Abstractive Text Summarization using BERT for Feature Extraction and Seq2Seq Model for Summary Generation

Kania Galih Widowati
School of Computer Science
BINUS University
Jakarta, Indonesia
kania.widowati@binus.ac.id

Nicholas Budiman
School of Computer Science
BINUS University
Jakarta, Indonesia
nicholas034@binus.ac.id

Kezia Foejiono
School of Computer Science
BINUS University
Jakarta, Indonesia
kezia.foejiono@binus.ac.id

Kartika Purwandari, S.Kom., M.Sc.
School of Computer Science
BINUS University
Jakarta, Indonesia
kartika.purwandari@binus.edu

***Abstract*—**

Text summarization is a technological development of the Natural Language Processing branch which takes points and main information from a paragraph. This is done to increase efficiency in understanding a paragraph or document. Text summarization itself has several types, namely extractive summarization, abstractive summarization, and hybrid summarization. In this study, we used the type of abstractive text summarization, which is a summarization resulting from deeper sentence understanding. That the resulting summarization will produce a new summary that is not always in the input text, so this will increase the reader's understanding. In text summarization, various methods have been used such as Support Vector Machine (SVM), Long Short Term Memory (LSTM), Bayesian Networks, and so on. In this study, we used pre-trained model Bidirectional Encoder Representations (BERT) derived from transformers, and Sequence to Sequence (Seq2Seq) and the dataset used was the CNN Daily Mail dataset. The pre-trained BERT model is used to perform feature extraction, the results will be forwarded by Seq2Seq for summarization. We also conduct evaluations using ROUGE, by comparing the results of the summary model with the results of the summary made by humans.

Keywords—abstractive text summarization, BERT, Sequence-to-Sequence

I. Introduction

In the midst of the rapid development of science and technology today, humans are very easy to access various existing data. In fact, almost all aspects of human life today have involved technology, especially Artificial Intelligence. On the one hand, this development has a positive impact that can provide reciprocity in improving the development of technology and knowledge, but on the other hand, with the amount of information that cam be accessed, humans begin to feel heavy to read and understand all the information available. Therefore, the presence of a text summarization engine is needed to help humans understand the core information of a document.

Text summarization is one of the important solutions in Natural Language Processing that involves creating the shortest version of a document, so that information can be obtained more quickly and precisely without losing the main information of the actual document [1]. Three types of text summarization are being popularly developed, namely Extractive Summarization, Abstractive Summarization, and Hybrid Summarization. Extractive summarization is a summarization technique where the output sentence will be the same as the sentence from the given document [2]. Abstractive summarization is a summarization technique based on a deeper understanding of the sentences in the original document and produces a new summary that is not always present in the original text [3]. Then the last one is Hybrid summarization which combines Extractive summarization which takes important sentences with Abstractive summarization which produces a new summary [4].

There are several popular algorithm used by many researchers in developing text summarization. Popular algorithms used for extractive summarization include TextRank, Latent Semantic Analysis (LSA), Support Vector Regression (SVR), Bayesian Network, and others. The popular algorithms used for Abstractive summarization are Sequence-to-Sequence, transformers, pointer-generator networks, recursive neural networks, and others. In this paper, we will try to use the BERT algorithm to extract features from a document and generate an abstractive summary using the Sequence-to-Sequence algorithm. We use these two algorithms because we want to produce an abstractive summary so that readers get a varied summary but with the same core information [3]. In addition, we used the BERT algorithm because it works better with language processing than other models [4].

In this paper, we will discuss text summarization using BERT and Sequence-to-Sequence algorithms. Here are some of the main points we will discuss in this paper. First, the performance when combining the two algorithms compared to using one of them. Second, the evaluation metrics are used to accurately assess the summary's quality when using the combination of the two algorithms. And finally, about the level of complexity of the input text and whether it will affect the performance of text summarization when combining the two methods.

II. Literature Review

Automatic text summarization is one of the fields in Natural Language Processing (NLP) that is experiencing rapid development to summarize text by reducing sentences that are less important but still retain important information from several documents [5] [6]. Given the ever-increasing human need for important information and the rapidly expanding availability of information, text summarization is becoming a technology that is greatly needed [7]. Even the International Data Corporation (IDC) predicts that by 2025, there will be 175 Zettabytes (ZB) of data in existence [8]. Automatic summarization is predicted to assist people in swiftly assembling crucial information from a variety of sources [9]. While researchers have developed many text summarization algorithms, they have yet to find one that comes close to human text summarization [10]. This is due to the different needs in text summarization.

Based on its technique, automatic text summarization is divided into two, namely Extractive Summarization and Abstractive Summarization [11] [12]. With the text summarization method known as "Extractive Summarization," the model simply copies a few sentences from the original text [13]. In Extractive Summary, the model does not need to understand the meaning/context of the sentence. This technique is considered easier than Abstractive Summarization because it guarantees grammar and accuracy by copying most of the text from the source document [7]. An approach to text summarization known as "abstractive summarization" involves the model attempting to comprehend the subject matter of the original text. By creating new sentences that cover the same subject and main points as the source document, abstractive summarization creates a summary. Because it is capable of paraphrasing and generalizing new sentences, abstractive summarization has the potential to produce summaries in the same way that humans do [6] [7]. In addition to Techniques, automatic text synthesis can be divided based on the source of the document, with single-document synthesis using a single document as the source and multi-document synthesis using a number of documents on the same subject [5] [7].

One of them is a study conducted in 2019 by Qicai Wang and colleagues, which utilizes BERT as a word embedding with reinforcement learning for abstractive text summarization. To calculate the performance of the model, they compared ROUGE metrics using CNN Daily Mail Dataset [14]. In this study, the author uses extractive strokes and then takes sentences that are in accordance with the topic and then summarizes them again abstractively so it can produce a new and much denser summary. In this study the writer used BERT to tokenized words and sentences. There are several processes, the first is to train sub models, the second is to train models end to end with reinforcement learning and then mapped to the process of installing functions [14]. When sentences from articles are fed into the BERT model, it may express tokenized words as related word embeddings, and vector representations of each phrase are then produced [14].

Shehab carried out experiments with the DistilBERT and SqueezeBERT models in a different paper in 2022. For extractive summarization tasks, the DistilBERT model was employed in this investigation. The Adam optimizer was used to train the model, using a batch size of 3000 and a learning rate of 2103. In order to assess the evaluation algorithm, precision, recall, and F-score were computed. On a GoogleGPU session, the training period was 25 minutes for every 1000 checkpoints. A single GPU was used to train the DistilBERT summarizer, which was able to retain 98% of the baseline model's summary generation performance while reducing its size by 36%. In contrast, training the SqueezeBERT model took 60 hours and used the same hyperparameters as the DistilBERT experiment. The SqueezeBERT summarizer was able to maintain 98% of the summary performance of the baseline model despite a size decrease of 49% [15].

In addition to BERT, other techniques for text summarization include sequence to sequence. regarding Yong Zhang's 2019 research. They employed a convolutional seq2seq model, a CNN model with GLU and residual connections, a hierarchical attention mechanism, and a copying mechanism to improve text summarization performance [3]. Tian Shi's 2020 study, which also suggested text summary with sequence to sequence. By focusing on the key information present and the significance of the offered papers, this study aims to significantly increase the density of the Long summary. Neural Abstractive Text Summarization (NATS) is used for this. Tian Shi used the CNN Daily/Mal dataset to assess the model's performance in a manner similar to the earlier study. The effectiveness of attention-based seq2seq models was greatly improved, according to the results, by the pointing mechanism. G11110 outperformed C10110 on CNN/Daily Mail data, however C10110 had higher ROUGE scores for the text summarizing and headline generating tasks in the NATS toolkit's test on the Newsroom dataset [16].

An artificial text summarizing model must be capable of producing summaries with a high level of relevance and minimum word repetition if it is to provide high-quality text summaries [10]. As a result, researchers are competing to test different algorithms and obtain the best outcomes. Researchers have become interested in two models: BERT and Sequence-to-Sequence. Transformers-based model called Bidirectional Encoder Representations from Transformers (BERT) is learned without supervision [12]. The BERT approach can be used for either an abstract or an extractive summary [12]. In an extractive summary, BERT chooses key phrases and examines the most pertinent terms to produce a summary [6]. BERT is tasked with comprehending the context of each sentence and how they relate to one another in order to provide an abstractive summary [6]. In addition to BERT, the Sequence-to-Sequence paradigm is expanding quickly. In this architecture, a decoder builds a new summary from the encoder's representation after processing text to produce a structured representation of words [17]. It has been demonstrated that both models increase the precision of autonomous text summarization engines [11] [17].

III. METHODOLOGY

A. *Workflow*

In this paper, we will create an abstractive text summarization model that can be used to summarize texts such as news texts, email texts, novel texts, and others. Our model combines BERT model for feature extraction and Sequence-to-Sequence model for generating new summarized text.

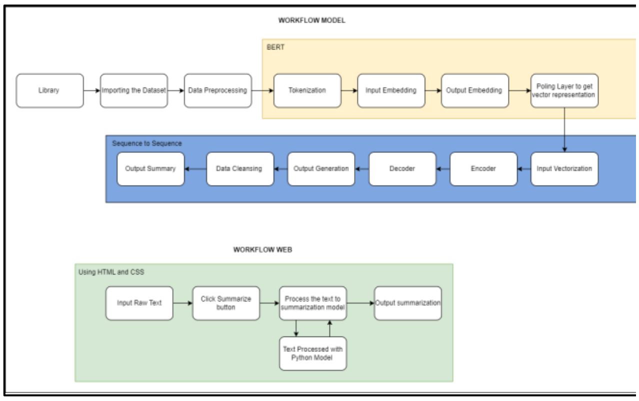

Fig 1: Model details

The first step in the models is to write the required library then import the dataset. The dataset we use for training is the CNN Daily-Mail dataset, then when testing using the BBC Dataset where we declare variables with the name text and contain several paragraphs which will be tokenized and determine stopwords. In addition, word count is also calculated to find out how many words appear and will be used to calculate the probability for each word. In the flowchart above, there is input embedding and output embedding and produces a vector representation that will be used at the stage using the sequence-to-sequence method. In the sequence-to-sequence vector obtained from the calculation results using the BERT method, the encoder and decoder will be carried out and produce a summary output. However, this result is still not final because cleansing needs to be done such as removing punctuation marks (\"~). After data cleaning, a summary output will be generated.

In the website section, the thing that needs to be done is to enter the text that will be summarized in the text input section, then the user continues by pressing the "summarize" button. At this stage, the text will be processed with the python model that has been made at the top, then it will provide output in the form of summarize results which will appear in the summary result section.

B. *Dataset*

The dataset we use is the CNN Daily-Mail dataset. We directly call the pretrained model with the Sequence-to-Sequence method but the tokenizer is done using the BERT method. To run this model, we need to import library transformers for AutoTokenizer and AutoModelForSeq2SeqLM. CNN Daily-Mail dataset is taken through the Hugging Face Model Hub so that when the call is made it will automatically download. Then in the training and testing section, we use the BBC dataset, where this data has several topics, namely Business, Politics, Entertainment, Sport and Tech. This BBC dataset is in the form of a .txt file and each category has approximately 500 files. So we only use one file for testing then also enter the file from the human summary results then compare the summary results from the model and the summary results from humans, which are contained in the BBC dataset. In the model we created, test1_.txt is a file that we took from BBC dataset to do testing, and key1.txt is a summary result from humans.

IV. IMPLEMENTATION

In building an Abstractive Summarization model using a combination of BERT and Sequence-to-Sequence, several modules and classes are needed. We use Python programming language in building this Abstractive Summarization model. The modules we use are torch and transformers. The torch module is used as a deep learning framework to create, train, and evaluate the summarization model. Them, the transformers module is used to process natural language by learning remote dependencies in word order using neural networks. From transformers module we use two classes to create Abstractive Summarization models. The two classes are AutoTokenizer and AutoModelForSeq2SeqLM. Class AutoTokenizer serves to provide tokenization of the text for use with transformers. The AutoTokenizer class can automatically generate tokens on text based on pretrained transformer models. Then class AutoModelForSeq2SeqLM serves to provide a way to use transformers for Sequence-to-Sequence tasks. AutoModelForSeq2SeqLM will be used to generate text summaries based on the pretrained transformer models.

To generate Abstractive Summarization, the model will perform several processes as follows. First, the model requires several modules such as torch, transformers, pprint, and rouge. Next, tokenizer processing using BERT and summarization model using Sequence-to-Sequence are performed. The tokenizer will be used to convert the text into a set of tokens that can be understood by the model. The model will then use the tokens to learn and produce a summary of the text. Next, a file containing the text to be summarized is provided. The model will read the text in the file and perform some data preprocessing. To produce a better summary, split the text data into sections consisting of a maximum of 512 characters. Then each part will be summarized separately. The text parts will be encoded using a tokenizer, then the encoded parts are fed into the summarization model. Then the model will generate the summarized text. The summary result will be decoded by the tokenizer to produce semi-final summary parts. The semi-final summary parts will be combined into one final summary. And finally, to analyze the performance of the model, we use Rouge Metric to compare the summary produced by the model with the summary produced by humans.

V. Result

A. *Text Summarization Result of the Model*

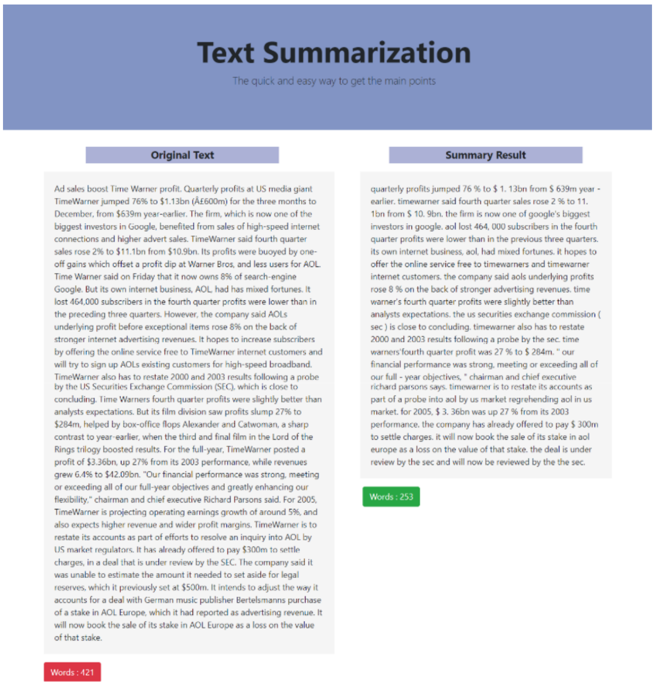

Fig2: The result of model

In Figure 2, the Abstractive summarization model using BERT and Seq2Seq methods successfully produces a text summary by reducing 168 words. In addition, the summary results also show abstractive properties, namely the creation of new sentences.

B. *Evaluation Strategy*

We employ rogue metrics to assess the model. Because ROUGE measures how well the text summarization system extracts pertinent information from the reference text, it has the capacity to evaluate how one text relates to another. By contrasting the system summary with the specified human reference, rouge metrics are thought to be able to help objectively assess the quality of the system for summarizing information. The text to be compared in this instance is the summary text produced by the model and the summary text produced by a person.

There are various Rouge metric variants available, but Rouge-N and Rouge-L are the two that are most frequently used. The n-gram similarity between the reference text and the system-summarized text can be calculated using Rouge-N. Additionally, Rouge-N has the ability to count the n-grams that match in both texts and produce a score based on how similar they are. The longest word sequence similarity between the system summary text and the reference text can be measured by Rouge-L, in contrast. Rouge-L can also figure out how long the matching sequences are and then provide a score depending on how similar they are.

When comparing the model summaries to human summaries, we utilized Rouge-N with N=1 to determine how similar the unigrams (individual words) were between the model summaries and the human summaries, and N=2 to determine how similar the bigrams were. Rouge-L was also utilized to determine the lengthiest word order similarity.

C. *Evaluation Result*

The following are the evaluation results with a text length of 421 words using Rouge Metrics.

	Rouge-1	Rouge-2	Rouge-L
Recall	40.7%	18.7%	40.7%
Precision	29.3%	11.2%	29.3%
F1-score	34.1%	14.0%	34.2%

Tabel 1. Rouge Metrics Evaluation for longer text

For Rouge-1, Rouge-2, and Rouge-L in this instance, the recall (r) and f1-score (f) are quite high, demonstrating that the model is able to collect the majority of the pertinent information from the reference text. However, the precision (p) ratings are typically low, indicating that the models either tend to produce longer summaries or that the summaries may contain some irrelevant information.

Despite having a good recall and f1-score overall, the model might create summaries that are more accurate and consistent with the reference text with an increase in the precision score. To get a fuller understanding of how well the text summarization model performs, additional analysis—such as comparisons with other models or with human judgment—is necessary.

The following are the evaluation results with a text length of 185 words using Rouge Metrics.

	Rouge-1	Rouge-2	Rouge-L
Recall	32.8%	22.2%	32.8%
Precision	23.3%	13.4%	23.3%
F1-score	27.3%	16.7%	27.3%

Tabel 2. Rouge Metrics Evaluation for shorter text

The ROUGE evaluation results show that input in the form of shorter articles has an accuracy level that is not much different from longer articles. From these two tables, it can be concluded that this model is good for all text lengths.

VI. Conclusion

One of the most essential fields in light of the quickening pace of technological advancement is text summarization. Extractive and abstractive text summarization are becoming more and more necessary. In this study, we demonstrate that a suitable text summarizing model may be created by combining the BERT approach for feature extraction with Sequence-to-Sequence for producing summarized text. We demonstrate that the fusion of the two models can result in summaries that are pertinent to the source text using Rouge Metrics' evaluation technique. Then, based on Rouge's size, we can conclude that both shorter texts and longer texts will produce text summaries with similar levels of relevance. The text summarizing model proved to be quite effective in this experiment at creating accurate and pertinent summaries. To make sure that the summaries maintain fullness of information and clarity in the larger context, it is also crucial to examine instances of the summaries created and carry out further evaluation. The performance of the text summarization model can be more thoroughly assessed by contrasting it with other models or human judgment.

Bibliografi

[1] A. P. Widyassari, A. Affandy, E. Noersasongko, A. Z. Fanani, A. Syukur dan R. S. Basuki, "Literature Review of Automatic Text Summarization : Research Trend, Dataset and Method," *IEEE,* p. 1, 2019.

[2] Y. Ruifeng, W. Zili dan L. Wenjie, "Fact-level Extractive Summarization with Hierarchical Graph Mask on BERT," *International Conference on Computational Linguistics,* pp. 5629-5639, 2020.

[3] Y. Zhang, D. Li, Y. Wang, Y. Fang and W. Xiao, "Abstract Text Summarization with a Convolutional Seq2seq Model," *National University of Defense Technology,* p. 1, 2019.

[4] S. S. Lakshmi and M. U. Rani, "Hybrid multi-document text summarization via categorization based in BERT deep learning models," *International Journal of Health Sciences,* pp. 5346-5369, 2022.

[5] A. Joshi, E. Fidalgo, E. Alegre dan L. Fernandez-Robles, "DeepSumm: Exploiting topic models and sequence to sequence metworks for extractive text summarization," *Expert System with Applications,* vol. 211, 2023.

[6] Y. Jingyi, H. Chenlong, H. Kamigaito, H. Takamura dan M. Okumura, "Abstractive Document Summarization with Word Embedding Reconstruction," *Proceedings of Recent Advances in Natural Language Processing,* pp. 1586-1596, 2021.

[7] N. C. Prakash, A. P. Narasimhaiah, J. B. Nagaraj, P. K. Pareek, N. B. Marythikumar dan R. I. Manjunath, "Implementation of NLP based automatic text summarization using spacy," *International Journal of Health Science,* pp. 7508-7521, 2022.

[8] M. K.B, B. S. M, I. Venkat dan S. R. B, "Text Summarization using ML and NLP," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. 9, no. 4, pp. 1188-1190, 2020.

[9] D. B. Deshmukh, "Text Summarization using Python NLTK," *International Journal os Advanced Research (IJAR),* pp. 202-209, 2022.

[10] K. A. R. Issam dan S. Patel, "Topic Modelling Based Extractive Text Summarization," *International Journal of Innovative Technology and Exploring Engineering,* vol. 9, pp. 1710-1719, 2020.

[11] C. Jiaao dan Y. Diyi, "Multi-View Sequence-to-Sequence Models with Comversational Structure for Abstractive Dialogue Summarization," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* pp. 4106-4118, 2020.

[12] Y. Liu dan M. Lapata, "Text Summarization with Pretrained Encoders," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing,* pp. 3730-3740, 2019.

[13] Y. K. Khor, C. W. Tan dan T. M. Lim, "Text Summarization on Amazon Food Reviews using TextRank," *International Conference on Digital Transformation and Applications (ICDXA),* pp. 113-120, 2021.

[14] W. Qicai, L. Peiyu, Z. Zhenfang, Y. Hongxia, Z. Qiuyue dan Z. Lindong, "A Text Abstraction Summary Model Based on BERT Word Embedding and Reinforcement Learning," *Applied Science,* p. 2, 2019.

[15] S. Abdel-Salam dan A. Rafea, "Performance Study on Extractive Text Summarization Using BERT Models," *Information MDPI,* p. 8, 2022.

[16] T. Shi, Y. Keneshloo, N. Ramakrishnan dan C. K. Reddy, "Neural Abstractive Text Summarization with Sequence to Sequence Models," *ACM Journal,* p. 28, 2020.

[17] Z. Haoyu, C. Jingjing, X. Jianjun dan W. Ji, "Pretraining-Based Natural Language Generation for Text Summarization," *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL),* pp. 789-797, 2019.