The objective of this project research is to implement the Abstractive Text Summarization and generate summaries from the dataset of Coronavirus news. The objectives are as follows:

- 1.2.1. Study and review text summarization methods.
- 1.2.2. Study about the neural network like Recurrent Neural Network (RNN) i.e. Long Short-Term Memory model (LSTM), encoder-decoder model and their applications.
- 1.2.3. Study and apply abstractive text summarization methods on COVID19 news as well as augmenting the datasets in preprocessing steps.
- 1.2.4. Evaluate results using NLP metrics.

1.3 Scopes

The scope of this research to apply abstractive method for text summarization on Covid-19 using Canadian Broadcasting Corporation or CBC news dataset and can be further defined as follows:

- 1.3.1. The implementation of this research is conducted in Python
- 1.3.2. The evaluation of text summaries results is done using a set of metrics of Recall-Oriented Understudy for Gisting Evaluation (ROUGE) framework as well as comparison with the related work.

1.4 Expected Benefit

The expected benefit of this research project is that the outcome is obtained in the following orders.

- 1.4.1. Study and implementation of Abstractive Text Summarization.
- 1.4.2. Study and research on neural network and its classes.
- 1.4.3. Apply NLP metrics for evaluating automatic text summarization task.

1.5 Thesis Organization

This thesis is organized as follows: Chapter 1 introduces the problem, research questions, and objectives of this research. In Chapter 2, Literature Review and related research are discussed. Chapter 3 describes the methodology and processes of the research. Chapter 4 presents the experiment results and evaluation. Chapter 5 presents our conclusion and discusses the possible directions for future work.

CHAPTER 2 BACKGROUND AND LITERATURE REVIEW

This chapter provides background and related work on automatic text summarization in the following topics: (1) Text summarization: extractive and abstractive methods, (2) Recurrent Neural Network and Long Short Term Memory, (3) Encoder-Decoder Architecture and Transformer, (4) Attention mechanism and (5) Recall-Oriented Understudy for Gisting Evaluation (ROUGE) framework.

2.1 Text Summarization

Text summarization is an important NLP task. It involves shortening original texts while still preserving the meaning. There are two approaches to text summarization: extractive and abstractive methods.

2.1.1 Extractive Text Summarization

Several researchers apply the extractive text summarization method as in [4]. Extractive summarization is the method that selects information from the original text by ranking or scoring sentences and generating a summary. Sentence scoring solves the problem of deciding which sentences to include in the summary by including the sentences with the highest scores. Another method is to use Term Frequency-Inverse Document Frequency Method (TF-IDF).

Term Frequency (TF) and Inverse Document Frequency Methods (IDF) are statisticals that show how the importance of a word given in a document. This is done by multiplying two metrics: how many times a word appears in a document (TF), and the inverse document frequency of the word across a set of documents (IDF). After getting the scores of TF and IDF, sentences with high TF-IDF scores will be used to generate a summary. It should be noted that the problem with TF-IDF is that sometimes longer sentences have high scores simply due to the fact that they contain more words.

2.1.2 Abstractive Text Summarization

Abstractive text summarization [5] is the summarization that is close to how a human summarizes a document. First, a document is processed and a summary is generated using new words, sentences, and rephrasing in Figure 2.1. The abstractive text summarization was mostly applied by using deep learning or machine learning such as a long short-term memory (LSTM) [6]. However, this method could take a long time to process because it includes lots of methods such as content preprocessing, word embedding, fundamental model design, discourse rules, etc. [7].

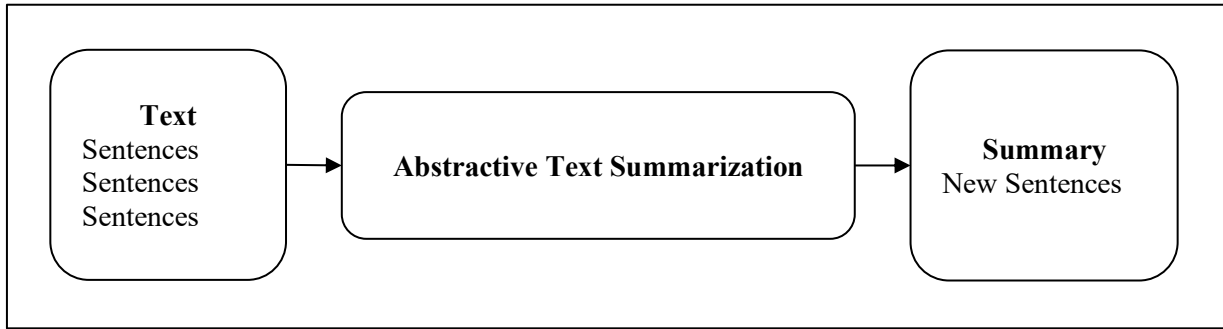


Figure 2.1 Abstractive text summarization

Talukder *et al.*[7] reviewed and compared the technique to summarize texts based on the abstractive text summarization such as Word Graph Methodology, Semantic Graph Reduction Algorithm, and Markov Clustering Principle.

Another research using the abstractive text summarization was Logeesan *et al.* [8] who applied the abstractive text summarization on stock market news articles. [8] uses keyword-based weighting to extract the sentences and graph algorithms to analyze the relatedness among the sentences. Each sentence was represented as a node in the graph to show the relationship between the sentences in order to create a summary of the document that contained only the significant contents.

2.2 Recurrent Neural Network

Recurrent Neural Network (RNN)[9] is a type of artificial neural network which uses sequential data or time series data and can be used with text processing tasks including text summarization. RNN is essentially a fully connected neural network that contains a refactoring of some of its layers into a loop. That loop is typically an iteration over the addition or concatenation of two inputs, a matrix multiplication and non-linear functions as in Figure 2.2. The concept of RNN is to introduce the memory in neural networks to their feedback, and these networks can learn information based on the context in which the output generated from the previous move which is used as input for the current move in order to prevent losing information during the process. Also, RNN provides a solution by using a hidden layer. Figure 2.3 shows the depiction of RNN being unfolded into a network. For the input sequence of n words, the network would be separated into an n -layered neural network in which a layer denoted each word. For example, if we consider the language model that tries to predict the next word in the previous sentence such as trying a prediction of the last word in “there are crowds on the ...,” we do not need any future context. So, the next word is likely be “street”. It means that there is a small gap between the relevant information, and RNN can learn from the past information. However, the sentence may have more contexts that need to pass through the next word such as “I would really love rose and ... but I hate Mac’n cheese.” In this information the next word is the name of thing that is loved but the following information is about something being hated so the gap between the relevant information becomes very large. As the gap expands, RNN becomes unable to learn to connect the information.

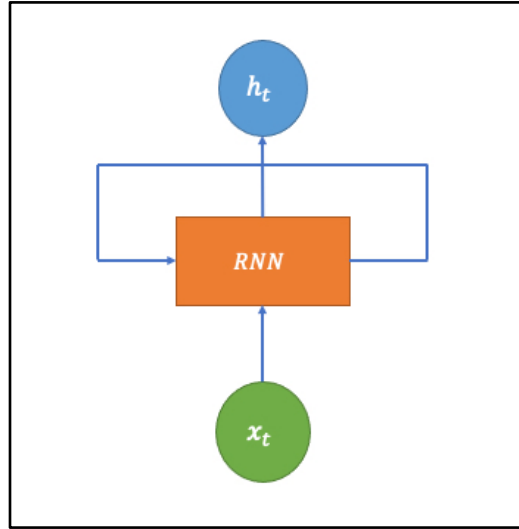


Figure 2.2 A recurrent neural network [9]

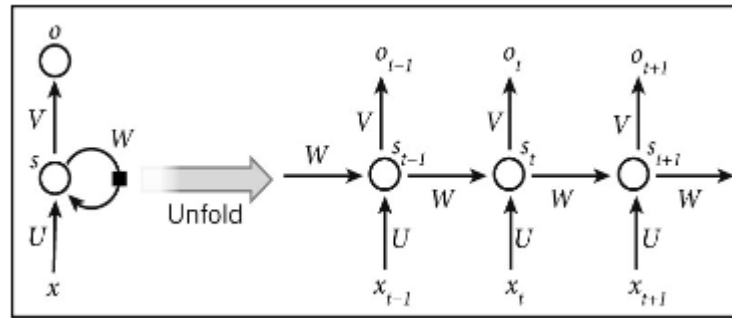


Figure 2.3 An unfolded recurrent neural network [5]

2.2.1 Long Short-Term Memory

Long short-term memory (LSTM)[9,10] is a unit of RNN that can identify and remember the data pattern for a certain period. Moreover, LSTM can solve the problem of gradient vanishing in the RNN training process, which uses Back-Propagation Through Time. LSTM can also lead to many successful runs and learn faster. LSTM is achieved by an efficient, gradient-based algorithm for an architecture. Figure 4 shows the architecture of an LSTM cell that includes the forget gate, input gate, and an output gate while working using the sigmoid activation function. The forget gate decides which information to keep or forget. The input gate updates the state of the cell and the output gate decides what would be the next hidden state. It is composed by multiplicative gate units which are able to be opened and closed for learning constant error flow. For example, in order to predict the next word of “I would really love rose and ... but I hate Mac’n cheese.” that needs to contain the context of loved things and hated things and the LSTM has three gates that can control this information. The forget gate can thus prevent overload information in the LSTM and the input

gate decides which word to update or replace by a new word. Finally, the output gate decides the output of LSTM. The LSTM and Recurrent Neural Network Language Model (RNNLM)[11] can be used to compose word attention and sentence attention for language modeling such as BoydCut[12], an NLP framework for identifying sentence boundaries based on Bidirectional LSTM-CNN Model, which was applied for Thai sentence segmentation. Moreover, the BoydCut framework can utilize words, characters, and parts of speech (POS) to create new sentences. Another related work is, Hayatin *et al.*[15] applied abstractive text summarization to COVID-19 news and also used the LSTM to generate the summaries based on transformer architecture.

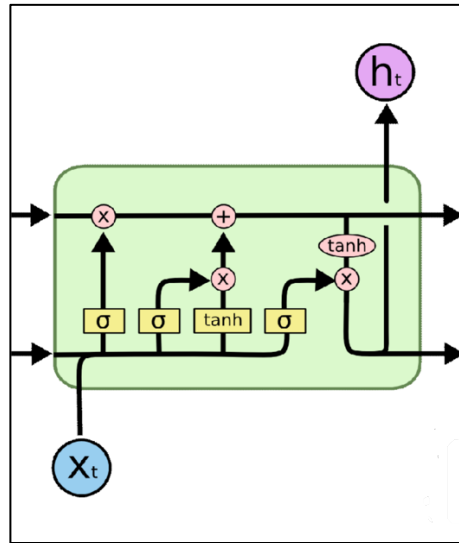


Figure 2.4 The repeating module in a LSTM contains four interacting layers [5]

2.3 Encoder-Decoder Architecture and Transformer

The Encoder-Decoder Architecture [20] is a way of organizing recurrent neural networks for sequence prediction problems such as machine translation and text summarization. The encoder reads the input sequence and summarizes the information as the internal state vectors and then generates the context vector which will be passed to the decoder as input. The decoder generates an output sequence based on the context vector as shown in Figure 2.5 shows an example used in the machine translation task which has a major problem with the variable-length sequences. A way to handle the input and output problem are the design of two major components.

Another deep learning model is discussed here which is Transformer. A transformer, similar to RNN, is designed to process sequential input data, such as natural language. However, unlike RNN, the transformer processes the entire input all at once. An example is Bidirectional Encoder Representations from Transformers (BERT) [22] which are bi-directional, pre-trained language models that have been employed as encoders for natural language understanding tasks. The pre-trained model of BERT can also be used for many NLP tasks including text summarization.

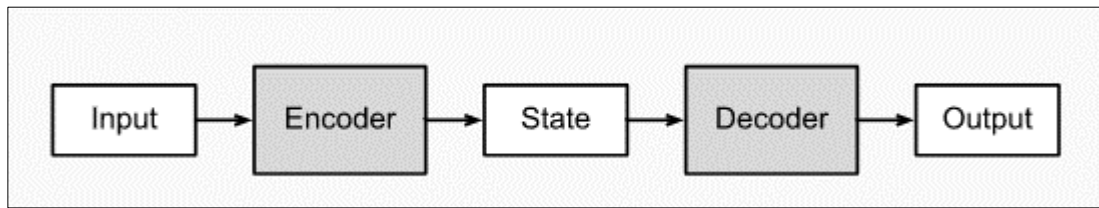


Figure 2.5 The Encoder-Decoder architecture

2.4 Attention Mechanism

The attention mechanism as explained in [23] used in this research is based on the concept of the attention mechanism applied in the sequence-to-sequence model. The attention mechanism has to be used with the deep learning architecture known as alignment models.

As for attention, it is used for humans to focus on the importance of something such as a sentence. If we pay attention to the sentence, we can understand the information from the sentence. For deep learning tasks, an attention mechanism is applied to the model that can pay attention to certain factors during processing data because the context data is important while processing the text input as shown in Figure 2.6. Moreover, the attention mechanism could help the deep learning approach to learn the long-range data with the context vector that contains the information in every time step and has the important information of each input corresponding to the input as shown in Figure 2.7. This helps the model to overcome the vanishing gradient problem and also could help the model run faster compared to the model that does not use the attention mechanism.

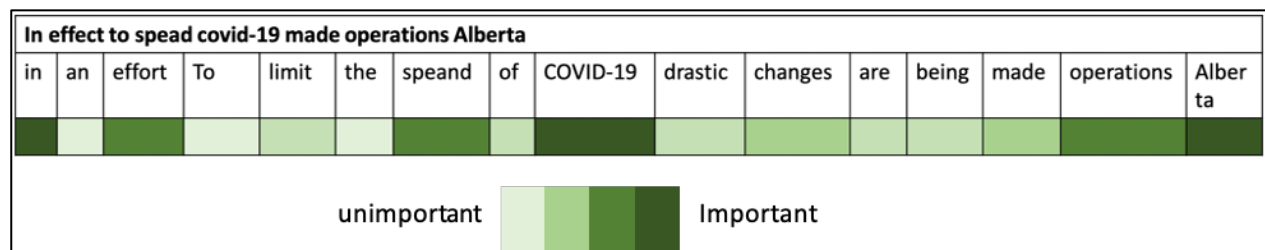


Figure 2.6 The attention mechanism focuses on a word in the sentence

	unimportant					Important				
	<i>northern</i>	<i>health</i>	<i>authority</i>	<i>warns</i>	<i>of</i>	<i>limited</i>	<i>for</i>	<i>workers</i>	<i>in</i>	<i>country</i>
in										
an										
effort										
to										
limited										
...										
country										

Figure 2.7 The attention mechanism for the text summary

For this research, we attempt to use the attention mechanism layer in the deep learning model to improve the model in generating a summary by keeping the important information correspondingly, as well as improving the speed and ascertain the correctness of the input.

2.4.1 Bahdanau et al.'s Attention Mechanism

The attention mechanism was first proposed by Bahdanau *et al.*[24] who used the sequence-to-sequence model by adding the alignment scores that were computed from the information of all hidden states of the encoders and the information of the previous state of the decoder which allows the model to find relationships between tokens.

After finding that the sequence-to-sequence model had a problem of the bottleneck for the encoding part that squeeze the information to a fixed-length that cause the performance of the sequence-to-sequence model to drop the length of the input sequence, [24] proposed a mechanism that allows the language model to automatically soft search for a part of an encoding related to the decoder by looking at the information both forward and backward for hidden state. Then, the context vector was sent to compute the attention score call Bi-directional RNN as shown in Figure 2.8.

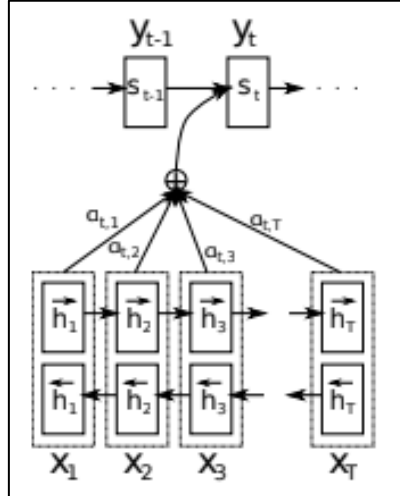


Figure 2.8 The architect of Bahdanau's attention [24]

2.5 Data Augmentation

Data Augmentation [16] (DA) is a set of algorithms that constructs new data from the available dataset by using the strategy that increases the diversity of the dataset. In some cases, the datasets can be limited due to various reasons such as data sparsity. Furthermore, when the data is changed and augmented, this can increase the robustness, DA is widely used for machine learning areas such as computer visualization, especially classification tasks, e.g. image classification, and semantic classification. Mostly, DA is used for preventing overfitting and for NLP tasks by considering the reusable tokens and shuffling them as a new dataset. DA has 4 types based on the symbolic or neural methods.

2.5.1 Rule-based Augmentation

Rule-based augmentation constructs rules from augmented examples by inserting new words and rearranging words from the existing data. Rule-based augmentation consists of 3 methods that are processed at the token level. First, the random swapping which randomly selects one word in the sentence and randomly pushes the selected word into a new position of a word in the sentence. Second, the random deletion which randomizes a word in the sentence and then removes a selected word, Finally, the random insertion which a new word and introduces to a sentence in a random position.

Table 2.1 Examples of Rule-based augmentation

Rule-based augmentation	Example	Example with Rule-based augmentation
Random swapping	I am running	I like running
Random deletion	I am running	I running
Random insertion	I am running	I am joy running

Table 2.1 shows how the sentences would be for the result for three methods of the Rule-based augmentation. This is the strategy that is processed at the token level of the document in order to increase the samples of the document in the simplest way.

2.5.2 Graph-structured Augmentation

Graph-structured augmentation consists of relations and encodings for knowledge graphs, grammatical structures in syntax trees. For example, if we want to replace the word “lion”, the graph-structured augmentation will know the word that has the relation to the “lion” node such as “tiger”, or “jaguar”. In addition, there are kinds of graphs that have been developed with notables such as WordNet [17], Penn Treebank [18].

Jiajun *et al.*[19] applied the Graph-structured augmentation for classification to identify the category labels of graphs with limitations of the dataset that can easily lead to the overfitting. [9] exploited Graph-structured augmentation, using a generic model evolution framework called M-Evolve. they have used M-Evolve, a generic model evolution framework, which combines Graph-structured augmentation, data filtration and model retraining. For the experiment, M-Evolve could help prevent overfitting in the graph and yielded the improvement of 3-12% of accuracy on the classification tasks.

2.5.3 MixUp Augmentation

MixUp augmentation is a meshing the existing examples together, MixUp can be done by combining of a half of a sentence with half another sentence and concatenate it with half of another sentence in order to create a new example in Table 2.2.

Table 2.2 Examples of MixUp augmentation

Original sentences	MixUp augmentation
I want to buy some snacks and drinks. I cross the street when the green light turns on.	I cross the street to buy some snacks and drinks. I want to buy it when the green light turns on.

2.5.4 Feature Space Augmentation

Feature Space augmentation describes augmenting data that considers the intermediate representation space of Deep Neural Networks in which the input data is transformed into task-specific predictions. Also, Feature Space augmentation isolates the features and applies the noise to the new data. Moreover, the noise can be sampled from the standard uniform or gaussian distractions.

2.6 Recall-Oriented Understudy for Gisting Evaluation Framework (ROUGE)

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) framework [13,14] is a set of metrics and software packages used for evaluating natural language processing tasks such as summarization. Also, it includes several automatic evaluation methods that measure the similarity

between summaries. This is done by benchmarking a summary with one reference summary produced by humans using precision (P) as (2.2), the ratio of overlapped word over system and recall (R) as (2.1). As for F-score which is the statistical measurement combining precision and recall as (2.3) with β value that needs to be created to compute, the ratio of overlapped words over the reference summary, in the context. It means that the ROUGE framework is evaluates the summary by comparing the reference summary and generated summary in order to show how the generated summary is close to the reference summary. The ROUGE framework also has different kinds of measurement such as ROUGE-N, ROUGE-L and ROUGE-S

$$Recall(R) = \frac{\text{number of overlapping words}}{\text{total words in reference summary}} \quad (2.1)$$

$$Precision(P) = \frac{\text{number of overlapping words}}{\text{total words in generated summary}} \quad (2.2)$$

$$F = \frac{(1+\beta^2).P.R}{P+\beta^2R} \quad (2.3)$$

2.6.1 ROUGE-N

ROUGE-N measures the overlap of n-grams such as unigram, bigram, trigram, and higher-order n-gram between the reference summary and generated summary.in which n represents the length of the n-gram (n). For example, *ROUGE* – 1 is used to evaluate the summary based on the unigram or each word in the summary and also *ROUGE* – 2 evaluates the summary by using the bigrams or a pair of words in the summary as follows:

$$ROUGE - N = \frac{\sum_{S \in \{Ref.Summary\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{Ref.Summary\}} \sum_{gram_n \in S} Count(gram_n)} \quad (2.4)$$

Other research work such as Hayatin *et al.* [15] used ROUGE-1, and ROUGE-2 for evaluating the summary of the COVID-19 news based on transformer architecture and encoder-decoder layers in the model. the best score of evolution for the ROUGE scores was 0.58 and 0.42 respectively.

2.6.2 ROUGE-L

ROUGE-L which is represented as the longest common subsequence (LCS) to evaluate the summary sentences will be treated as a sequence of the maximum length of N-grams. Apart from ROUGE-N and ROUGE-L, there are variants of ROUGH including ROUGE-W which is evaluating the results based on weighted LCS, and ROUGE-S which is based on Skip-bigram.

CHAPTER 3 METHODOLOGY

In this chapter, the methodology is discussed and the process of the experiment is described in detail.

3.1 Overview of the Methodology

The methodology can be divided into the following three steps:

- 1) Data collection: the data collection that we use is “Evolution of CBC news articles during COVID-19” in the Kaggle online database.
- 2) Text Preprocessing with the data augmentation
- 3) Generate the summary: using the abstractive summarization method
- 4) Evaluate the results: using the ROUGE framework.

3.2 Data Collection

Our dataset is from “Evolution of CBC news articles during COVID-19” on Kaggle, the world's largest data science community, which contains news articles related to COVID-19 from CBC news. CBC (Canadian Broadcasting Corporation) is the news publisher in Canada. The dataset contains a total of 6,786 records since December 2020. Figure 3.1 shows the example records from the Kaggle. Each record consists of columns of authors who publish articles on the internet, the title of each article, date of publication, time that articles were published, details of the title, and contents of articles. Some of the articles include symbols and the URL as references in comma-separated values (CSV) format. In addition, about 93% of news articles, of them contain approximately 800 words while the summary of each contains approximately 15 words.

Also, this collection of data can be plotted as a word cloud only in January, 2020 with 117,666 words as Figure 3.1 shows. China appears the biggest one because at that time most of the news articles contained the word China. Due to the limitation of the amount of data that can be processed on the computer, we consider the period of March and April 2020 as this period containing the word ‘COVID’ most frequently as in Figure 3.2 and it consists of approximately 3,000 records.

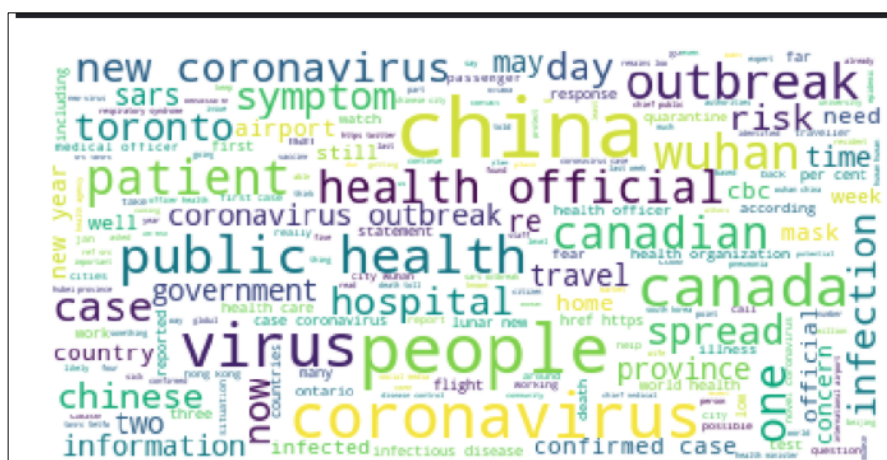


Figure 3.2 Word cloud in January, 2020

3.3.1 Tokenizing

Tokenizing is the method that separates a piece of text into smaller units called tokens. Tokens can be words, characters, or sub words. Broadly, tokenization is classified into 3 types such as word, character and sub word (n-gram) tokenization. For example, “I love hamburgers.” The tokenization can be separated into I-love-hamburgers. Moreover, tokenization can be done by character tokenization like h-a-m-b-u-r-g-e-r-s or by sub word tokenization like hamburger-s.

3.3.2 Stop Word Removal

Stop word is a commonly used word like “a, an, the, in” that a search engine ignores for a better result of retrieving information. Also, the stop word would not be needed for computation because it will take more computing time. So, the reason is that we have to remove the stop word as in Figure 3.5 which is to remove “an” as a stop word.

I love an ice cream → “I”, “love”, “ice”, “cream”

Figure 3.5 Stop word removal

3.3.3 Stemming

Stemming is the process of reducing the words to their word stem or root form. In general, the stemming is not identical to the morphological root of the word. It is usually sufficient that related words map to the same stem. Moreover, the advantages of the stemming are making a word to be in a simple form, faster to compute than using the original form, and easy to handle exceptions. In addition, the stemming has many algorithms in order to transform the origin word into root word, such as prefix-suffix stemming, production technique, etc. Is Figure 3.6 “running”, “runs”, “ran” are the root forms of “run”.

Running, runs, ran → run

Figure 3.6 Stemming method

3.4 Data Augmentation

Data augmentation is the strategy for increasing the diversity of training examples without collecting a new data. Also, the data augmentation can gain more computation efficiency with limited resources. Another profit from the data augmentation is that it can help to prevent the overfitting during the training step. Our selected data augmentation method is rule-based techniques includes random insertion, random deletion, and random swapping methods. By setting the probability that those 3 methods occur in each sentence, each sentence can be preformed by the

3 methods and then inserted as a new row of the training data. Each news record in the dataset can be manipulated by these three techniques. Therefore, from 3,000 records, after data augmentation, the news records become 32,000 records, as illustrated in Table 3.1. Moreover, this data augmentation was done only on the news documents. We did not take the data augmentation on the news's summary rows in order not to augment the summary.

Table 3.1 Data augmentation generates a new sentence from the original dataset.

Original document	Lily Overacker and Laurell Pallot start each gay-straight alliance meeting with everyone introducing themselves, saying their pronouns and sharing highs and lows of the week.
Data augmented by insertion, deletion and swapping	<i>Lily Overacker and Laurell Pallot start each gay-straight alliance meeting with everyone themselves, saying their pronouns and sharing highs and lows of the week.</i>
	<i>Lily Overacker and Laurell Pallot start each gay-straight alliance meeting with everyone introducing themselves, saying their and highs lows of the week.</i>
	<i>Lily Overacker and Laurell Pallot start each gay-straight alliance whatever with everyone introducing themselves, saying their pronouns place sharing highs and lows of a week.</i>
	<i>Lily Overacker it and Laurell Pallot start each gay-straight alliance meeting with everyone introducing themselves, saying their pronouns menage and sharing highs and lows of the week.</i>
	<i>lily Overacker and Laurell Pallot first each gay-straight coalition cope with with everyone introducing themselves, locution their pronouns and sharing highs and lows of the week.</i>
	<i>lily Overacker and Laurell Pallot start each gay-straight alinement meeting with everyone introducing themselves, saying their pronouns and sharing highs and lows of the week.</i>
	<i>Lily Overacker and Laurell want Pallot start each gay-straight alliance meeting with everyone introducing break themselves, saying their pronouns and sharing highs and lows of the week.</i>
	<i>Lily Overacker and Laurell Pallot start each gay-straight alliance meeting with everyone introducing themselves, saying their pronouns and a good deal sharing highs and lows of the week.</i>

3.5 Summary Generation

Figure 3.7 shows how the data flow and how to generate the summary. First, after we gather the data from Kaggle, we separate part of the data to be augmented and the other part is non-augmented. Both parts of data will then be sent to text pre-processing step as mentioned earlier. After that, the texts will be transformed into vectors before sending them to the model summary which includes the encoder, decoder, and Attention layers to generate a summary. This will create the candidate summary. Then, the candidate summary and the reference summary by human will be evaluated using ROUGE. The process is the same for model with data augmentation and without data augmentation.

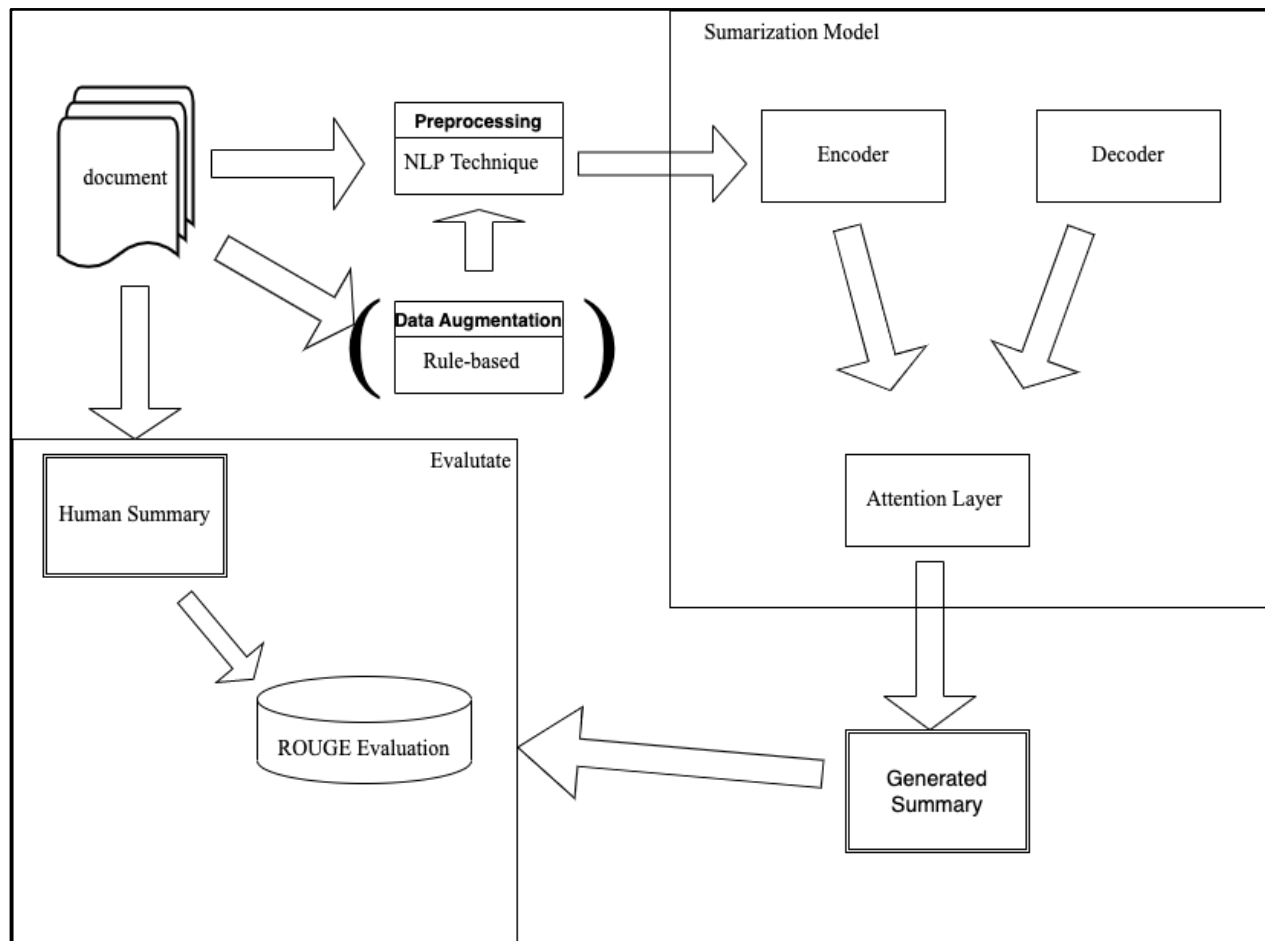


Figure 3.7 The model diagram for creating summary

The Summarization Model as shown in Figure 3.7 consists of the layers such as encoder layer, decoder layer, LSTM layer, and attention layer. All layers working together to generate the summary. For the document that passes the text preprocessing will be a token, and the token is ready to go to the encoder layer. The encoder layer converts the token to a vector for computation. Then the LSTM layer, which is used to compute the prediction vector in the sequence, compute the vector and predict the next vector in a sequence. Also the LSTM layer works with the attention layer in order to select the important vector value. Once the LSTM returns the result, the decoder takes the result vector and converts the vectors into the summary. It should be noted that the attention layer could be active in this layer to return the important word as shown as diagram in Figure 3.8.

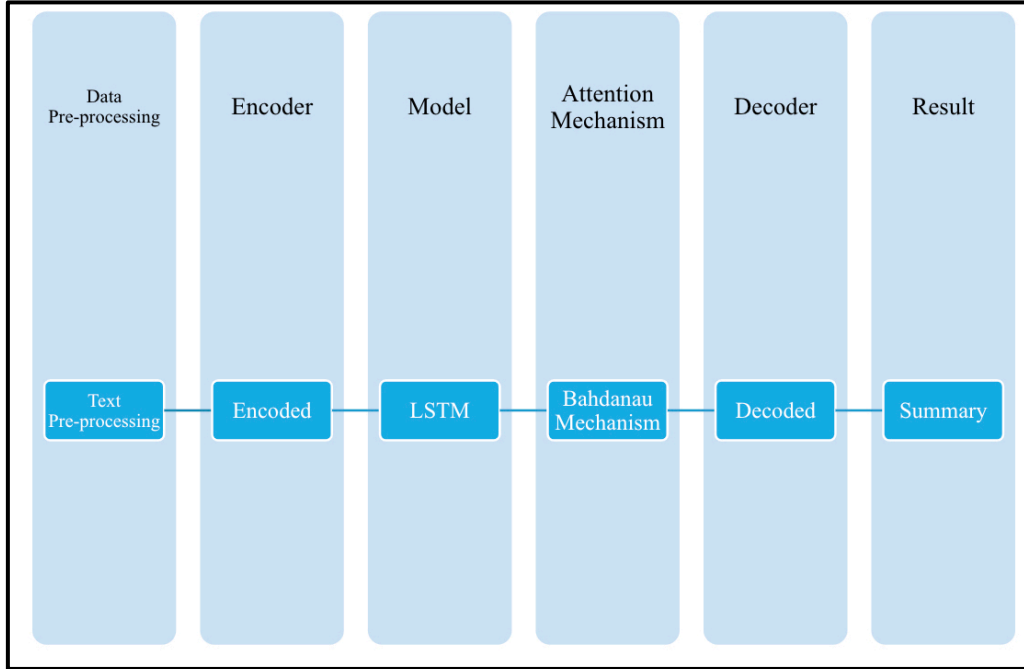


Figure 3.8 The summarized model layer flows

3.6 Evaluation of the Result

The evaluation is conducted by inputting the news that was never used for training and the model will generate the summary of the news. The results acquired from the LSTM model as mentioned above will be evaluated using ROUGE-N metrics which includes ROUGE-1 and ROUGE-2.

ROUGE-N metrics are calculated by:

$$ROUGE - N_{single}(candidate, reference) = \frac{\sum_{r_i \in reference} \sum_{n-gram \in r_i} Count(n-gram, candidate)}{\sum_{r_i \in reference} numNGrams(r_i)} \quad (3.1)$$

where the r_i are the sentences in the reference document, $Count(n-gram, candidate)$ is number of times the specified n-grams occur in the document and $numNGrams(r_i)$ is the number of n-grams of the reference sentence r_i . The results from the test set and real dataset could be compared in order to acquire the score that showed which data indicated the relative of the content of summary against a reference or a set of references (human-produced). We consider using ROUGE-1, ROUGE-2 and ROUGE-L specifically for our evaluation.

CHAPTER 4 EXPERIMENTAL RESULTS

This chapter presents the discussion of data augmentation and results, as well as the ROUGE evaluation of summaries of COVID-19 news, generated using the LSTM and the encoder-decoder architecture with an attention mechanism.

4.1 Data Augmentation

Before the text preprocessing step, we separated part of the data to be augmented using rule-based techniques. For each news article, 9 augmented articles were generated 3 articles from random insertion, 3 articles from random swapping, and 3 articles from random deletion. The sample results of augmented data are shown in Table 4.1-4.3

Table 4.1 Example of random insertion compared to the original sentence

Original sentence	Many couples have rushed to postpone their weddings because of project COVID-19
<i>Random insertion</i>	<i>Many couples have rushed talk of the town atomic number to postpone their weddings because of COVID-19</i>

Table 4.2 Example of random swapping compared to the original sentence

Original sentence	Many couples have rushed to postpone their weddings because of project COVID-19
<i>Random swapping</i>	<i>Many weddings have rushed to postpone their couples because of project COVID-19</i>

Table 4.3 Example of random deletion compared to the original sentence

Original sentence	Many couples have rushed to postpone their weddings because of project COVID-19
<i>Random deletion</i>	<i>Many couples have postponed their weddings because of COVID-19</i>

4.1.1 Dataset

Before model training, the dataset would be sperate into the training data which is training the model for fitting, and testing data for estimating the model accuracy. The dataset that was used for training the model without using the data augmentation has 3,831 records. it would be the training data for 3,031 records and the testing data for 800 records. This applied also for the dataset with augmentation which is 10 times more than the original dataset, hence the total data of 38,310 records. For the training data 37,510 records were used and 800 records were used for testing. For the validation of both models, the full dataset which contains 6,786 records were used for

measurement with ROUGE scores.

4.2 Model Training

For our model, at the first, we tried to use one weight for both types of documents but the feature extraction did not support a different matrix in the same weight parameter in the model. Therefore, the solution was that we made one model with 2 sets of models; one set for the document that did not use the data augmentation and the other for the document that used the data augmentation.

Due to the capacity of the model, we experimented with 3,000 records of CBC news in April, 2020. The dataset from April 2020 constitutes half of the CBC news that was collected and contains high mentions of the COVID-19 term.

There were the layers and the parameters exploited during the training step which include the input layer, LSTM layer, and attention layer as set as parameters as shown in Table 4.4 with a learning rate of 0.001. Moreover, an LSTM unit for every layer have contains 300 units and using the tanh as activation function. It has dropout at 0.4 and recurrent dropout at 0.4 for the encoder. For the decoder, the dropout is 0.4 and recurrent dropout at 0.2. The model architecture is as shown in Figure 4.1. It should be noted that the parameters going through the model are different between the one that uses data augmentation and the other that does not use data augmentation as shown in Table 4.5 and 4.6. There was no further fine-tuning conducted during the training.

Table 4.4 The model parameter setting

Layers	Encoder LSTM	Decoder LSTM	Attention Layer
Node Units	300	300	-
Dropout	0.4	0.4	-
Recurrent Dropout	0.4	0.2	-
Activation Function	tanh	tanh	tanh
Recurrent Function	sigmoid	sigmoid	sigmoid

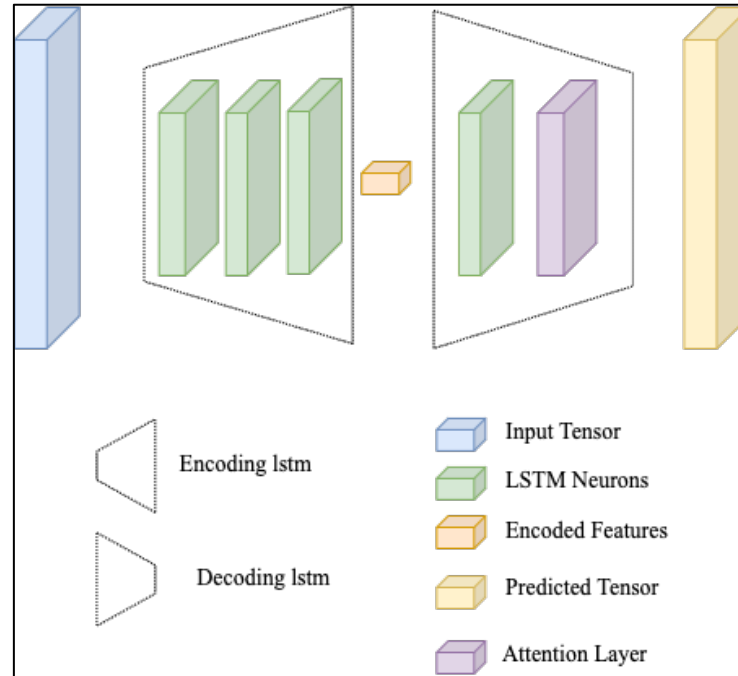


Figure 4.1 The model architecture

Table 4.5 The model summary without the dataset that use the data augmentation.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 80)]	0	[]
embedding (Embedding)	(None, 80, 100)	3350200	['input_1[0][0]']
lstm (LSTM)	[(None, 80, 300), (None, 300), (None, 300)]	481200	['embedding[0][0]']
input_2 (InputLayer)	[(None, None)]	0	[]
lstm_1 (LSTM)	[(None, 80, 300), (None, 300), (None, 300)]	721200	['lstm[0][0]']
embedding_1 (Embedding)	(None, None, 100)	798000	['input_2[0][0]']
lstm_2 (LSTM)	[(None, 80, 300), (None, 300), (None, 300)]	721200	['lstm_1[0][0]']
lstm_3 (LSTM)	[(None, None, 300), [0]', (None, 300), [0]']	481200	['embedding_1[0][0]', 'lstm_2[0][1]',

	300), (None, 300)]		'lstm_2[0][2]'
attention_layer (AttentionLayer)	[(None, None, 300),(None, None, 80)]	180300	['lstm_2[0][0]', 'lstm_3[0][0]']
concat_layer (Concatenate)	(None, None, 600)	0	['lstm_3[0][0]', 'attention_layer[0][0]']
time_distributed_layer (TimeDistributed)	(None, None, 7980)	4795980	['concat_layer[0][0]']
Total params: 11,529,280			
Trainable params: 11,529,280			
Non-trainable params: 0			

Table 4.6 The model summary of the dataset that uses the data augmentation

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 80)]	0	[]
embedding (Embedding)	(None, 80, 100)	3440600	['input_1[0][0]']
lstm (LSTM)	[(None, 80, 300), (None, 300),(None, 300)]	481200	['embedding[0][0]']
input_2 (InputLayer)	[(None, None)]	0	[]
lstm_1 (LSTM)	[(None, 80, 300), (None, 300), (None, 300)]	721200	['lstm[0][0]']
embedding_1 (Embedding)	(None, None, 100)	455000	['input_2[0][0]']
lstm_2 (LSTM)	[(None, 80, 300), (None, 300), (None, 300)]	721200	['lstm_1[0][0]']
lstm_3 (LSTM)	[(None, None, 300),[0],(None, 300), (None, 300)]	481200	['embedding_1[0][0]', 'lstm_2[0][1]', 'lstm_2[0][2]']
attention_layer	[(None, None,	180300	['lstm_2[0][0]',

(AttentionLayer)	300),(None, None, 80)]		'lstm_3[0][0]'
concat_layer (Concatenate)	(None, None, 600)	0	['lstm_3[0][0]', 'attention_layer[0][0]']
time_distributed_layer (TimeDistributed)	(None, None, 4550)	2734550	['concat_layer[0][0]']
Total params: 9,215,250			
Trainable params: 9,215,250			
Non-trainable params: 0			

4.2.2 Loss Score of Model without Using Data Augmentation

In model training, each iteration will return the loss score which indicates the accuracy of the model in each iteration. If the loss is zero then the model's prediction is flawless. The goal of training a model is to find a set of weights and biases that have low loss, on average.

For the model using the weight for the documents without the data augmentation, after training 1,000 iterations, the loss score was the score that was compared to the result of both the testing and training dataset, which is shown in the graph in Figure 4.2.

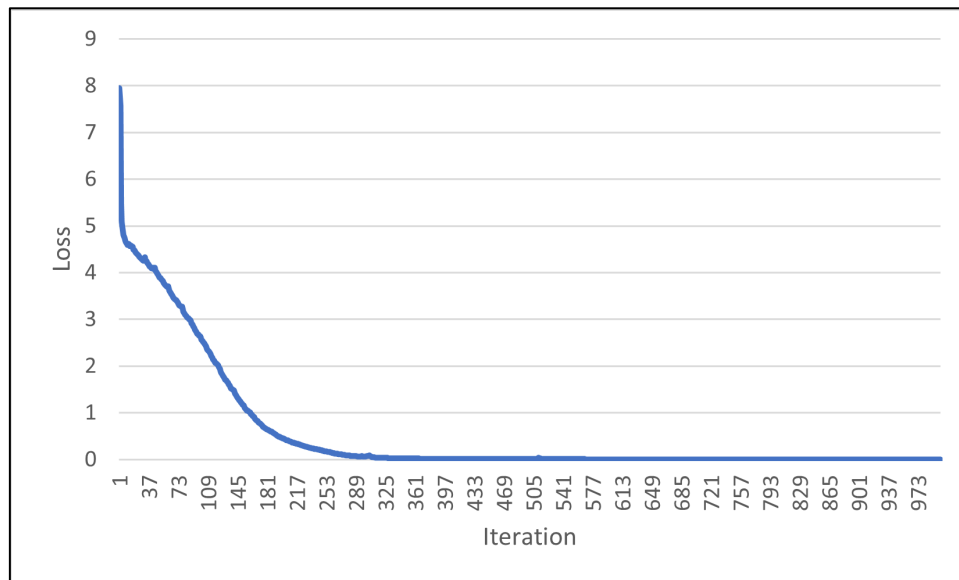


Figure 4.2 The Loss score for a model that does not use the data augmentation.

The loss score was lower after the 35th iteration and close to 0 around the 307th iteration.

Furthermore, after the 1,000th iteration the loss scores still did not reach 0 (for example, it still showed 0.01232134). It meant that we could train the model and set the end of the iteration at 307th, and the loss score would not show the difference between the iteration at 307th and 1,000th iterations.

4.2.3 Loss Score of Model Using Data Augmentation

For the model using the weight for the documents with the data augmentation, after training 1,000 iterations, the loss score was shown as in Figure 4.3.

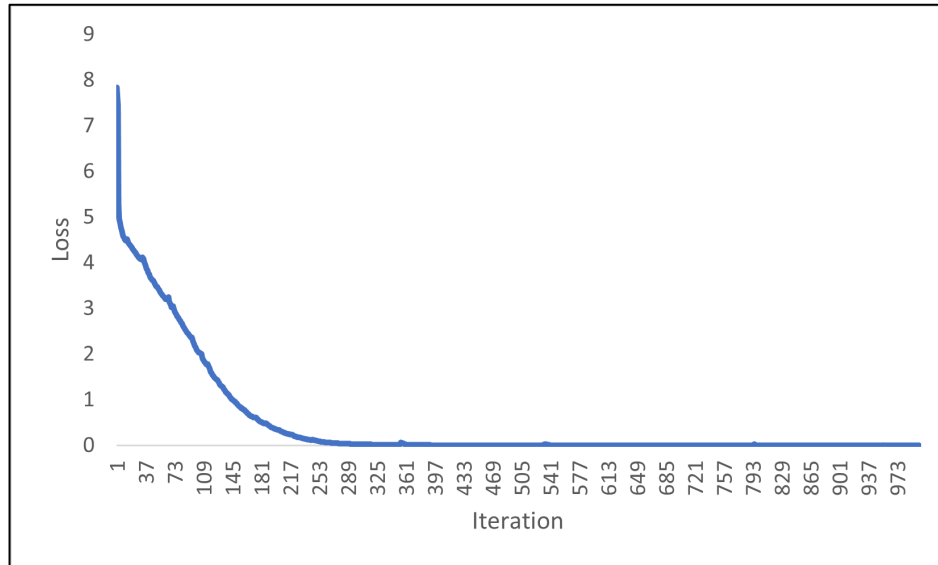


Figure 4.3 The Loss score for a model that uses the data augmentation.

The loss score became lower after the 20th iteration and was close to 0 around the 273th iteration. When we could compare the iterations between two models, the model using the data augmentation could return the loss score close to 0 earlier than the other model.

The difference of each model showed that the loss score of the model that used the weight matrix of the training model for the data augmentation could be reduced faster than that of the model that did not use the data augmentation. Also, for the total time spent on training the model data augmentation is 37,537 seconds and for the model without data augmentation is 40,991 for 1,000 iterations. Therefore, it meant that, with the data augmentation, the time that we spent in training the model could be faster as shown in Figure 4.4 and Figure 4.5.

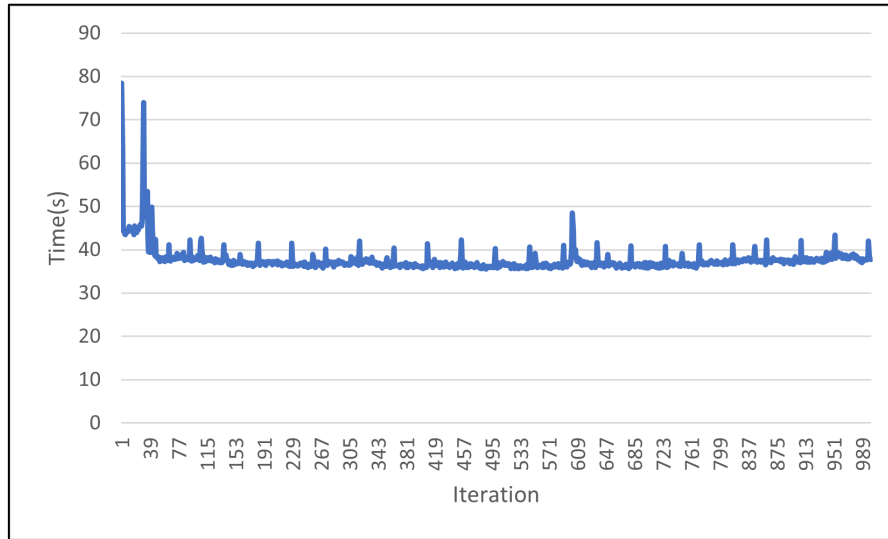


Figure 4.4 The time for each iteration of the model that uses the data augmentation.

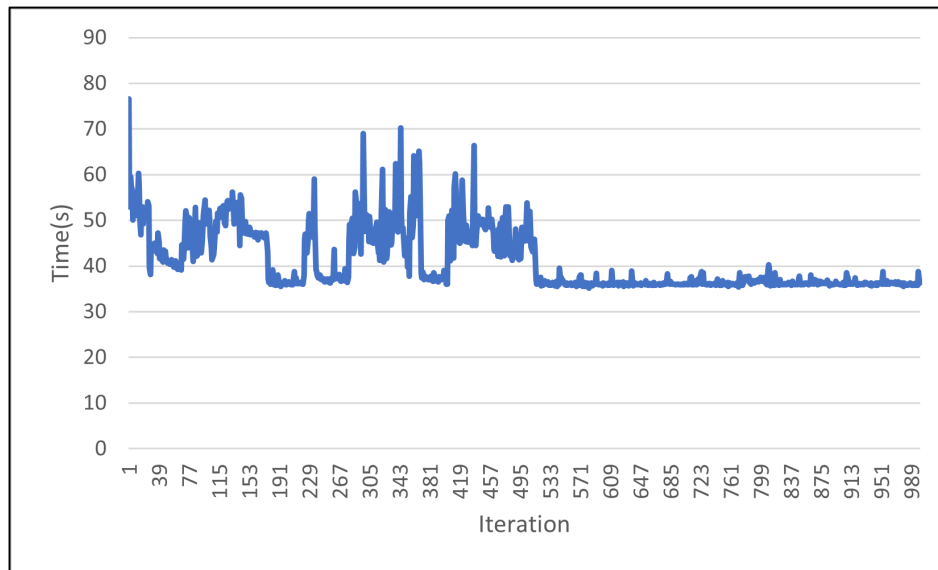


Figure 4.5 The time for each iteration of the model that does not use the data augmentation.

4.3 Summary Generation and Comparison

For the input text as shown in Figure 4.6 below, the model would process and generate the summary shown as in Table 4.7 in comparison with the human summary of the same text.

"If a week is a long time in politics, a year is an eternity. And for Premier Jason Kenney it must seem like a lifetime. A year ago Kenney was celebrating a decisive electoral victory over the NDP after running on a campaign platform that could be summed up in three words: jobs, economy, pipelines. A year later he is dealing with an unprecedented disaster that can be summed up in three words: pandemic, recession, debt. Alberta may circumvent Health Canada to gain access to COVID-19 tests and drugs, premier says Kenney must be looking back at April 16, 2019, with more than a little nostalgia. The year began happily enough for him as he used his summer of repeal, among other things, to fulfil election promises scrapping the provincial carbon tax, cutting corporate taxes, reversing the NDP's environmental strategy, and getting tough with the federal Liberal government as part of a 'fight back' strategy. When the economy continued to sputter and jobs continued to disappear, Kenney maintained course. He cut government spending, made plans to balance the budget, introduced business-friendly legislation, championed more private delivery of publicly funded health care, and set up a "'fair deal'" panel to explore how Alberta could have more control over provincial issues including a police force and pension plan. He was doing pretty much what he had vowed to do during the election campaign. "'Promise made, promise kept,'" became his mantra. Even when his popularity began to slip late last year after Albertans realized the cuts would be deeper than advertised, and thousands of civil service jobs would be lost, Kenney pushed ahead, using his "'historic'" one-million-vote election victory as both shield and weapon against his critics. Finance Minister Travis Toews, left, delivers the 2020 budget the Alberta legislature as Premier Jason Kenney looks on. (Jason Franson/THE CANADIAN PRESS) On February 27, he introduced a provincial budget with so many sunny predictions you needed sunglasses to get through it. Oil prices would stabilize at \$58 USD a barrel. Employment would surge. The provincial deficit would be wiped out in two years. "'There are signs that in 2020 we will turn a corner and see real economic growth return to the Alberta economy,'" declared Finance Minister Travis Toews. We turned a corner, all right. And got mugged in a dark alley by COVID-19 and OPEC+. A week after the budget was tabled, Alberta recorded its first case of the novel coronavirus. Oil prices tumbled after Russia and Saudi Arabia got into a bitter fight over production. A year ago Kenney was on top of the world. Today, he must feel like he's been tossed into the pit. 'Sharp downturn' Kenney says this year's budget deficit could hit an unprecedented \$20 billion. Unemployment could reach 25 per cent. He predicts the price of Western Canadian Select oil could slump below zero later this month. A report from the Conference Board of Canada released Wednesday predicted Alberta's economy will continue to lag behind any global recovery later this year. "'The provincial government is in a tight spot,'" says the report. "'It has, appropriately, been deploying fiscal stimulus to mitigate against the sharp downturn in economic activity. At the same time, it must grapple with the evaporation of billions in royalty revenues on which it depends heavily.'" Alberta economy on track for most severe annual decline on record, national report predicts Kenney is spending billions of dollars, all the while he's losing billions of dollars. He's investing \$1.5 billion in the Keystone XL pipeline this year and putting up \$6 billion in loan guarantees next year. Why Alberta is throwing billions behind the Keystone XL pipeline Ironically, Kenney is on track to become the biggest-borrowing, biggest-spending, most interventionist premier in Alberta history. For a free-market, free-enterprise, small-government conservative, this must be galling. Canada's oilpatch sheds almost \$9B in spending: IHS Markit The pandemic and downturn are not Kenney's fault but he has made things worse by picking a fight with doctors over compensation and billing practices that led to the Alberta Medical Association launching a \$250-million lawsuit against his government last week. This 2015 file photo shows the Keystone Steele City pumping station, into which the planned Keystone XL pipeline is to connect to, in Steele City, Neb. (Nati Harnik/The Associated Press) His government gave itself a black eye by promising to maintain education funding and then days later cutting \$128 million that triggered mass layoffs. NDP Leader Rachel Notley, somebody who was premier during her own crisis — the Fort McMurray fire — has some advice for Kenney: "'Albertans need to know that they can trust Jason Kenney, that he's going to put his own agenda that was developed at a different time in a different place aside in order to come up with the best solutions for them,'" says Notley. "'That, I think, is evidenced by the doctor dispute and the education funding.'" OPINION | Kenney under fire from the front-line workers in the war on COVID-19 Notley says she's trying to work with the government during the crisis. In a private meeting with Kenney a month ago, she suggested he allow NDP members to sit on the government's economic recovery council. Kenney apparently didn't laugh her out of the room. He simply didn't respond. On Wednesday, though, his office sent me an email saying, in so many words, "'no way.'" "'The NDP, of course, is welcome to provide its suggestions via the Legislature Assembly or elsewhere,'" said the premier's office. "'We have admittedly been disappointed that at times the NDP seems more interested in playing petty partisan politics.'" Both the NDP and UCP can be accused of playing partisan games. That's one thing that hasn't changed in the past year. This column is an opinion. For more information about our commentary section, please read this editor's blog and our FAQ."

Figure 4.6 The original text input

The resulted summaries generated by the model with and without data augmentation can be seen in the Table 4.7.

Table 4.7 Generated summary compared with human summary from the original text

Summary generated by	Result
Human	Ontario health-care workers to hold minute of silence for cleaner who died from COVID-19
Model without data augmentation	keeping a big family' of sask woman ' says he was needed to isolate through advocates
Model with data augmentation	3 long term care homes run by same company report total of 71 deaths from covid 19

As will be seen later, the summary generated by the model with data augmentation, which differs from human generated summary, receives a higher ROUGE score than the one without data augmentation.

4.4 ROUGE Framework

As previously mentioned, the ROUGE framework would be used to evaluate the summaries generated by our models in comparison with the human reference summaries. The ROUGE score could be a value between 0 – 1. Tables 4.8 and 4.9 show the ROUGE scores of the two models for the whole dataset which had a total of 6,786 records.

Table 4.8 The ROUGE scores for the model with the data augmentation

Model with data augmentation	ROUGE 1	ROUGE 2	ROUGE L
Highest score	0.97	0.97	0.97
Lowest score	0.17	0.07	0.17
Average score	0.80	0.76	0.80
SD	0.38	0.44	0.38

Table 4.9 The ROUGE scores for the model without the data augmentation

Model without data augmetation	ROUGE 1	ROUGE 2	ROUGE L
Highest score	0.97	0.97	0.97
Lowest score	0.18	0.07	0.18
Average score	0.58	0.51	0.58
SD	0.39	0.45	0.39

To summarize, the ROUGE framework was used to compare between the reference summaries and the summaries generated by our models. Table 4.7 and Table 4.8 show the variety of the scores because some of the generated summaries had introduced the new words that did not exist in the reference summaries, and this affected the ROUGE scores which were computed based on the words in the reference summaries. Therefore, while the ROUGE framework could be used for text summarization to a certain extent, but we had to be cautious with the abstractive summarization because the summaries were newly generated and not extracted.

4.5 Comparison with Related Work

As Hayatin *et al.* [15] proposed a transformer-based model for text summarization using a number of identical layer encoder decoders (MTDGT-N) which includes tokenization, word embedding, etc. MTDGT-N model was experimented on the same dataset as our work. Therefore, we compare the ROUGE scores of our work and the models of [15] which focused on MTDGT2, MTDGT5, and MTDGT6 (the postfix number is a representation of the number of identical layer encoder decoders) as shown in Table 4.10.

Table 4.10 The comparison between the MTDGT model result and the model result

Model	ROUGE-1	ROUGE-2
The transformer C model (TCM)	0.45	0.26
MTDTG2	0.51	0.34
MTDTG5	0.56	0.38
MTDTG6	0.58	0.42
Model with data augmentation	0.79	0.75
Model without data augmentation	0.58	0.51

From the above table, we can see that our model using the data augmentation can outperform the MTDGT-N models with the ROUGE-1 as 0.79 and ROUGE-2 as 0.75. It can also be seen that our model has the higher ROUGE scores compared with for straightforward transformer model (the transformer C model).

CHAPTER 5 CONCLUSION AND FUTURE WORK

This research work applied the abstractive text summarization on the dataset of Covid-19 news from the CBC available from the Kaggle online database, with the data augmentation. The abstractive text summarization method, as opposed to the extractive summarization method, generates summaries based on content but does not necessarily use the same words or phrases as in the original documents. Moreover, data augmentation on the rule-based technique was applied before training the model in order to increase the volume and diversity of the dataset and improve the robustness and performance of the model as it prevents the model from overfitting. Also, the attention mechanism was exploited in order to make the model keep the important information from the document and use it to generate the summary.

After the data preparation, we applied the encoder-decoder architecture, LSTM, and attention mechanism to generate summarization. For the evaluation, the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) framework was utilized. ROUGE is a set of metrics and software packages used for evaluating natural language processing tasks such as summarization. It includes several automatic evaluation methods that measure the similarity between reference and candidate summaries. In comparison with the related work by Hayatin *et al.* [15], our model, with data augmentation, received higher ROUGE scores (ROUGE 1 and ROUGE 2) than their best transformer models scores at 0.79 and 0.75 respectively. One point that should be noted when using ROUGE metrics for evaluating abstractive summarization is that because some of the generated summaries have introduced new words that do not exist in the reference summaries, this affected the ROUGE score which was computed based on words in the reference summaries. Therefore, while the ROUGE framework can be used for text summarization to a certain extent, we have to be cautious with abstractive summarization because the summaries are newly generated and not extracted.

For future research, since our model only focuses on the rule-based technique for the data augmentation, it could be beneficial to experiment with mix-up augmentation or graph-structured technique in order to add diversity to the dataset. Also, the summarization method in this research used only the abstractive text summarization without special parameter tuning or adjusting specific layer based on LSTM. Therefore, it might be a benefit to experiment with the parameter tuning or changing the LSTM layer to Gated Recurrent Unit (GRU) or another layer that can handle the long input sequence. Moreover, the dataset could be expanded to include more recent Covid-19 news data which might be different now in 2022 since vaccines are available and Covid-19 has been changed from pandemic to endemic in some areas. In addition, our research used the ROUGE framework to evaluate the generated summary and candidate summary but it could be beneficial to experiment with a new framework that can be used to evaluate the summaries.

REFERENCES

1. Manjari, K. U., Rousha, S., Sumanth, D. and Devi, J. S., 2020, "Extractive Text Summarization from Web Pages Using Selenium and TF-IDF Algorithm", **Proceedings of the Fourth International Conference on Trends in Electronics and Informatics**, 15-17 June 2020, Tirunelveli, India, pp. 648-652.
2. Singkul, S., Khampingyot, B., Maharattamalai, N., Taerunguang, S. and Chalotthon, T., 2019, "Parsing Thai Social Data: A New Challenge for Thai NLP", **Proceedings of the 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing**, 30 October - 1 November 2019, Chiang Mai, Thailand, pp. 1-7.
3. Shah, J., Sagathiya, M., Redij, K. and Hole, V., 2020, "Natural Language Processing Based Abstractive Text Summarization of Reviews", **Proceedings of the International Conference on Electronics and Sustainable Communication Systems**, 2-4 July 2020, Coimbatore, India, pp. 461-466.
4. Andhale, N. and Bewoor, L.A., 2016, "An Overview of Text Summarization Techniques", **Proceedings of the 2016 International Conference on Computing Communication Control and Automation (ICCUBE)**, 12-13 August 2016, Pune, India, pp. 1-7.
5. Batra, P., Chaudhary, S., Bhatt, K., Varshney, S. and Verma, S., 2020, "A Review: Abstractive Text Summarization Techniques Using NLP", **Proceedings of the 2020 International Conference on Advances in Computing, Communication & Materials**, 21-22 August 2020, Dehradun, India, pp. 23-28.
6. Rahman, M. and Siddiqui, F. H., 2019, "An Optimized Abstractive Text Summarization Model Using Peephole Convolutional LSTM", **Symmetry** **2019**, Vol.11, No.10, pp. 1290-1314.
7. Talukder, M. A. I., Abujar, S., Masum, A. K. M., Akter, S. and Hossain, S. A., 2020, "Comparative Study on Abstractive Text Summarization", **Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)**, 1-3 July 2020, Kharagpur, India, pp. 1-4.

8. Logeesan, J., Rishoban, Y. and Caldera, H.A., 2020, "Automatic Summarization of Stock Market News Articles", **Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval**, 18-20 December 2020, Seoul, Republic of Korea, pp. 1–5.
9. Syed, A. A., Gaol, F. L., and Matsuo T., 2021, "A Survey of the State-of-the-Art Models in Neural Abstractive Text Summarization", **IEEE Access**, Vol. 9, No. 1, pp. 13248 – 13265.
10. Hochreiter S. and Schmidhuber J., 1997, "Long Short-Term Memory", **Neural Computation**, Vol. 9, No. 8, pp. 1735-1780.
11. Kuremoto, T., Tsurda, T. and Mabu, S., 2019, "Summarizing Articles into Sentences by Hierarchical Attention Model and RNN Language Model", **Proceedings of the 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics**, 19-21 October 2019, Suzhou, China, pp. 1-6.
12. Sirirattanajakarin S., Jitkongchuen, D. and Intarapaiboon, P., 2020, "BoydCut: Bidirectional LSTM-CNN Model for Thai Sentence Segmenter", **Proceedings of the 1st International Conference on Big Data Analytics and Practices (IBDAP)**, 25-26 September 2020, Bangkok, Thailand, pp. 1-4.
13. Ong, W. H., Tay, K. G., Chew, C.C. and Huong, A., 2020, "A Comparative Study of Extractive Summary Algorithms Using Natural Language Processing", **Proceedings of the 2020 IEEE Student Conference on Research and Development**, 27-29 September 2020, Batu Pahat, Malaysia, pp. 406-410.
14. Lin, C.-Y., 2004, "Rouge: A package for automatic evaluation of summaries." **Proceedings of the Text Summarization Branches Out**, 25-26 July 2004, Barcelona, Spain, pp. 74-81.
15. Hayatin, N., Ghufroon, K. M. and Wicaksono, G. W., 2021 "Summarization of COVID-19 News Documents Deep Learning-based Using Transformer

- Architecture”, **TELKOMNIKA Telecommunication, Computing, Electronics and Control**, Vol.19, No.3, pp. 754-761.
16. Shorten, C., Khoshgoftaar, T. M. and Furht B., 2021, “Text Data Augmentation for Deep Learning”, **Journal of Big Data**, Vol. 8, No. 101, pp.1-34.
 17. Miller, G. A., 1995, “Wordnet: a lexical database for English”, **Communication of the ACM**. Vol. 38, No.11, pp. 39-41.
 18. Marcus, M. P., Marcinkiewicz, M. A. and Santorini, B., 1993, “Building a large annotated corpus of English: the penn treebank”, **Computation Linguistics**, Vol. 19, No. 2, pp. 313-330.
 19. Jiajun, Z., Jie, S. and Xuan, Q. 2020. “Data Augmentation for Graph Classification”. **Proceedings of the 29th ACM International Conference on Information & Knowledge Management**, 19-23 October 2020, New York, USA, pp. 2341–2344.
 20. Nayak, T. and Ng, H., T., 2020, Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction, “**Proceedings of the AAAI Conference on Artificial Intelligence**”, Vol. 34, No. 05, pp. 8528-8535.
 21. Drive into Deep Learning, 2020, **9.6. Encoder-Decoder Architecture**, [Online], https://d2l.ai/chapter_recurrent-modern/encoder-decoder.html [24 April 2022]
 22. Devlin, J., Chang, M., Lee, K. and Toutanova, K., 2019, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, “**Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**”, 2-7 June 2019, Minneapolis, Minnesota, USA, pp. 4171-4186.
 23. Machine Learning Mastery, 2019, **Attention in Long Short-Term Memory Recurrent Neural Networks**, [Online], Available: <https://machinelearningmastery.com/attention-long-short-term-memory-recurrent-neural-networks> [24 April 2022]
 24. Bahdanua D., Cho K. and Bengio Y., 2015, “Neural Machine Translation by Jointly Learning to Align and Translate”, **Proceedings of the 3rd International Conference on Learning Representations**, 7-9 May 2015, San Diego, CA, USA, pp. 1-15.
 25. Jiang, W., Zou, Y., Zhao, T., Zhang, Q. and Ma, Y., 2020, “A Hierarchical Bidirectional LSTM Sequence Model for Extractive Text Summarization in

Electric Power Systems”, **Proceedings of the 13th International Symposium on Computational Intelligence and Design (ISCID)**, 12-13 December 2020, Hangzhou, China, pp. 291-294.

APPENDIX A

The Loss Score

The loss score of the model for with and without using the data augmentation.

Table A.1 The loss score for the modal

Epoch	Loss Score of the model	
	Data Augmetation	without Data Augmetation
1	7.8424	7.9473
10	4.5362	4.6186
20	4.3136	4.4531
30	4.0784	4.2597
40	3.7873	4.0992
50	3.4899	3.8996
60	3.2359	3.7142
70	3.0100	3.4101
80	2.7211	3.1283
90	2.4408	2.8867
100	2.1095	2.6086
110	1.8311	2.3111
120	1.5486	2.0370
130	1.3171	1.7099
140	1.1051	1.4868
150	0.9206	1.1921
160	0.7730	1.0144
170	0.6234	0.7996
180	0.5167	0.6487
190	0.4398	0.5483
200	0.3448	0.4545
210	0.2737	0.3756
220	0.2223	0.3220
230	0.1670	0.2679
240	0.1240	0.2225
250	0.1045	0.1815
260	0.0710	0.1453
270	0.0553	0.1114
280	0.0426	0.0862

290	0.0351	0.0695
300	0.0286	0.0654
310	0.0239	0.0494
320	0.0207	0.0385
330	0.0176	0.0326
340	0.0152	0.0284
350	0.0144	0.0251
360	0.0235	0.0226
370	0.0128	0.0206
380	0.0106	0.0189
390	0.0092	0.0175
400	0.0081	0.0163
410	0.0073	0.0153
420	0.0066	0.0144
430	0.0059	0.0137
440	0.0054	0.0130
450	0.0050	0.0124
460	0.0046	0.0119
470	0.0042	0.0114
480	0.0039	0.0110
490	0.0036	0.0106
500	0.0034	0.0103
510	0.0031	0.0251
520	0.0029	0.0124
530	0.0041	0.0106
540	0.0085	0.0100
550	0.0038	0.0095
560	0.0031	0.0092
570	0.0027	0.0090
580	0.0024	0.0087
590	0.0022	0.0086
600	0.0020	0.0084
610	0.0019	0.0083
620	0.0018	0.0082
630	0.0016	0.0080
640	0.0016	0.0079
650	0.0015	0.0078
660	0.0014	0.0077

670	0.0013	0.0077
680	0.0013	0.0076
690	0.0012	0.0075
700	0.0011	0.0074
710	0.0011	0.0074
720	0.0010	0.0073
730	0.0010	0.0073
740	0.0010	0.0072
750	0.0009	0.0072
760	0.0009	0.0071
770	0.0009	0.0071
780	0.0008	0.0071
790	0.0008	0.0070
800	0.0038	0.0070
810	0.0015	0.0069
820	0.0012	0.0069
830	0.0011	0.0069
840	0.0010	0.0069
850	0.0009	0.0068
860	0.0008	0.0068
870	0.0008	0.0068
880	0.0007	0.0068
890	0.0007	0.0068
900	0.0007	0.0067
910	0.0007	0.0067
920	0.0006	0.0067
930	0.0006	0.0067
940	0.0006	0.0067
950	0.0006	0.0067
960	0.0051	0.0066
970	0.0011	0.0066
980	0.0008	0.0066
990	0.0007	0.0066
1000	0.0006	0.0066

APPENDIX B

The Time per Iteration

The sample of time per iteration of the training data.

Table B.1 The time per each iteration for each model.

Epoch	Model without DA	Model with DA
1	76.557	78.428
10	51.109	44.502
20	52.749	44.245
30	43.758	73.946
40	41.668	39.562
50	40.597	37.903
60	41.088	38.374
70	41.489	38.159
80	44.666	38.332
90	44.224	38.031
100	49.904	37.892
110	47.948	37.168
120	53.230	37.566
130	53.558	37.403
140	50.030	38.710
150	48.001	37.555
160	46.363	36.875
170	46.711	36.815
180	36.093	37.131
190	37.292	37.102
200	35.929	36.908
210	38.750	37.301
220	35.982	36.502
230	48.298	37.080
240	37.431	36.806
250	36.662	36.651
260	43.568	36.400
270	37.082	36.254
280	49.135	36.831

290	51.441	37.500
300	48.657	36.525
310	45.030	37.064
320	52.187	37.086
330	51.834	37.445
340	47.781	37.124
350	44.050	36.484
360	49.070	37.061
370	38.459	36.389
380	37.046	37.007
390	36.679	35.879
400	37.393	36.107
410	51.785	36.452
420	45.374	36.716
430	45.415	36.048
440	46.968	36.311
450	48.489	36.353
460	48.950	36.406
470	49.378	36.449
480	53.005	35.609
490	47.029	36.165
500	48.004	36.157
510	43.534	36.858
520	36.786	35.698
530	36.557	36.329
540	36.404	35.639
550	36.439	36.702
560	35.731	36.142
570	37.570	35.844
580	36.318	36.097
590	35.993	41.063
600	36.298	37.058
610	36.378	38.055
620	36.445	36.711
630	35.726	36.861
640	36.282	36.882
650	36.127	37.573
660	36.160	36.576

670	35.964	36.318
680	36.102	40.939
690	36.062	36.444
700	35.929	37.064
710	37.360	36.772
720	37.147	36.883
730	35.961	36.064
740	36.107	36.588
750	36.230	36.583
760	36.261	36.263
770	36.345	36.513
780	37.532	36.966
790	36.718	36.775
800	36.761	36.748
810	39.662	36.923
820	35.931	37.624
830	36.113	37.627
840	36.205	37.165
850	36.060	37.363
860	36.231	36.580
870	36.754	38.125
880	36.378	37.332
890	35.969	37.700
900	35.897	37.419
910	36.094	38.304
920	36.023	37.205
930	36.371	37.618
940	36.092	38.627
950	36.198	38.605
960	36.395	38.429
970	35.977	38.337
980	36.326	38.693
990	35.746	37.421
1000	36.336	37.751

APPENDIX C

The Sample of ROUGE Score

The samples of Rouge scores for the 2 models for the full data set.

Table C.1 ROUGE scores for the dataset without data augmentation

human summary	model summary	rouge 1	rouge 2	rouge l
COVID-19 outbreak leads to major changes in Alberta courthouses.	northern sask health authority warns of limited ppe for front line workers	0.1000	0.0000	0.1000
12 Manitoba First Nations schools to shutter Monday for 3 weeks.	1 death toll tops 4 000 for courses from home learning	0.0952	0.0000	0.0952
What can N.S. learn from other provinces about park use during a pandemic?	what can n s learn from other provinces about park use during a pandemic	0.9091	0.8889	0.9091
Student athletes use online workouts to stay sharp during COVID-19 restrictions.	student athletes use online workouts to stay sharp during covid 19 restrictions	0.9524	0.9474	0.9524
"Officials suspend personal visits with inmates in Ontario, federal prisons over COVID-19 concerns".	covid 19 outbreak declared at barton jail after travel to 1	0.2727	0.1000	0.2727
4 new COVID-19 deaths in Leeds-Grenville-Lanark.	4 new covid 19 deaths in leeds grenville lanark	0.9412	0.9333	0.9412
German court opens first Syria torture trial.	german court opens first syria torture trial	0.9412	0.9333	0.9412
"What you need to know about COVID-19 in B.C. on April 9, 2020".	what you need to know about covid 19 in b c on april 11 2020	0.7692	0.5455	0.7692

Social-distance wedding celebration becomes social sensation online.	grieving family says calgary declares first of first nations mourns public health	0.1000	0.0000	0.1000
2 deaths under investigation at park hosting homeless camp in Victoria.	2 deaths under investigation at park hosting homeless camp in victoria	0.9474	0.9412	0.9474
Northwood rejects claims of lack of protective equipment.	northwood rejects claims of lack of protective equipment	0.9333	0.9231	0.9333
3 more COVID-19 deaths recorded in B.C. as total number of cases climbs to 472.	what you need to know about covid 19 in b c on april 15 2020	0.3333	0.1250	0.3333
"Basketball player charged after refusing to leave Kitchener park, mayor says".	basketball player charged after refusing to leave kitchener park mayor says	0.9474	0.9412	0.9474
Saskatchewan lab joins global effort to develop coronavirus vaccine.	orderly at under staffed chsld dies of covid 19 remembered for care she showed patients	0.0952	0.0000	0.0952
Calls to distress lines climb amid increased anxiety over COVID-19.	what it's like to support covid 19 restrictions will get on the front line	0.3529	0.1333	0.3529
"As COVID-19 spreads in Montreal, public health authority cracks down on citizens' movement".	as covid 19 spreads in montreal public health authority cracks down on citizens' movement	0.9565	0.9524	0.9565
Trudeau says government will do 'everything necessary' to protect Canadians from COVID-19.	doctors and northern alberta meat plant plants plants open here's physical distancing	0.1053	0.0000	0.1053
Game on (or soon to be) for B.C. golfers as clubs bring in COVID-19 restrictions.	game on or soon to be for b c golfers as clubs bring in covid 19 restrictions	0.9412	0.9333	0.9412
Gasoline prices fall after oil takes huge tumble.	coronavirus what's happening in canada and around the world april 21	0.1250	0.0000	0.1250
"Confusion, fear as workers in Quebec seniors' homes brace for	confusion fear as workers in quebec seniors' homes brace for long fight against covid 19	0.9600	0.9565	0.9600

long fight against COVID-19".				
Alberta oilpatch companies sending home non-essential employees in preventative bid to combat COVID-19.	'you're a rock star' premier heaps praise upon 'champion' asl interpreter	0.0870	0.0000	0.0870
"28 residents, 27 staff at Vancouver's Haro Park care centre confirmed to have COVID-19".	translink announces u s 1 aid trump can keep up medical steps to u s	0.0909	0.0000	0.0909
'Am I going to see anyone again?': Hospital patients isolated from loved ones as COVID-19 stops family visits.	'we will be protected from free hospital staff more than country after support	0.2222	0.0000	0.2222
G7 nations pledge to work together 'for as long as necessary' in coronavirus battle.	what you need to know about covid 19 in waterloo region on april 30	0.1250	0.0000	0.1250

Table C.2 ROUGE scores for the dataset with data augmentation

human summary	model summary	rouge 1	rouge 2	rouge l
Backyard BBQs and park picnics: Edmonton council seeks clear rules for outdoor gatherings.	backyard bbqs and park picnics edmonton council seeks clear rules for outdoor gatherings	0.9600	0.9565	0.9600
Supply chain strong despite some rising food prices: N.L. wholesaler.	supply chain strong despite some rising food prices n l wholesaler	0.9412	0.9333	0.9412
Virtual meet-ups available for LGBTQ students to connect.	virtual meet ups available for lgbtq students to connect	0.9333	0.9231	0.9333
Lululemon apologizes after staffer offends with 'bat fried rice' T-shirt.	lululemon apologizes after staffer offends with 'bat fried rice' t shirt	0.9474	0.9412	0.9474
"Liberals hope to ban firearms used in Polytechnique, Dawson College shootings: sources"	liberals hope to ban firearms used in polytechnique dawson college shootings sources	0.9524	0.9474	0.9524
Choosing a 'bubble family' not always easy	choosing a 'bubble family' not always easy	0.9091	0.8889	0.9091
"North Korean leader Kim Jong-un makes first public appearance in weeks, say state media"	north korean leader kim jong un makes first public appearance in weeks say state media	0.9630	0.9600	0.9630
How can non-essential businesses operate? Manitoba public health officials clarify.	covid 19 on p e i what's happening thursday april 9	0.1176	0.0000	0.1176
Widow of Sask. man who died from COVID-19 calls plan to reopen province 'reasonable'.	widow of sask man who died from covid 19 calls plan to reopen province 'reasonable'	0.9600	0.9565	0.9600
Mobile medical unit to house inmates sick with COVID-19 at Abbotsford Hospital.	mobile medical unit to house inmates sick with covid 19 at abbotsford hospital	0.9565	0.9524	0.9565
Celebrating a century: 3 Edmontonians celebrate turning 100 during	a big family' at manoir stanstead seniors' home staff have moved in to keep covid 19 reopening plan	0.2400	0.0870	0.2400

COVID-19 pandemic.				
"New moms, experts worry about postpartum depression during COVID-19 as services cut back".	new moms experts worry about postpartum depression during covid 19 as services cut back	0.9565	0.9524	0.9565
Support flowing into northern community of La Loche where outbreak has been recorded at long-term care home.	support flowing into northern community of la loche where outbreak has been recorded at long term care home	0.9630	0.9600	0.9630
"Vancouver hotels being used for homeless should be bought by government, advocates say".	vancouver hotels being used for homeless should be bought by government advocates say	0.9333	0.9231	0.9333
Beer is a pandemic must-have and London's craft breweries will (literally) deliver.	beer is a pandemic must have and london's craft breweries will literally deliver	0.9412	0.9333	0.9412
Cancer researchers hoping 'Trojan Horse' virus key to COVID-19 vaccine.	'i don't forget the really important self hospital amid your surge amid pandemic	0.1053	0.0000	0.1053
Union demands sick leave for casual front line health-care workers in Alberta.	union demands sick leave for casual front line health care workers in alberta	0.9600	0.9565	0.9600
Burst pipe floods wildlife care centre.	burst pipe floods wildlife care centre	0.9333	0.9231	0.9333
"With Muskrat work halted, Nalcor says it can't predict when megaproject will be completed".	with muskrat work halted nalcor says it can't predict when megaproject will be completed	0.9412	0.9333	0.9412
"Plummeting oil prices mean short-term gain at gas pumps, long-term pain for producers, analysts say".	vancouver island community reveals short term plan to help homeless during covid 19 pandemic	0.2222	0.0800	0.2222
'It's just not going to be the same': Students share disappointment of losing traditional grad to	salvation army confirms case of covid 19 on downtown eastside	0.3158	0.1176	0.3158

COVID-19.				
"What you need to know about COVID-19 in B.C. on April 15, 2020".	vancouver hotels being used for homeless should be bought by government advocates say	0.1429	0.0000	0.1429
Some N.S. apartment landlords institute no-visitor policies amid COVID-19.	some n s apartment landlords institute no visitor policies amid covid 19	0.9412	0.9333	0.9412
EI claimants are going weeks without income as federal call system slows to a crawl.	ei claimants are going weeks without income as federal call system slows to a crawl	0.9524	0.9474	0.9524

CURRICULUM VITAE

NAME

Mr. Chatchawarn Limloypipat

DATE OF BIRTH

8 August 1992

EDUCATION RECORD**HIGH SCHOOL**

Trattrakankhun School

Trat, Thailand, 2010

BACHELOR'S DEGREE

Bachelor of Science (Computer Science)

Rajamangala University of Technology

Tawan-ok Chanthaburi Campus,

Chanthaburi, Thailand, 2014

MASTER'S DEGREE

Master of Science (Computer Engineering)

King Mongkut's University of Technology

Thonburi, Thailand, 2021

EMPLOYMENT RECORD

Web developer

Marketingbear, Bangkok, Thailand,

March 2020 - Present

PUBLICATION

Limloypipat, C. and Facundes, N.,

“Abstractive Text Summarization for
Covid-19 News with Data Augmentation”,

**Proceedings of the 2022 International
Conference on Digital Government**

Technology and Innovation, 24 - 25 March
2022. Bangkok, Thailand, pp. 56- 59.