Ministry of Science and Higher Education of the Russian Federation
NATIONAL RESEARCH TOMSK STATE UNIVERSITY (NR TSU)
Research and Education Center “Higher IT School” (HITs)

APPROVED BY
Head of the Main Educational Program
Associate Professor, Candidate of Sciences (Technology)
__________________D.O. Zmeev
(signature)
«14» june 2024

BACHELOR’S THESIS

EXPLORING THE CAPABILITIES OF LUHN SUMMARIZATION, BART, PEGASUS, AND CLAUDE LLM FOR NEWS SUMMARIZATION

Main educational program 09.03.04 – Software Engineering

Specialization “ITS/TSU Software Engineering”

Arisudana Gede Yoga

Bachelor’s Thesis Supervisor
Senior Lecturer of REC “Higher IT School”
______________V.V. Prishchepa
(signature)
«14» june 2024

Author
Student of group No. 972006
______________G.Y. Arisudana
(signature)
«14» June 2024

Tomsk – 2024

The Ministry of Science and Higher Education of the Russian Federation
NATIONAL RESEARCH TOMSK STATE UNIVERSITY (NR TSU)

REC “Higher IT School”

APPROVED
MEP Head
Cand. Sc. (Technology),
Associate Professor
__________ D.O. Zmeev
16. 02. 2024

THE TASK

Regarding the bachelor’s final qualification work implementation by a student
Arisudana Gede Yoga / Арисудана Геде Йога

(student’s full name)

In the field of study 09.03.04 Software Engineering, specialization ITS/TSU Software Engineering

1. The bachelor’s final qualification work theme (in the English and in the Russian languages)
Exploring the Capabilities of Luhn Summarization, BART, Pegasus, and Claude LLM for news summarization / Изучение возможностей метода Луна, BART, Pegasus и Claude LLM в реферировании новостей

2. The deadlines for the completion of task:
a) to academic office – 14. 02 2024.
б) to State Examination Board – 15. 02. 2024

3. Description:
The purpose of the work is to explore the capabilities of Luhn Summarization, BART, Pegasus, and Claude LLM for news summarization.

goals and objectives of the FQW, expected results

The expected results of the work are: provided a text summarization model for each technique, analyzed each of the text summarization models, and provided a conclusion by comparing each of the summarization models.

The organization that is being researched:

Final qualification work supervisor

Senior Lecturer of REC “Higher IT School” (position, place of work) __________ (signature) / V. V. Prishchepa (initials, surname)

The task is accepted for execution

13. 02. 2024 (date) __________ (signature) / G.Y. Arisudana (initials, surname)

ABSTRACT

Bachelor’s thesis 62 pages, 23 figures, 11 tables, 39 sources, 1 appendix.

The purpose of this work is to explore and compare the capabilities of Luhn summarization, BART, Pegasus, and Claude LLM for news summarization, leveraging open-source resources.

Work results: The text summarization model for each technique that will be analyzed has been provided, the performance of each text summarization model has been analyzed using the results of each specific metric, and the comparison of the outcome metrics obtained from each of the summarization models has been concluded.

Key words: LUHN SUMMARIZATION, BART, PEGASUS, CLAUDE LLM, NEWS SUMMARIZATION

CONTENT

GLOSSARY

BART – (abbreviation) Bidirectional and Autoregressive Transformer. A denoising autoencoder for pretraining sequence-to-sequence models [1].

LLM – (abbreviation) Large Language Model. Machine learning models which can understand and generate a human language text [2].

ROUGE – (abbreviation) Recall-Oriented Understudy for Gisting Evaluation. A Package for Automatic Evaluation of Summaries [3].

API – (abbreviation) Application Programming Interface. An interface that can allows two to interact with each other [4].

Pre-trained Model – A saved network that was previously trained on a large dataset [5].

Tokenization – Process of converting a sequence of text into smaller parts [6].

Prefix – An affix added to the beginning of a word where it changes the meaning or value [7].

Inference – Process of running data points into a machine learning model to calculate an output [8].

Fine-tune – Process of taking a pretrained machine learning model and further training it on a targeted data set [9].

Training Hyperparameters – Parameters whose values control the learning process and determine the values of model parameters [10].

Trainer Class – Class that provides an API for feature-complete training in PyTorch [11].

Jupyter Notebook – A web-based interactive environment specifically designed for data science tasks [12].

PyTorch – A popular open-source deep learning framework renowned for its flexibility and ease of use [13].

Hugging Face – AI Company that provide platform and community for developers that are focusing on machine learning task [14].

Epoch – One complete pass of the training dataset through the algorithm [15].

Learning Rate – A concept that controls the size of these parameter updates and influences the speed and stability of the optimization process [16].

Batch Size – Number of samples that is used in one epoch to train a neural network [17].

Weight Decay – A regularization technique that is used to regularize the size of the weights of certain parameters [18].

Gradient Accumulation Steps – A technique used to support larger batch sizes given limited available of GPU memory [19].

INTRODUCTION

In today's information age, vast amounts of data are readily available. While this abundance holds immense value, extracting key insights and essential information from large volumes of text can be a significant time investment. News articles, in particular, often contain a wealth of information, but reading them in their entirety can be cumbersome. Ideally, a solution could exist to condense these texts and efficiently extract the most important details.

Fortunately, numerous text summarization techniques have emerged. These methods range from basic approaches, relying on simpler logic for summarizing text, to more advanced methods utilizing complex architectures and pre-trained models. Each technique undoubtedly offers unique advantages and limitations.

This paper aims to explore the capabilities of several prominent summarization techniques. Specifically, we will investigate the effectiveness of Luhn summarization, BART, Pegasus, and Claude LLM in summarizing news articles. By utilizing open-source news article resources, this research will evaluate the strengths and weaknesses of each method in extracting crucial information from news content. To achieve these next task must be done:

- Provide a text summarization model for each technique that will be analyzed.
- Analyze the performance of each text summarization model using the results of each specific metric.
- Examine by comparing the outcome metrics obtained from each of the summarization models.

1 Design

In order to explore the capabilities of Luhn summarization, BART, Pegasus, and Claude LLM for news summarization, a preliminary phase is undertaken, during which a number of preparatory steps are carried out. These include defining the data format for the news articles, configuring each technique for optimal performance on content of this kind and applying evaluation techniques in order to assess the results.

1.1 Requirements

In order to complete the mentioned text summarization model in its entirety, it is necessary for it to fulfill a number of additional functional and non-functional requirements.

Functional Requirements:

- Data Collection

  The text summarization model must be able to collect a substantial data set comprising numerous news articles, with the objective of enabling the model to be trained and to generate accurate and comprehensive summaries.

- Data Preprocessing

  The text summarization model should process the dataset by preprocessing the texts in the dataset in a manner consistent with its own algorithm or architecture.

- Feature Extraction

  The text summarization model should extract the features from the dataset that will be applied to train the model.

- Model Training

  The text summarization model that has been pre-trained should undergo fine-tuning with the data set in order to achieve satisfactory results.

- Model Evaluation

  The text summarization model must be evaluated using the selected evaluation metrics. This should include an assessment of the time taken for training the model, if applicable, and the generation of summaries from the dataset.

Non-functional Requirements:

- ROUGE Scores

  The text summarization model must have these scores to measure how much the summaries created by each tool overlap with summaries written by humans.

- BERT Score

  The text summarization model must have these scores to measure how similar the generated summaries are to the original articles in terms of meaning and overall content.

- Training Time

  The text summarization model must have its own training time so it can be compared to how long it takes each tool to learn from a set of news articles.
- Summarization Speed

  The text summarization model must have its own summarization speed so it can be compared to how quickly each tool can generate summaries for a group of unseen news articles.

1.2 Flowchart

In order to give an idea of how the model works, a general flow chart has been made for all the text summarization techniques (Figure 1), and each text summarization technique is explained in detail in its own activity diagram.

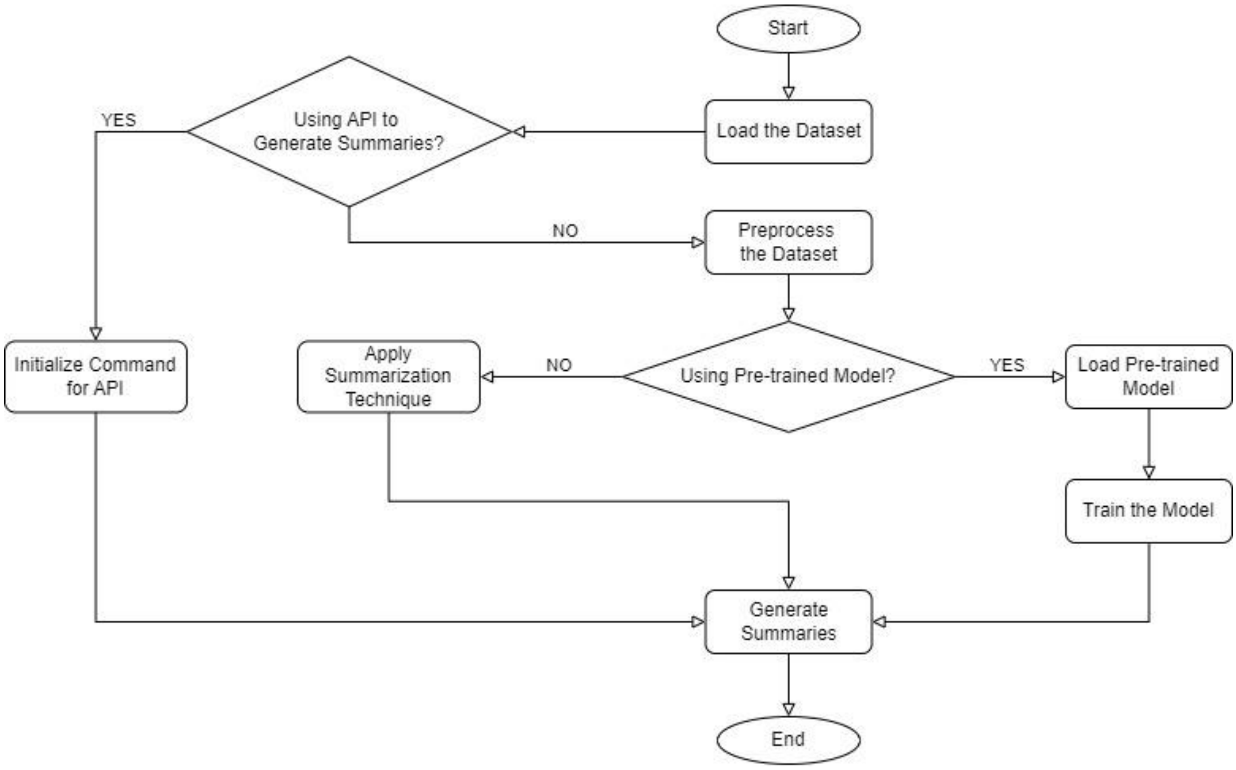


Figure 1 – General Flowchart of the Text Summarization Model

The flowchart illustrates the general design for each of the text summarization models. Starting from the first thing to do is load the dataset. In that step, the chosen dataset that will be used will be loaded, which is the CNN Dailymail Dataset [20], which is an English-language dataset containing just over 300k unique news articles as written by journalists at CNN and the Daily Mail. This step will be the starting point for every text summarization model, from the one with the simplest algorithm, Luhn, to the model that uses a pre-trained model, which is BART and Pegasus, and also for the one that uses an API, which is Claude LLM.

After successfully loading the dataset, the next step will be checking if the text summarization model uses an API to generate the summaries or not; if it doesn't use an API, then the next step will be preprocessing the dataset. Each text summarization model will need to preprocess the dataset by using its own algorithm; for Luhn, it can start by breaking the article into sentences and words and finishing it in their own way; for BART and Pegasus, it can start by adding a prefix to the dataset and then tokenize it. Meanwhile, if the text summarization model uses an API to generate the summaries, like Claude LLM, it will directly prepare the command that will be passed to Claude API to generate the summaries using the chosen dataset.

For the text summarization model that goes with the preprocessing of the dataset step, it will now go into the next step, where it will be determined if the text summarization model uses a pre-trained model or not. For the one that doesn't use a pre-trained model, which is Luhn, it will go to the step where the Luhn summarization technique or algorithm will be applied to the preprocessed dataset, where it will generate summaries by using the Luhn summarization technique.

Then, for the text summarization model that uses a pre-trained model, they will need to load their respective pre-trained model, where BART and Pegasus use their own suitable pre-trained model for this dataset, and after loading their dataset, they will need to train their model with the dataset, or it can be said as fine-tuning the model with the dataset, which is the CNN Dailymail Dataset. Finally, after fine-tuning the model, it can be used to generate summaries.

Overall, the flowchart above illustrates how each of the text summarization models works and shows the steps that will be needed to pass to finally generate the summaries. Each of the text summarization models has its own way, where the detailed steps of each text summarization model will be shown and explained in their respective activity diagrams below.

1.3 Activity Diagram

To explain this in a more detailed way, all of the activity diagrams below will explain all of the text summarization models in terms of how each of them started the preprocessing step until it generated the summaries from the dataset. Starting with Luhn Summarization that can be seen below (Figure 2).

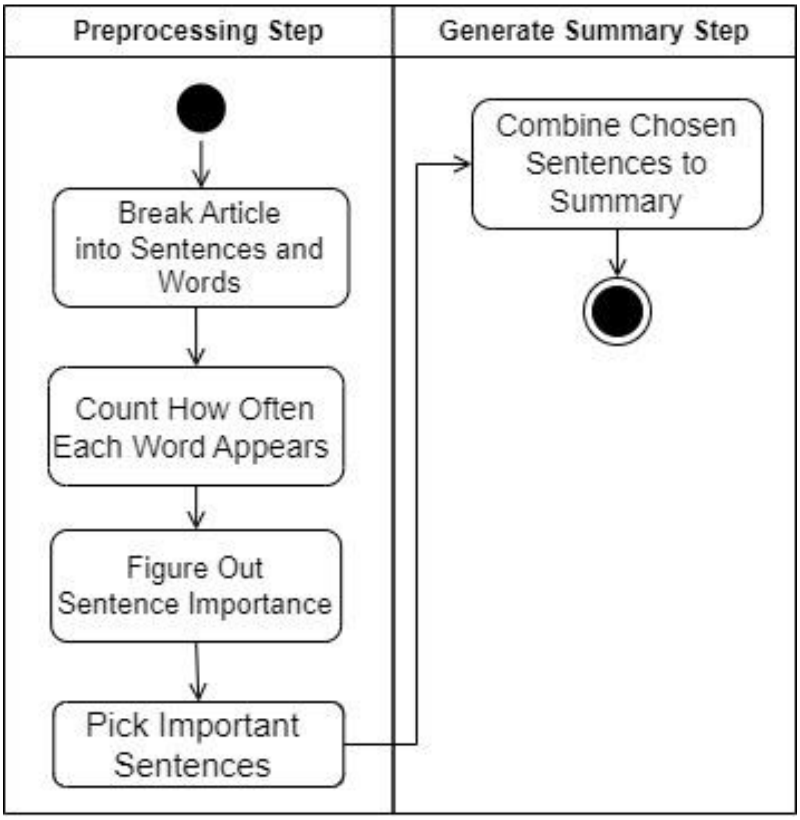


Figure 2 – Activity Diagram for Luhn Summarization

So, as can be seen it will be divided into two parts to explain the Luhn summarization technique. Starting with the preprocessing step, the dataset that has been loaded will be taken, and the article from it will then be broken into sentences and words. From there, those words will be inspected by counting how often each word from the article appears. After that, this technique will figure out "Sentence Importance", which is a sentence that is important or relevant in the making of the article. After it successfully figures it out, it will then take or pick the important sentences.

After successfully doing all of the mentioned steps, it will go to the next part, which is the generate summary step. Continuing the last step, which is picking the important sentences, it will be used by combining those sentences to generate the summary. Thus, we will end the process of the Luhn summarization technique, resulting in the summary that has been generated using all the mentioned steps. And then for the next activity diagram below is BART (Figure 3).

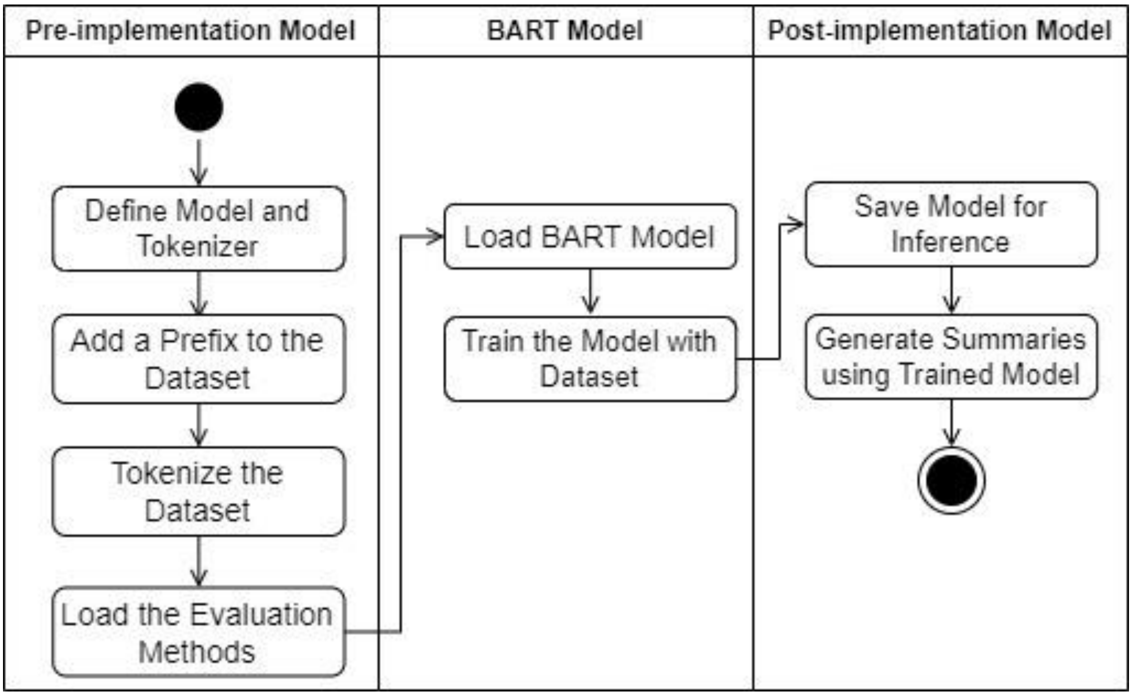


Figure 3 – Activity Diagram for BART

Here it will explain how BART generates summaries. It will be divided into three parts, which are the pre-implementation model, the BART model, and lastly, the post-implementation model. Starting in the pre-implementation, by first defining the model and tokenizer, which is the BART model that is suitable with the chosen dataset, and also the BART tokenizer that ensures compatibility with the BART model and provides access to BART-specific tokenization methods and parameters, which can be crucial for tasks like text summarization, after that going into the next step by adding a prefix, which is summarize, so BART knows this is a summarization task, then it will go through the next step, which is tokenizing the dataset, where in here preprocess the dataset, including tokenizing the dataset and determining the article as the input and the highlight as the target. After tokenizing and preprocessing the data, it will then prepare the evaluation methods (ROUGE and BERT) by loading them.

After successfully loading the evaluation methods, it will now go through the next part, where, after all those preparations are completed, it will now load the chosen BART model that's suitable, where it will choose a BART model that is specifically designed for conditional text generation tasks like summarization, translation, and text generation. Then, it will train the model with the dataset using selected training arguments by defining the training hyperparameters and also with the trainer class to evaluate it using the evaluation methods; this process can also be called fine-tuning the model.

When the training is done, it will indicate the fine-tuning process has succeeded, so it can go to the next part, which is the post-implementation model, where the trained model will be saved so it can be used to generate summaries. After successfully saving the model, it can now be loaded and used to generate summaries, thus finishing the process of making a text summarization model using BART. Following the activity diagram of BART (Figure 3), below diagram (Figure 4) is for Pegasus.

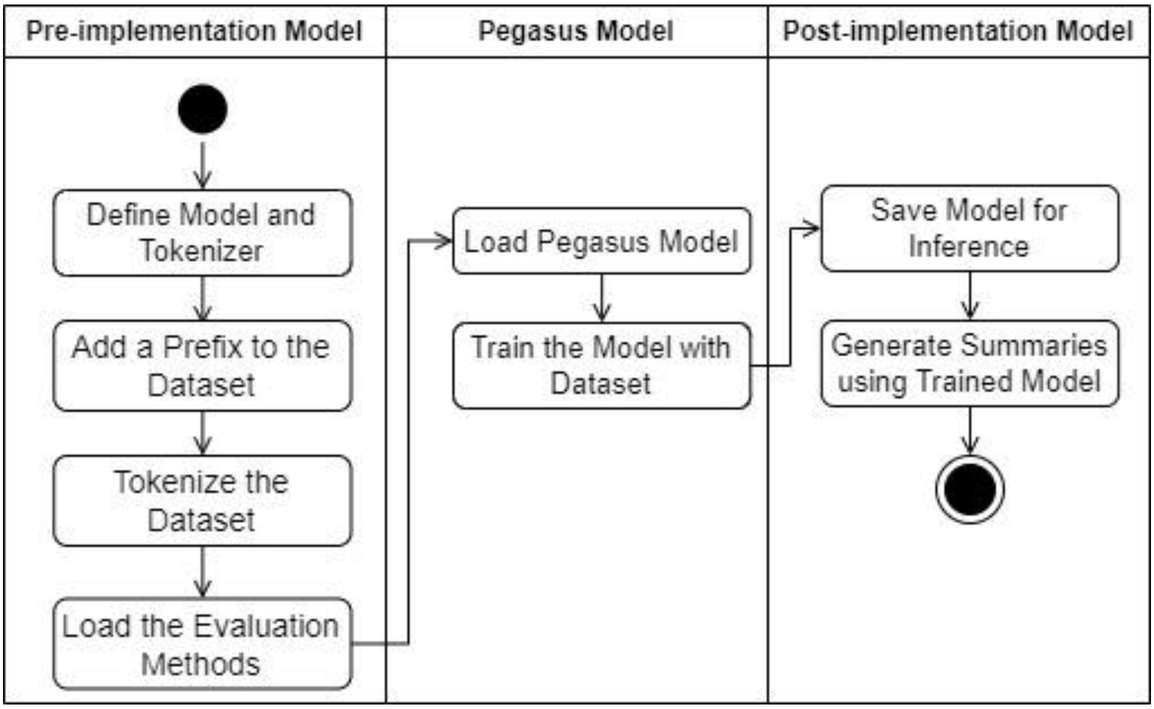


Figure 4 – Activity Diagram for Pegasus

For Pegasus, where it really looks similar to BART. It will also be divided into three parts, which are the pre-implementation model, the Pegasus model, and the post-implementation model. Also starting with defining the model and tokenizer, which in this case will be the model of Pegasus and its tokenizer that is compatible with the dataset, which can be specifically optimized for tokenizing input text for Pegasus models, which can result in better performance and compatibility. Then it will also be adding a prefix, which is summarize, to tell Pegasus it's a summarization task. It will also continue to tokenize and preprocess the dataset, like BART, where it will define the input, which is the article, and the target, which is the highlight. It will then also load the evaluation methods (ROUGE and BERT) that will be used.

Continuing the pre-implementation model, it will load the Pegasus model using a selected class that is tailored for conditional text generation tasks like summarization and provides a more convenient interface for generating summaries with Pegasus models. Then it will undergo the training process using selected training arguments by defining the training hyperparameters, and the trainer class will also evaluate it using the evaluation methods.

As before, the model that has been successfully trained will be saved to be used for generating summaries. The saved model will then be loaded and can generate summaries by using unseen data from the dataset or new data. Lastly, an activity diagram (Figure 5) for Claude LLM.

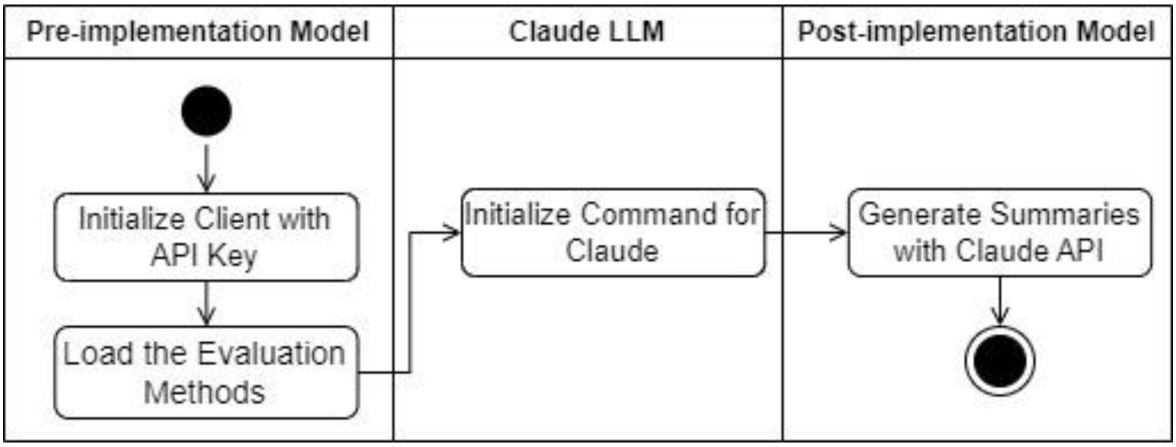


Figure 5 – Activity Diagram for Claude LLM

The last text summarization model or technique, which is Claude LLM, which will utilize the Claude API. This activity diagram (Figure 5) will also be divided into three parts, which are the pre-implementation model, the Claude LLM, and the post-implementation model. Starting with the pre-implementation model part, here the first step will be to initialize the client with an API key so it can access the Claude API, and the model that will be used will also be defined. Then, in the next step, the evaluation methods (ROUGE and BERT) will be loaded to evaluate the generated summaries.

After successfully preparing the pre-implementation of Claude LLM, the next step will be initializing or defining the command for Claude. In this step, a prompt or command to summarize the article will be needed to pass to the Claude API, so the Claude API knows what is needed to do.

Finally, after successfully initializing the command, the Claude API will generate the summaries from the dataset that have been passed by utilizing the command or function that has been initialized or defined.

2 Used Techniques, Technologies and Tools

To accomplish this paper, there are techniques for the text summarization that need to be chosen, which are Luhn, BART, Pegasus, and Claude LLM, including techniques to calculate the metrics, which are ROUGE and BERT. There will also be technologies and tools that are utilized for completing the task of this paper, such as programming language, a workplace to complete will also need to be determined so that the progress of scripting the code and visualizing its results will be much easier. That being said, Jupyter Notebook was chosen, and lastly, in the progress of completing this paper, PyTorch was chosen as the framework that will ease the progress of these machine learning tasks.

2.1 Jupyter Notebook

The Jupyter Notebook is the optimal platform for doing this research. Jupyter Notebook is a web-based interactive environment specifically designed for data science tasks [12]. Unlike traditional coding environments, Jupyter Notebook presents code, visualizations, and explanatory text within the same document, creating a "notebook" format that fosters a collaborative and exploratory workflow [21]. This interactive nature allows users to experiment with code snippets, visualize results in real-time, and document their findings all in one place.

Reasons for Choosing Jupyter Notebook:

- Clear and Documented Code

  Jupyter Notebooks will encourage users to write well-documented code with explanations alongside each code block. This promotes code readability and understanding, both for the researcher and anyone reviewing the work [22]. This is particularly important in research projects, where clear documentation facilitates replication and future improvements.
- Rapid Prototyping and Iteration

  Jupyter Notebook excels at rapid prototyping, allowing users to quickly test different code segments and visualize the results [23]. This is crucial in exploring various summarization techniques, as different configurations and approaches can be tested and compared with ease. This iterative process allows users to refine their code and optimize their summarization experiments.
- Integration with NLP Libraries

  Jupyter Notebook seamlessly integrates with popular NLP libraries in Python, such as NLTK and spaCy [24]. This allows users to leverage pre-built functions and modules directly within their notebooks, streamlining the development process and minimizing the need to write complex code from scratch.

- Interactive Visualization

  Jupyter Notebook supports various visualization libraries, enabling users to create informative plots and charts to represent their findings [25]. This visual representation allows for a deeper understanding of the summarization performance and facilitates clear communication of results in the thesis document.

In conclusion, Jupyter Notebook's interactive nature, emphasis on documented code, rapid prototyping capabilities, integration with NLP libraries, and support for visualization make it the ideal platform for exploring the capabilities of Luhn summarization, BART, Pegasus, and Claude LLM for news summarization. It fosters an iterative and exploratory research approach, while ensuring clear documentation and effective communication of the findings.

2.2 PyTorch

As also mentioned before, the framework that will be utilized in this paper is PyTorch as the foundation for building and training the various summarization models. PyTorch is a popular open-source deep learning framework renowned for its flexibility and ease of use [26]. Unlike some frameworks with rigid structures, PyTorch offers a more Pythonic approach, allowing users to write code that resembles natural language [24]. This characteristic makes PyTorch approachable for users with a strong Python background, facilitating rapid development and experimentation in deep learning projects.

Reasons for Choosing PyTorch:

- Dynamic Computational Graph

  PyTorch boasts a dynamic computational graph, which means the structure of the model is built during runtime [27]. This allows for greater flexibility in defining and modifying model architectures compared to static graph frameworks. This is particularly beneficial in research projects like this, where exploring different summarization techniques and architectures is crucial.
- Extensive Ecosystem of Libraries

  PyTorch thrives within a rich ecosystem of deep learning libraries [28]. Notably, libraries like PyTorch Text and TorchText provide pre-built modules and functionalities for text processing tasks like tokenization and text embedding, which are essential for building text summarization models. Utilizing these libraries streamlines the development process and allows users to focus on the core aspects of the summarization techniques.
- GPU Acceleration Support

  PyTorch seamlessly integrates with Graphics Processing Units (GPUs) for accelerated training of deep learning models [29]. This is particularly important for training large and complex summarization models, as GPUs significantly reduce training time. Efficient training is crucial for exploring various techniques and iterating on model designs within the timeframe of a research project.

In conclusion, PyTorch's combination of user-friendliness, dynamic computational graph, extensive library ecosystem, GPU acceleration support, and strong community support makes it the ideal framework for building and exploring text summarization models in this thesis. Its flexibility allows for rapid experimentation with various summarization techniques, while its rich ecosystem and documentation ensure efficient development and a smooth research process.

2.3 Luhn Summarization

Luhn summarization (1958) stands as a foundational technique known for its simplicity and efficiency, Luhn summarization operates under the assumption that important sentences in a document tend to have a higher frequency of unique terms [30]. By analyzing word frequency and assigning weights based on their importance, the technique aims to extract the most informative sentences, thereby creating a concise summary. Starting by outlining its step-by-step process and the mathematical formula it employs to identify key sentences in a document, as will be shown below:

- Preprocessing
  - The document is divided into individual sentences.
  - Words within each sentence are converted to lowercase (optional, for case-insensitivity).
  - Common words (stop words) like "the," "a," and "an" are removed (optional, to focus on content-rich words).
- Term Frequency (TF) Calculation
  - The frequency of each unique word (excluding stop words) is calculated within the entire document. This gives an initial indication of a word's potential importance.
- Term Weighting (TW)
  - A weight is assigned to each term based on its frequency. A common approach utilizes the following formula [31]:

$$TW(t) = (TF(t)\, log(\frac{N}{DF(t)})), \qquad (1)$$

where ***TW(t)*** – the term weight of term t, which represents the relative importance of a specific word (term) within the document;

TF(t) – the frequency of term t within the document, which refers to the number of times the specific word (term) t appears in the entire document;

N – the total number of words in the document (excluding stop words);

DF(t) – the document frequency of term t, which represents the number of sentences in which the specific word (term) t appears;

t – a specific word (term) within the document that the weight is being calculated.

This formula considers both the frequency of a term and its overall document distribution. Terms that appear frequently but only in a few sentences receive a higher weight compared to terms that appear frequently throughout the document. The concept of "significant words" aligns with this notion. These words can be high-frequency words, unique terms, or named entities that are critical to the document's content.

- Sentence Weighting
  - The weight of each sentence is calculated by summing the weights of the unique terms it contains. Sentences with more frequent and potentially informative terms (significant words) will naturally receive higher weights.
- Sentence Selection
  - Sentences are ranked based on their weight.
  - A predefined number of sentences with the highest weights are chosen to form the summary. These sentences are considered to be the most informative and capture the essence of the document.
- Summary Creation
  - The selected sentences are combined to create a concise summary of the original document.

Luhn summarization offers a straightforward and computationally inexpensive approach to summarization. However, it may struggle with documents that rely heavily on synonyms or complex sentence structures [32].

Luhn summarization, despite its simplicity, serves as a valuable baseline technique in the field of text summarization. It provides a foundation for understanding how word frequency and term weighting can be used to identify important content within a document. As we explore more advanced techniques in this thesis, the core principles of Luhn summarization will act as a reference point for appreciating the evolution of summarization methods.

2.4 BART

BART (Bidirectional and Autoregressive Transformer) stands out as a powerful contender. Unlike Luhn summarization, which relies on statistical methods to extract keywords, BART leverages a sophisticated neural network architecture to capture the intricacies of news articles and generate summaries that are both concise and informative. BART adopts a standard sequence-to-sequence (seq2seq) architecture, commonly used in machine translation tasks [1]. This architecture is comprised of two key components (Figure 6).

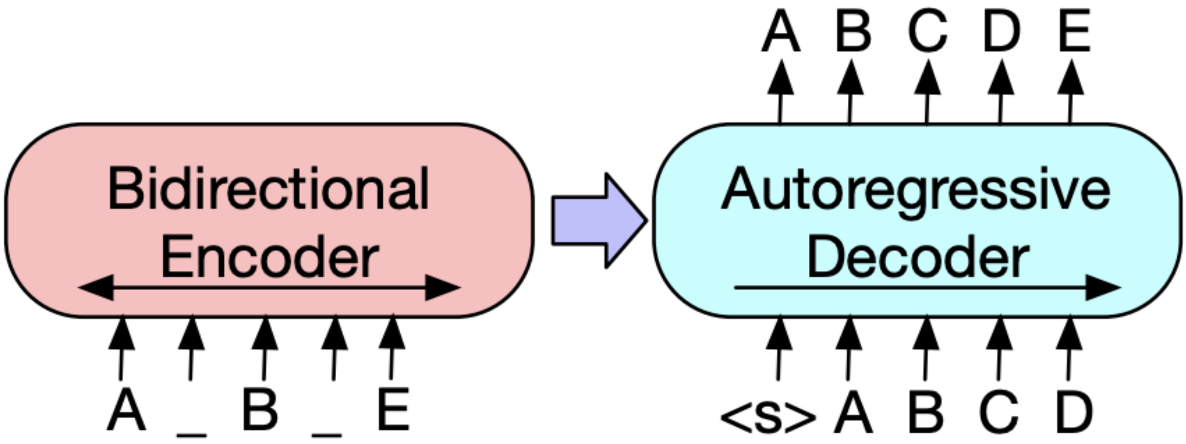


Figure 6 – Architecture of the BART Model with Encoder and Decoder Components [1]

- Bidirectional Encoder
  Inspired by the well-known BERT model (Bidirectional Encoder Representations from Transformers) [33], BART's encoder excels at understanding the complexities of the input text. Unlike traditional left-to-right encoders, BERT's bidirectional nature allows it to consider the entire input sequence, including words before and after a specific word. This holistic view empowers BART's encoder to capture the context and relationships between words, leading to a more comprehensive understanding of the news article's content.
- Autoregressive Decoder
  The decoder component, similar to GPT (Generative Pre-trained Transformer) [34], tackles the task of generating the news summary. It operates in an autoregressive manner, meaning it builds the summary one word at a time. However, unlike a simple left-to-right decoder, the autoregressive decoder considers the previously generated words in the sequence when making decisions about the next word. This iterative approach ensures that the decoder considers the overall context and coherence while crafting the summary.

To improve BART's robustness and ability to handle unseen data, the model is often trained using various noising techniques [1]. These techniques (Figure 7) essentially involve strategically introducing distortions or disruptions to the input data, forcing the model to learn and adapt.

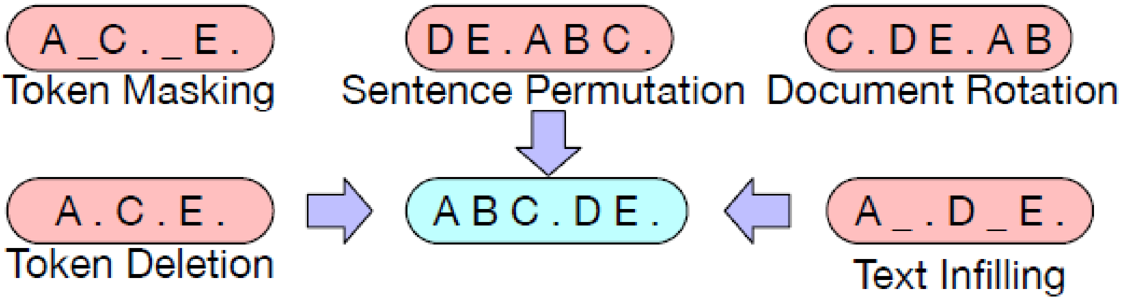


Figure 7 – Techniques for Noising the Input Data [1]

- Token Masking

  Similar to BERT, BART employs token masking [33]. Here, random words within the news article are replaced with a special "[MASK]" token. The model then attempts to predict the masked words, enhancing its capability to reconstruct missing information and comprehend the underlying context [1].
- Token Deletion

  This technique goes a step further by eliminating random words from the input text entirely [1]. During training, BART must not only predict the missing words but also determine the appropriate locations for their insertion within the sequence. This challenges the model to reason about the logical flow and information gaps within the news article.
- Text Infilling

  Inspired by SpanBERT [35], text infilling involves selecting text spans of varying lengths from the news article and replacing them with a single "[MASK]" token [1]. By learning to predict the content of these masked spans, BART strengthens its ability to handle information gaps and summarize even when faced with incomplete or challenging news articles.
- Sentence Permutation

  This strategy disrupts the inherent order of the news article by randomly shuffling the sentences [1]. The model then attempts to reconstruct the original sequence during training. This process bolsters BART's understanding of sentence relationships and narrative flow, crucial aspects for generating coherent news summaries.
- Document Rotation

  Here, a random starting point is chosen within the news article, and the remaining text is rotated to begin at that point [1]. The model then learns to identify the most logical starting point for the summary, ensuring its conciseness and relevance to the core message of the news article.

By incorporating these noising techniques, BART is exposed to various challenges during training. This process equips the model with the ability to handle imperfections and ambiguities that might be present in real-world news data, ultimately leading to more robust and informative news summaries.

2.5 Pegasus

Beside BART, which stands out as a powerful contender as a text summarization model, Pegasus came next as a compelling contender. Unlike statistical methods like Luhn summarization, Pegasus leverages a sophisticated neural network architecture to capture the intricacies of news articles and generate summaries that are both concise and informative.

Pegasus adopts a standard Transformer encoder-decoder architecture, commonly used in various natural language processing tasks [36]. This architecture (Figure 8) comprises two key components.

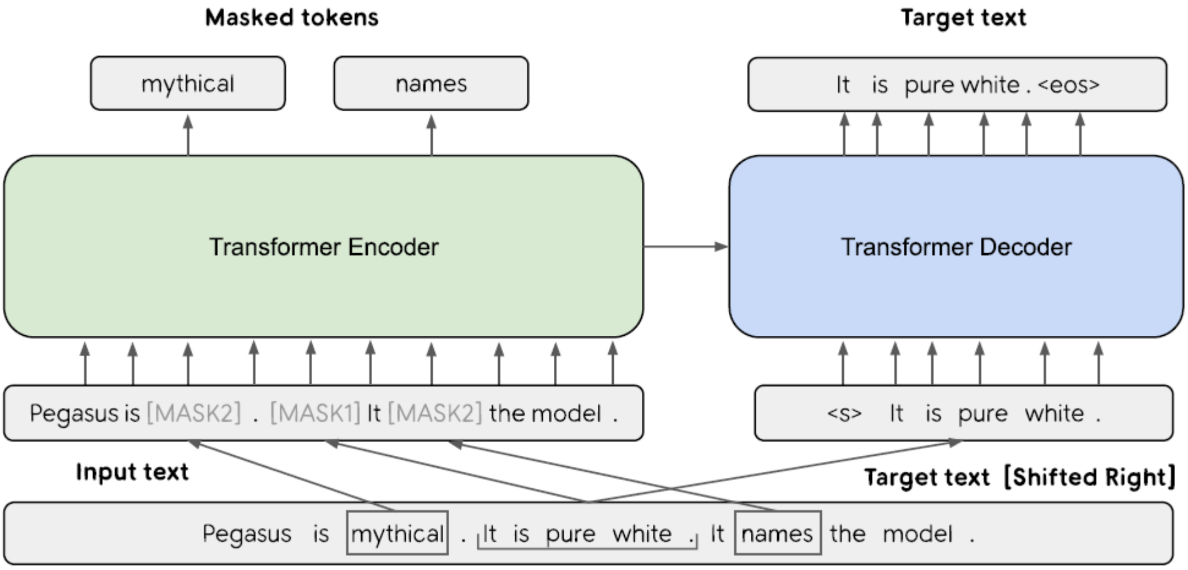


Figure 8 – Base Architecture of the Pegasus model [36]

- Transformer Encoder

  The encoder's function is to analyze and understand the input text, which is typically a news article in our case. Unlike traditional encoders that process text sequentially (word by word), the Transformer encoder employs a powerful technique called self-attention [37]. Self-attention allows the encoder to consider the entire input sequence simultaneously, capturing the relationships between words regardless of their order. This holistic approach empowers Pegasus to grasp the context and dependencies within the news article, leading to a more comprehensive understanding of its content.

- Transformer Decoder

  Once the encoder has analyzed the news article, the decoder takes center stage. Its responsibility is to generate the news summary. Similar to the encoder, the decoder also utilizes self-attention to consider the relationships between words as it builds the summary one word at a time. However, the decoder additionally incorporates an attention mechanism that allows it to focus on specific parts of the encoded input (the news article). This ensures that the decoder generates a summary that is relevant and reflects the key points of the original text.

To enhance its summarization abilities, Pegasus is trained using two main objectives, which are Gap Sentences Generation and Masked Language Modeling, and they will be detailed below.

- Gap Sentences Generation (GSG)

  This objective focuses on the decoder's capability to generate summaries from incomplete information. During training, a sentence from the news article is masked entirely (replaced with a special token), essentially creating a "gap" in the text. The decoder is then tasked with using the remaining context (surrounding sentences) to generate a summary that incorporates the missing information from the masked sentence [36]. This strategy strengthens Pegasus' ability to handle complex news articles where certain details might be implicit or require inference.

- Masked Language Modeling (MLM)

  This objective tackles the task of predicting missing words within the context of a sentence [33]. During training, random words throughout the news article are masked (replaced with a special token). The encoder and decoder work together to predict the masked words, forcing them to learn the relationships between words and improve their overall understanding of language. This process hones Pegasus' ability to deal with ambiguities or phrased information that might be present in real-world news articles.

By incorporating both GSG and MLM objectives during training, Pegasus is exposed to various challenges that mimic real-world news data. This comprehensive training approach equips Pegasus with the ability to not only grasp the core content of news articles but also generate summaries that are informative and address potential information gaps.

2.6 Claude LLM

Lastly, as the last solution of text summarizer model in the exploration of various Large Language Models (LLMs) for news summarization in this paper, as a recent arrival with bright future, Claude LLM stands out. Developed by Anthropic AI, Claude LLM is designed to be helpful, honest, and harmless in its interactions [38].

Currently, Claude LLM is primarily accessible through its API (Application Programming Interface). This API allows users to interact with the model and leverage its summarization abilities for users' news articles, in this case. Here's a general process for how to utilize Claude LLM's API according to the task of this paper.

- API Access
  To get started, an API key from Anthropic AI is required, and this key acts as authorization to interact with Claude LLM's services.
- Data Preparation
  Prepare the news articles in a format compatible with Claude LLM's API. This might involve converting the text into a specific format or including any necessary metadata.
- API Call
  Once the data and API key are ready, make an API call to Claude LLM. This call typically involves sending the news article text and specifying the desired summary length or other parameters.
- Response Retrieval
  Claude LLM's API will process the requests that have been sent and return a response containing the generated summary of the news article.

2.7 ROUGE

After exploring all the chosen techniques for the research of this paper, some evaluation methods are chosen, and for the first one, a popular metric which is ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [3]. ROUGE measures how similar a generated summary is to high-quality human-written references (original articles).

ROUGE offers various metrics that assess overlap between the generated summary and reference text using different n-gram lengths (sequences of n consecutive words). And for evaluating the text summarization model in this paper, ROUGE-1, ROUGE-2, and ROUGE-L are chosen to be utilized, and below is more detailed information about the chosen ROUGE variants.

- ROUGE-1 (Unigram Overlap)
  This metric focuses on individual words (unigrams) that appear in both the summary and the reference. The higher the number of matching unigrams, the better the summary is considered to capture the key content. Here is an example to make it easier to understand:
  Original Article: "The fox was found on the tree over there."
  Original Article Words: "The," "fox," "was," "found," "on," "the," "tree," "over," "there," (9 words).

Generated Summary: "The fox is on that tree."

Generated Summary Words: "The," "fox," "is," "on," "that," "tree," (6 words).

Overlapping Words: "The," "fox," "on," "tree," (4 words).

- ROUGE-2 (Bigram Overlap)

  This metric considers pairs of consecutive words (bigrams) present in both the summary and the reference. It emphasizes capturing phrasal information beyond individual words. Here is an example to make it easier to understand:

  Original Article: "The fox was found on the tree over there."

  Original Article Pairs: "The fox," "fox was," "was found," "found on," "on the," "the tree," "tree over," " over there," (8 pairs).

  Generated Summary: "The fox is on that tree."

  Generated Summary Pairs: "The fox," "fox is," "is on," "on that," "that tree," (5 pairs).

  Overlapping Pairs: "The fox," (1 pair).

- ROUGE-L (Longest Common Subsequence)

  This metric identifies the longest sequence of words that appears in both the summary and the reference. It rewards summaries that capture the overall flow and factual details from the original article. Here is an example to make it easier to understand:

  Original Article: "The fox was found on the tree over there."

  Original Article Words: "The," "fox," "was," "found," "on," "the," "tree," "over," "there," (9 words).

  Generated Summary: "The fox is on that tree."

  Longest Common Subsequence: "The fox ~~is~~ on ~~that~~ tree" → "The fox on tree"(4 words).

After understanding the concept of each of the ROUGE variants that are chosen for evaluating the text summarization model, let's dig down to how they really calculate each of the ROUGE variants. And there are three key metrics that will be used, which are these.

- ROUGE Precision

  This measures the proportion of n-grams in the generated summary that are also present in the reference. A high precision indicates the summary is concise and avoids irrelevant information.

$$ROUGE_{Precision} = \frac{\#\ overlapping\ n\text{-}grams}{total\ n\text{-}grams\ in\ generated\ summary} \quad (2)$$

- ROUGE Recall

  This measures the proportion of n-grams in the reference that are also present in the generated summary. A high recall indicates the summary captures most of the important information from the original article.

$$ROUGE_{Recall} = \frac{\#\ overlapping\ n\text{-}grams}{total\ n\text{-}grams\ in\ original\ article} \quad (3)$$

- ROUGE F1-Score

  This is a harmonic mean of precision and recall, providing a balanced view of summary quality.

$$ROUGE_{F1-Score} = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (4)$$

2.8 BERT Score

Beside ROUGE, other evaluation methods will be utilized to evaluate the results of the text summarization model. The other choice of evaluation method is BERT Score, leveraging the power of the Bidirectional Encoder Representations from Transformers (BERT) pre-trained model [39]. Unlike ROUGE, which focuses on n-gram overlap, BERT Score takes a semantic similarity approach, which can be seen below (Figure 9).

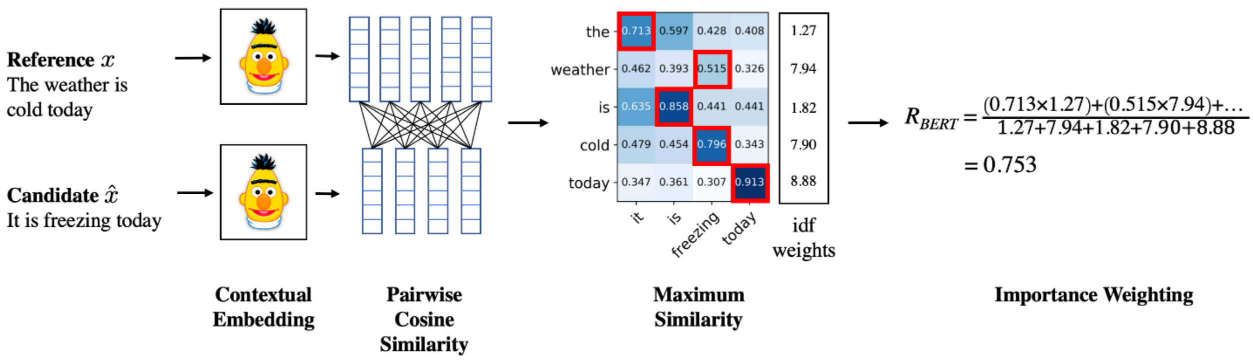


Figure 9 – Illustration of the Computation of the Recall Metric [39]

- Contextual Embeddings
  BERT Score utilizes a pre-trained BERT model to generate contextual embeddings for both the generated summary and the original article (reference text). These embeddings capture the meaning of words in context, similar to how humans understand language.
- Pairwise Cosine Similarity
  BERT Score then calculates the cosine similarity between each sentence in the summary and every sentence in the reference text. Cosine similarity is a measure of how similar two vectors (embeddings) are, reflecting the semantic closeness between sentences.
- Maximum Similarity
  For each sentence in the summary, BERT Score identifies the sentence in the reference text with the highest cosine similarity. This essentially finds the closest match in terms of meaning for each sentence in the summary.
- Importance Weighting (Optional)
  BERT Score can optionally incorporate importance weighting. This assigns higher weights to words or phrases deemed more critical for understanding the sentence. This step can further refine the semantic similarity score.

After understanding the concept of each of the BERT scores, let's dig down to how they really calculate each of the BERT scores. Like ROUGE, there will be three key metrics that will be used, which are these.

- BERT Precision
  This measures the proportion of sentences in the summary that are semantically similar to sentences in the reference text. High precision indicates the summary avoids irrelevant information and focuses on content present in the original article.

$$P_{BERT} = \frac{\#\, correct\, YES\, predictions}{total\, \#\, of\, YES\, predictions} \tag{5}$$

- BERT Recall
  This measures the proportion of sentences in the reference text that have a semantically similar counterpart in the summary. High recall suggests the summary captures most of the important information from the original article.

$$R_{BERT} = \frac{\#\, correct\, YES\, predictions}{total\, \#\, of\, actual\, YES\, instances} \tag{6}$$

- BERT F1-Score

  Similar to ROUGE F1-Score, this metric provides a balanced view by considering both precision and recall.

$$F_{BERT} = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{7}$$

3 Implementations

3.1 Luhn Summarization

The first text summarization model that will be explored is the Luhn summarization technique, whose detailed technique can be seen in the previous chapter. In this section, the implementation of Luhn summarization will be detailed. Starting with loading the dataset, which will use the CNN Dailymail dataset [20] that has been provided by Hugging Face.

After loading the data, the next step to do is to define a function to apply the Luhn summarization technique, as can be seen in the code snippet below (Code Sample 1).

```
def apply_luhn_summarization(article, num_sentences):
    # Tokenize the article into sentences
    sentences = sent_tokenize(article)

    # Tokenize the entire article into words
    words = word_tokenize(article)

    # Calculate word frequencies using Counter
    word_frequencies = Counter(words)

    # Initialize a dictionary to store sentence scores
    sentence_scores = {}

    # Calculate scores for each sentence
    for sentence in sentences:
        # Tokenize the sentence into words
        sentence_words = word_tokenize(sentence)

        # Check the frequency of each word in the original case
        for word in sentence_words:
            # Check if the word is in the word frequencies
            if word in word_frequencies:
                # If the sentence is not in the scores dictionary, initialize it
                if sentence not in sentence_scores:
                    sentence_scores[sentence] = word_frequencies[word]
                else:
                    # If the sentence is already in the dictionary, add the word's frequency
                    sentence_scores[sentence] += word_frequencies[word]

    # Get the top 'num_sentences' sentences with the highest scores
    summary_sentences = nlargest(num_sentences, sentence_scores, key=sentence_scores.get)

    # Combine the selected sentences to create the summary
    summary = ' '.join(summary_sentences)

    return summary
```

Code Sample 1 – Function to Apply Luhn Summarization Technique

So, as can be seen in the above script (Code Sample 1), it will take the article from the dataset and how many sentences are to be generated as the summarization parameter that will pass on to the Luhn summarization technique. What happens there? As has been mentioned in the activity diagram, the article will be tokenized first into sentences and then words. Then, it will calculate the frequencies of the words and initialize a dictionary to score the sentence score. After that, what will happen is that it will calculate the score for each sentence to look for the top or most important sentence that will be used as the summary. After calculating, it can be seen that after that, it will get the sentence with the highest score or the most important one to be combined as the summary.

Next, after defining the function for applying the Luhn summarization, a function to evaluate or calculate the ROUGE and BERT scores for Luhn will be needed; thus, firstly, the scorer for BERT will be defined like below (Code Sample 2).

```
bert_model = "bert-base-uncased"
scorer = BERTScorer(lang="en", model_type=bert_model)
tokenizer = AutoTokenizer.from_pretrained(bert_model)
```

Code Sample 2 – Define BERT as the Evaluation Metric for Luhn Summarization

After defining the BERT as one of the evaluation metrics, a function for evaluating the summary will be made, as can be seen below (Code Sample 3), consisting of the calculation of the ROUGE scores combined with the results of the calculation of the BERT scores.

```python
def calculate_rouge_and_bert_scores(df, scorer, tokenizer):
    # Initialize the ROUGE scorer
    rouge = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL'], use_stemmer=True)

    # Lists to store the ROUGE and BERT scores
    rouge1_scores = []
    rouge2_scores = []
    rougeL_scores = []
    bert_precision_scores = []
    bert_recall_scores = []
    bert_f1_scores = []

    for _, row in df.iterrows():
        reference_summary = row['highlights']
        luhn_summary = row['luhn_summary']

        # Check if reference_summary or luhn_summary is None or empty
        if reference_summary is None or luhn_summary is None or reference_summary == "" or luhn_summary == "":
            # Append 0.0 for precision, recall, and fmeasure
            rouge1_scores.append({'precision': 0.0, 'recall': 0.0, 'fmeasure': 0.0})
            rouge2_scores.append({'precision': 0.0, 'recall': 0.0, 'fmeasure': 0.0})
            rougeL_scores.append({'precision': 0.0, 'recall': 0.0, 'fmeasure': 0.0})
            bert_precision_scores.append(0.0)
            bert_recall_scores.append(0.0)
            bert_f1_scores.append(0.0)
        else:
            # Calculate ROUGE scores
            rouge_scores = rouge.score(luhn_summary, reference_summary)

            # Calculate BERT score
            bert_results = scorer.score([luhn_summary], [reference_summary])

            # Extract individual precision, recall, and F1 scores
            bert_precision = round(bert_results[0].mean().item(), 4)
            bert_recall = round(bert_results[1].mean().item(), 4)
            bert_f1 = round(bert_results[2].mean().item(), 4)

            # Append the F1 score for each ROUGE metric
            rouge1_scores.append(round(rouge_scores['rouge1'].fmeasure, 2))
            rouge2_scores.append(round(rouge_scores['rouge2'].fmeasure, 2))
            rougeL_scores.append(round(rouge_scores['rougeL'].fmeasure, 2))
            bert_precision_scores.append(bert_precision)
            bert_recall_scores.append(bert_recall)
            bert_f1_scores.append(bert_f1)
```

Code Sample 3 – Function to Calculate ROUGE and BERT Scores for Luhn Summarization

From the code above (Code Sample 3), it could be seen that it will first define the ROUGE scorer, then create a list to store the results of ROUGE and BERT. Then, it will iterate through the article summary and its highlights to calculate the ROUGE and BERT scores, and after getting the results of each metric, it will be appended to the list that has been made, so then it can be shown.

After finishing preparing the function to be used for the Luhn summarization technique, the dataset can finally be applied to the technique, as shown below (Code Sample 4).

```python
# Start measuring time
start_time = time.time()

# Apply Luhn Method to each article in the test dataset and store summaries
dataset_test['luhn_summary'] = dataset_test['article'].apply(lambda x: apply_luhn_summarization(x, num_sentences=3))

# Calculate ROUGE and BERT scores for test dataset
calculate_rouge_and_bert_scores(dataset_test, scorer, tokenizer)

# End measuring time
end_time = time.time()

# Calculate runtime
runtime = end_time - start_time

# Format runtime
formatted_runtime = "{:.4f}".format(runtime)
print("Runtime:", formatted_runtime, "seconds")
```

Code Sample 4 – Apply the Luhn Summarization and Get the ROUGE and BERT Scores

As can be seen above (Code Sample 4), firstly, to be used as a comparison, a variable to measure the running time of applying the Luhn summarization technique and calculating the ROUGE and BERT scores is defined, and after starting the measuring time, it can be seen that the Luhn summary technique is applied to the dataset, then after it finishes getting the summary, the ROUGE and BERT scores will be calculated, and after all of the process is finished, the measuring time will finish. Lastly, the results of the evaluation metrics (Code Sample 5 and Figure 10).

```
Rouge1 Scores: 0.19
Rouge2 Scores: 0.09
RougeL Scores: 0.13
BERT Precision Scores: 0.4657
BERT Recall Scores: 0.5985
BERT F1 Scores: 0.5238
Runtime: 1900.7265 seconds
```

Code Sample 5 – Results of ROUGE and BERT Scores with its Runtime for Luhn Summarization

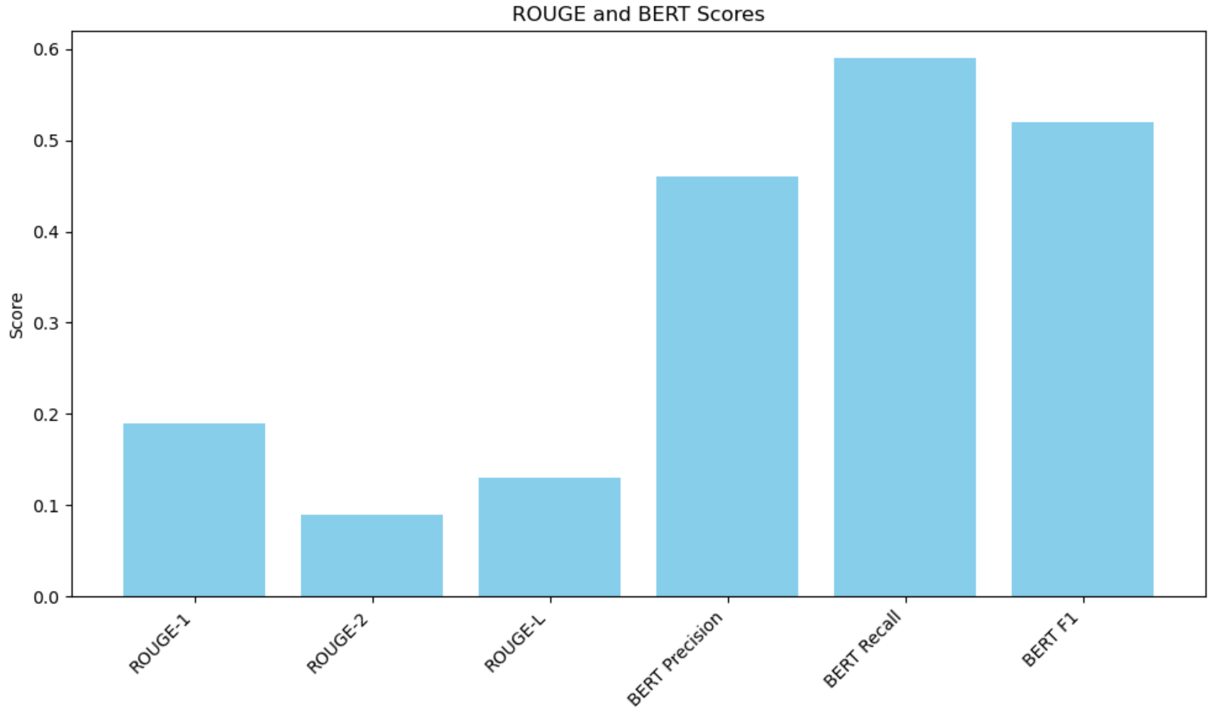


Figure 10 – Visualization of ROUGE and BERT Scores for Luhn Summarization

Some of the summarization can be seen below (Table 1) and for the original article that have been used can be seen in the Appendix (Table A.1).

Table 1 – Summary and Metrics Scores for Luhn Summarization

Summary	ROUGE and BERT Scores
The attack in Kenya killed 142 students, three security officers and two university security personnel, and was the nation's deadliest since the bombing of the U.S. Embassy in 1998. As Kenyans mourned those killed last week in one of the deadliest terrorist attacks in the nation, citizens used social media to share the victims' stories, hopes and dreams. Using the hashtag #147notjustanumber -- a reference to the number of people, mostly students, killed at Garissa University College on Thursday -- Kenyans tweeted pictures of the victims in happier times.	ROUGE-1: 0.20 ROUGE-2: 0.16 ROUGE-L: 0.13 BERT Precision: 0.57 BERT Recall: 0.13 BERT F1: 0.20

End of Table 1

Summary	ROUGE and BERT Scores
Five Americans who were monitored for three weeks at an Omaha, Nebraska, hospital after being exposed to Ebola in West Africa have been released, a Nebraska Medicine spokesman said in an email Wednesday. They all had contact with a colleague who was diagnosed with the disease and is being treated at the National Institutes of Health in Bethesda, Maryland. More than 10,000 people have died in a West African epidemic of Ebola that dates to December 2013, according to the World Health Organization.	ROUGE-1: 0.25 ROUGE-2: 0.20 ROUGE-L: 0.22 BERT Precision: 0.70 BERT Recall: 0.05 BERT F1: 0.30
He's been charged with engaging in conduct in preparation of acts of terrorism, and with engaging in conduct with the intention of assisting others to commit acts of terrorism. Yahya Rashid, a UK national from northwest London, was detained at Luton airport on Tuesday after he arrived on a flight from Istanbul, police said. A 19-year-old man was charged Wednesday with terror offenses after he was arrested as he returned to Britain from Turkey, London's Metropolitan Police said.	ROUGE-1: 0.35 ROUGE-2: 0.29 ROUGE-L: 0.33 BERT Precision: 0.64 BERT Recall: 0.21 BERT F1: 0.42
Arsenal have been in superb form since the start of the year, transforming what looked to be another mediocre season struggling to secure fourth place -- and with it Champions League qualification -- into one where they at least have a shot at winning the title. After going ahead, Arsenal rarely looked in any danger of conceding, showing more of the midfield pragmatism epitomized by the likes of Francis Coquelin, who also played a crucial role in the goal. Belgian international Christian Benteke scored the only goal of the game, his eighth in six matches, to secure a vital three points to give the Midlands club breathing space.	ROUGE-1: 0.12 ROUGE-2: 0.10 ROUGE-L: 0.11 BERT Precision: 0.64 BERT Recall: 0.25 BERT F1: 0.16
But there were good signs, as seen on the live stream and Dallas Zoo's Twitter feed -- like its ears moving, its efforts to stand, and its nursing (or at least trying to nurse) from mom. But the giraffe definitely did have watchers in the form of fellow giraffes who saw the scene unfold from an abutting barn, one of them being Katie's BFF Jade. The zoo describes her as the "diva" among a herd of 12 giraffes at the zoo who loves to "toss her head around" when she doesn't like something.	ROUGE-1: 0.19 ROUGE-2: 0.12 ROUGE-L: 0.12 BERT Precision: 0.54 BERT Recall: 0.15 BERT F1: 0.17

3.2 BART

Up next is BART, first thing to do of course loading the dataset, and how to do is the same as before, so now just go to the next step, because BART uses pre-trained model, the BART model with its tokenizer will be needed to define, as can be seen as below is how to define it (Code Sample 6).

```
model_checkpoint = "facebook/bart-base"
tokenizer = BartTokenizer.from_pretrained(model_checkpoint)
```

Code Sample 6 – Define BART Model and Tokenizer

After defining the BART model that will be used and its corresponding tokenizer to help process the dataset so it can be fine-tuned with BART, a prefix variable will be needed to tell BART that this is a summarization task, so then it can be passed to a function that will be defined next, which is a preprocess function, where the processing of the article will happen, like what has been shown below (Code Sample 7).

```
prefix = "summarize: "

def preprocess_function(examples):
    # Add a prefix to each document in the "article" column
    # The prefix might be useful for the model to understand that it's dealing with document summaries
    inputs = [prefix + doc for doc in examples["article"]]

    # Tokenize the modified documents using the tokenizer
    model_inputs = tokenizer(inputs, max_length=256, truncation=True)

    # Tokenize the "highlights" column to create labels for the model
    labels = tokenizer(text_target=examples["highlights"], max_length=32, truncation=True)

    # Include the tokenized labels (input_ids) in the model_inputs dictionary
    # This is necessary for training the model with the token-level classification objective
    model_inputs["labels"] = labels["input_ids"]

    return model_inputs
```

Code Sample 7– Define Prefix and Preprocessing Function

When the preprocessing function is defined, the next step is to use it to tokenize the dataset by applying the function below (Code Sample 8) to the dataset, be it training, testing, or validation, and also to make it more effective by dynamically padding the sentences to the longest length in a batch during the collection or training process instead of padding the whole dataset to the maximum length.

```
tokenized_dataset = dataset.map(preprocess_function, batched=True)
data_collator = DataCollatorForSeq2Seq(tokenizer=tokenizer, model=model_checkpoint)
```

Code Sample 8 – Preprocess the Dataset

After successfully preprocessing the dataset, the next step is to prepare the evaluation metrics (Code Sample 9), which are ROUGE and BERT, by loading them and defining the function (Code Sample 10) to compute those metrics with the predictions or summaries.

```
# Load a evaluation method (ROUGE) with the Evaluate library
rouge = evaluate.load("rouge")

# Define BERT model and scorer
bert_model = "bert-base-uncased"
scorer = BERTScorer(lang="en", model_type=bert_model)
```

Code Sample 9 – Prepare the Evaluation Metrics

```
def compute_metrics(eval_pred):
  # Unpack the tuple containing predictions and labels
  predictions, labels = eval_pred

  # Decode the predicted sequences, skipping special tokens
  decoded_preds = tokenizer.batch_decode(predictions, skip_special_tokens=True)

  # Replace label values of -100 (masked tokens) with pad token id
  labels = np.where(labels != -100, labels, tokenizer.pad_token_id)

  # Decode the labels, skipping special tokens
  decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)

  # Calculate ROUGE scores
  rouge_results = rouge.compute(predictions=decoded_preds, references=decoded_labels, use_stemmer=True)

  # Compute average length of generated predictions
  prediction_lens = [np.count_nonzero(pred != tokenizer.pad_token_id) for pred in predictions]
  rouge_results["gen_len"] = np.mean(prediction_lens)

  # Calculate BERT score
  bert_results = scorer.score(decoded_preds, decoded_labels)

  # Extract individual precision, recall, and F1 scores
  bert_precision = round(bert_results[0].mean().item(), 4)
  bert_recall = round(bert_results[1].mean().item(), 4)
  bert_f1 = round(bert_results[2].mean().item(), 4)

  # Combine ROUGE and BERT scores into a single dictionary
  combined_scores = {**rouge_results, **{
      "bert_precision": bert_precision,
      "bert_recall": bert_recall,
      "bert_f1": bert_f1
  }}

  # Round the scores to 4 decimal places and return the dictionary
  return {k: round(v, 4) for k, v in combined_scores.items()}
```

Code Sample 10 – Function to Calculate ROUGE and BERT Scores

After that, the next step is to load the BART model by utilizing a class that is suitable for loading the model, like shown below (Code Sample 11).

```python
model = BartForConditionalGeneration.from_pretrained(model_checkpoint)
```

Code Sample 11 – Load BART Model

Then, after loading the model, it's time to fine-tune the BART model with the dataset by defining the training hyperparameters and also the Trainer class (Code Sample 12) that will evaluate the ROUGE and BERT metrics.

```python
args = Seq2SeqTrainingArguments(
  output_dir='bart-cnn',
  evaluation_strategy="epoch",
  learning_rate=5e-5,
  per_device_train_batch_size=8,
  per_device_eval_batch_size=4,
  weight_decay=0.01,
  save_total_limit=2,
  num_train_epochs=8,
  gradient_accumulation_steps=4,
  predict_with_generate=True,
  report_to="wandb",
  run_name = "bart-cnn-training"
)

trainer = Seq2SeqTrainer(
    model,
    args,
    train_dataset=tokenized_dataset["train"],
    eval_dataset=tokenized_dataset["test"],
    data_collator=data_collator,
    tokenizer=tokenizer,
    compute_metrics=compute_metrics,
)
```

Code Sample 12 – Training Hyperparameters and Trainer Class for BART

After many experiments with the value for each of the hyperparameters, those (Code Sample 12) are the final values that are chosen, and below are the personal reasons for choosing those values after all the experiments.

- Evaluation Strategy (*epoch*)

  Setting the evaluation strategy to *epoch* is a common thing in machine learning tasks; it can simplify the task of monitoring the results of the model's performance.

- Learning Rate (*5e-5*)

  Choosing *5e-5* as the learning rate is the best way for this training process because it's a moderate value that balances the step size of the parameter, so the learning speed of this model will be normal and not too fast or slow in terms of this training process.
- Batch Size (*train = 8* and *evaluation = 4*)

  Setting the batch size of the *train* to *8* and the *evaluation* to *4* suits the best for this model, with those values, the training process can process more samples, which may lead to faster training time even though it may lead to memory constraints, and with smaller evaluations to help reduce memory usage so it can balance the training process and keep it working.
- Weight Decay (*0.01*)

  With *0.01* as the value, it strikes a balance in the regularization technique that helps prevent overfitting during training. With a moderate weight decay, it will encourage the model to learn a simple pattern and avoid a complex one, making the model learn in a less complex way.
- Save Total Limit (*2*)

  For saving the total limit, setting it to *2* is the most convenient way personally, so it will only save 2 checkpoints during training to avoid consuming too much disk space.
- Number of Epochs Trained (*8*)

  The number of epochs trained is best determined by the experiment; for some of the experiments where other values are tried to be assigned, assigning *8* as the value shown as the best score for this model.
- Gradient Accumulation Steps (*4*)

  Assigning *4* as the value suits the best for this model. With the chosen batch size and the device's memory, this value effectively simulates the chosen batch size without requiring additional memory, allowing it to be more stable for the training process.
- Predict with Generate (*true*)

  Giving *true* as the value for this parameter is necessary, and because of it, it allows the model to generate summaries directly during the evaluation process.

After finishing the training progress of the BART model with the dataset or fine-tuning it and evaluating it with the chosen metrics, the results are ready and can now be shown below (Code Sample 13), and because of using epoch as the strategy for evaluating it, a visualization of the epoch for ROUGE (Figure 11, Figure 12, and Figure 13) and BERT (Figure 14, Figure 15, and Figure 16) can also be seen.

Rouge1	Rouge2	Rougel	Bert Precision	Bert Recall	Bert F1
0.302400	0.141800	0.259500	0.614600	0.527100	0.566900
0.298800	0.129900	0.253800	0.612300	0.528300	0.566500
0.306600	0.145800	0.262500	0.614400	0.527000	0.566700
0.296000	0.130700	0.244900	0.606400	0.520900	0.559700
0.305800	0.135000	0.254000	0.611900	0.527700	0.566000
0.304100	0.139400	0.253500	0.612400	0.527800	0.566200
0.310700	0.135100	0.254900	0.614900	0.529300	0.568100

Code Sample 13 – Results of ROUGE and BERT Scores for BART

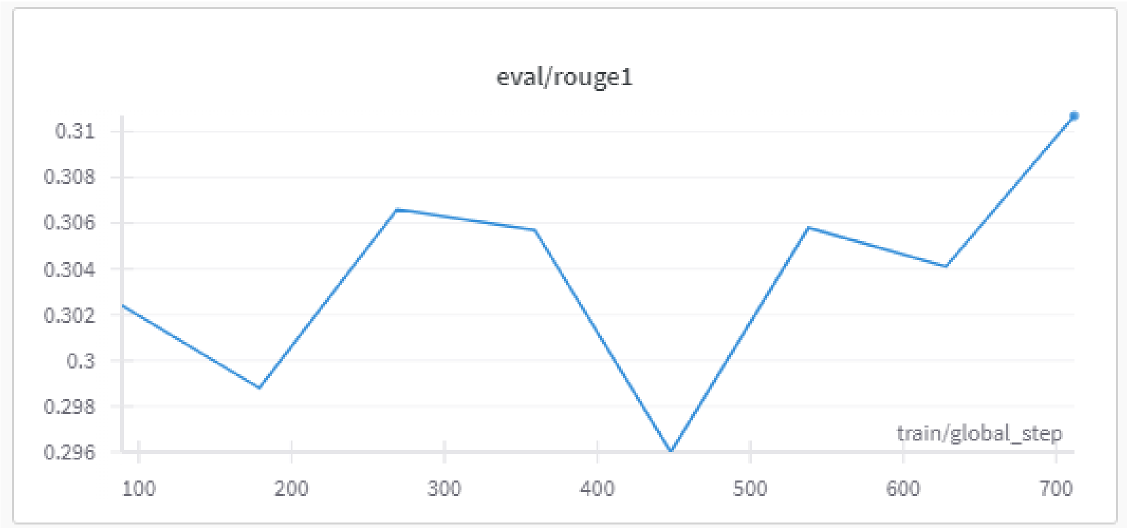


Figure 11 – Visualization of Epoch in ROUGE-1 for BART

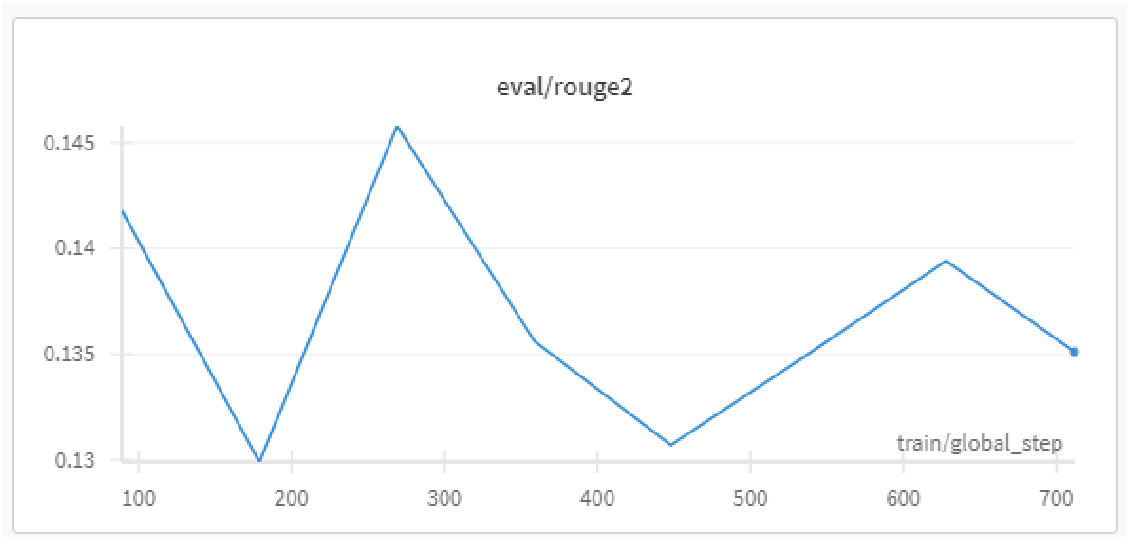


Figure 12 – Visualization of Epoch in ROUGE-2 for BART

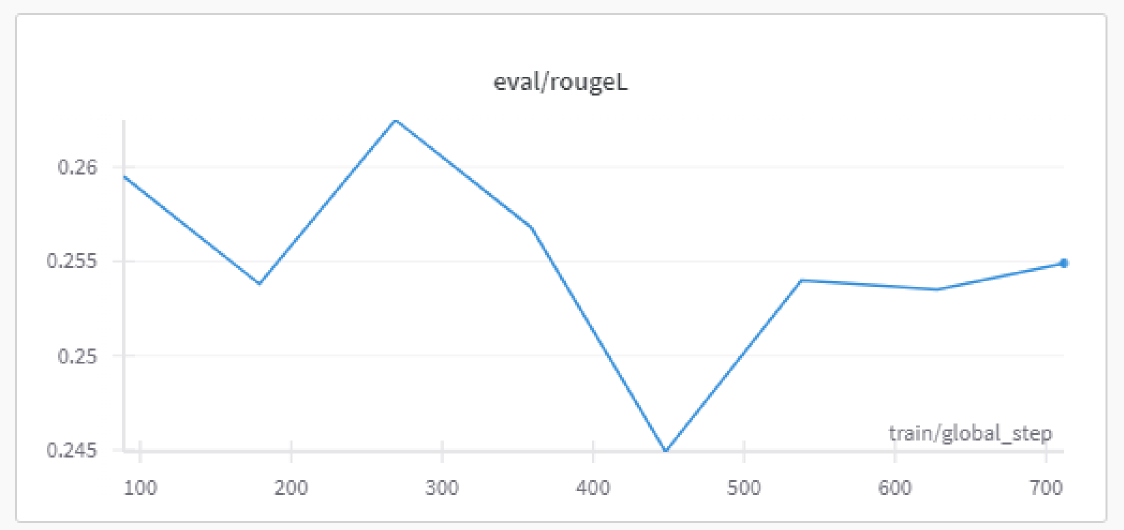


Figure 13 – Visualization of Epoch in ROUGE-L for BART

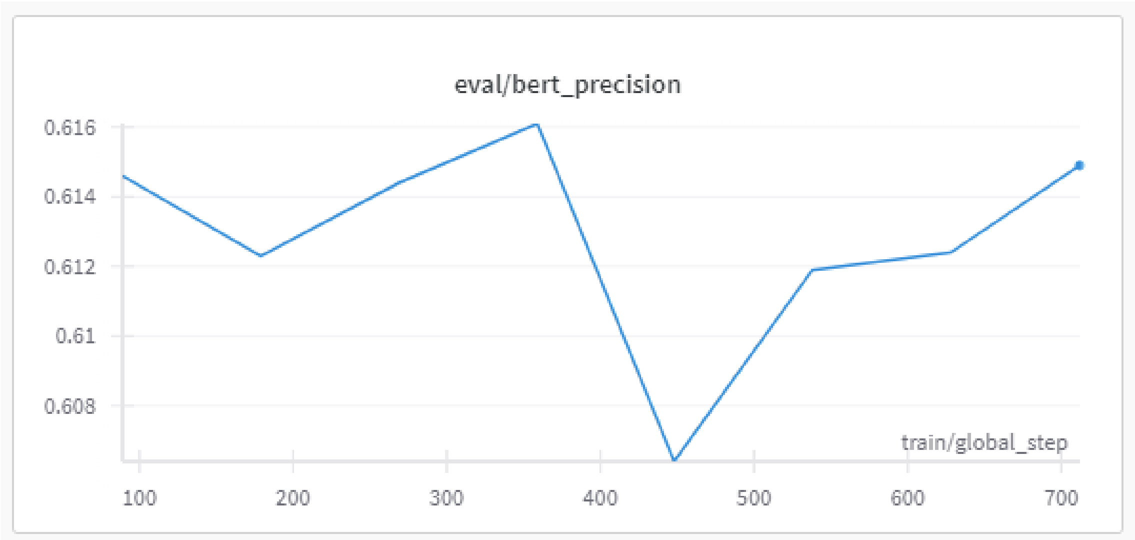


Figure 14 – Visualization of Epoch in Precision of BERT for BART

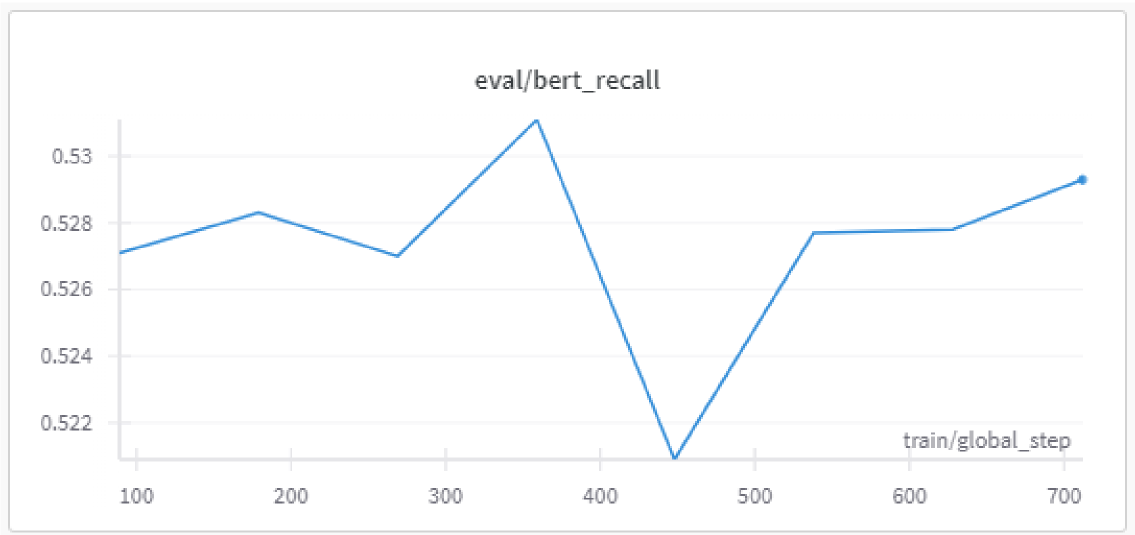


Figure 15 – Visualization of Epoch in Recall of BERT for BART

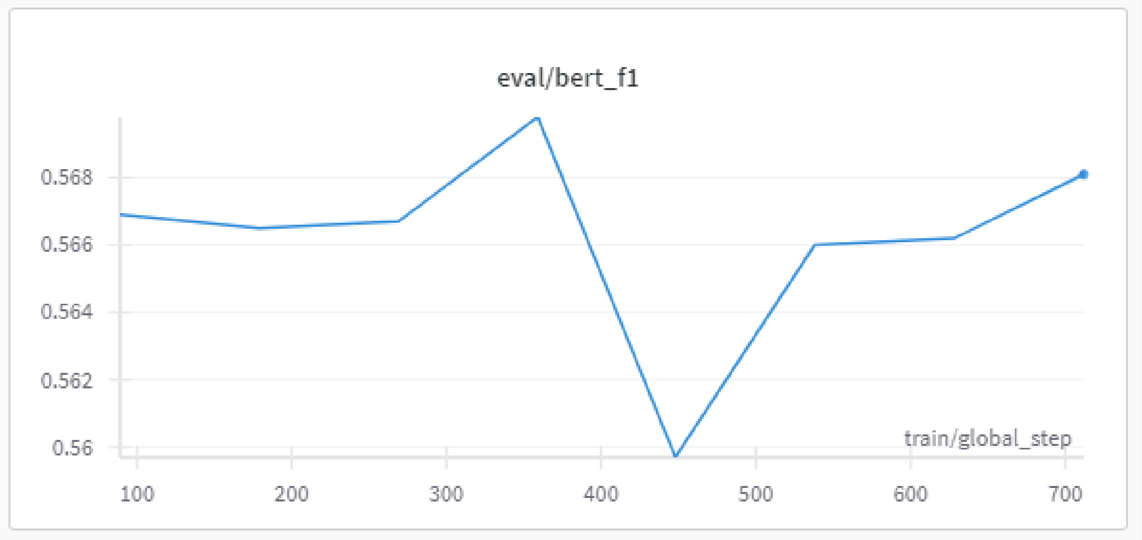


Figure 16 – Visualization of Epoch in F1-Score of BERT for BART

After gaining and saving the model that have been fine-tuned with BART using the dataset, it can now be use to generate the summaries, firstly by loading the model and also its tokenizer (Code Sample 14) and defining a function to generate summary (Code Sample 15), it will be ready to generate the summary for the dataset.

```
model_path = "bart-cnn/checkpoint-7000"
model = BartForConditionalGeneration.from_pretrained(model_path)
tokenizer = BartTokenizer.from_pretrained(model_path)
```

Code Sample 14 – Load Fine-tuned BART Model and Tokenizer

```
def generate_summary(article):
    # Tokenize the input article
    inputs = tokenizer.encode("summarize: " + article, return_tensors="pt", max_length=1024, truncation=True)

    # Generate summary output
    summary_ids = model.generate(inputs, max_length=150, min_length=40, length_penalty=2.0, num_beams=4, early_stopping=True)

    # Decode the generated summary
    summary = tokenizer.decode(summary_ids[0], skip_special_tokens=True)

    return summary
```

Code Sample 15 – Define Function to Generate Summary

Now, after the preparation to generate summary is done, a function to get the article from the dataset that will utilizes the function to generate summary will be made (Code Sample 16), and again the time to generate summaries will also be calculated (Code Sample 17).

```python
# Start measuring time
start_time = time.time()

# Iterate through each article in the test dataset and generate summaries
for article in dataset['test']['article']:
    summary = generate_summary(article)

# End measuring time
end_time = time.time()

# Calculate runtime
runtime = end_time - start_time

# Format runtime
formatted_runtime = "{:.4f}".format(runtime)
print("Total runtime:", formatted_runtime, "seconds")
```

Code Sample 16 – Generate Summaries for the Test Dataset and Count its Runtime

```
Total runtime: 1079.5847 seconds
```

Code Sample 17 – Total Runtime to Generate Summaries for Test Set using BART Fine-tuned Model

And lastly there will be some examples of the summary made by this model with its respective evaluation metrics scores (Table 2) and for the original article that have been used can be seen in the Appendix (Table A.1).

Table 2 – Summary and Metrics Scores for BART

Summary	ROUGE and BERT Scores
The attack in Kenya killed 142 students, three security officers and two university security personnel. Kenyans tweet pictures of the victims in happier times. The Kenyan Authorities have not released the Kenyan.	ROUGE-1: 0.56 ROUGE-2: 0.31 ROUGE-L: 0.32 BERT Precision: 0.60 BERT Recall: 0.27 BERT F1: 0.43
Five patients have been released from Omaha, Nebraska, hospital. One of the five had a heart-related issue on Saturday and has been discharged. The others have already gone home.	ROUGE-1: 0.49 ROUGE-2: 0.22 ROUGE-L: 0.25 BERT Precision: 0.59 BERT Recall: 0.28 BERT F1: 0.43
Yahya Rashid, 19, was detained at Luton airport on Tuesday. He's been charged with engaging in conduct in preparation of acts of terrorism.	ROUGE-1: 0.53 ROUGE-2: 0.31 ROUGE-L: 0.44 BERT Precision: 0.66 BERT Recall: 0.37 BERT F1: 0.51

End of Table 2

Summary	ROUGE and BERT Scores
Arsenal kept their slim hopes of winning this season's English League title alive by beating Burnley 1-0 at Turf Moor, with the help from Welsh international Aaron Ramsey.	ROUGE-1: 0.37 ROUGE-2: 0.16 ROUGE-L: 0.27 BERT Precision: 0.38 BERT Recall: 0.03 BERT F1: 0.17
Katie gave birth to a not-so-little baby on Friday at the Dallas Zoo. The giraffe is 6 feet tall, and has no immediate word on the gender or condition.	ROUGE-1: 0.47 ROUGE-2: 0.22 ROUGE-L: 0.24 BERT Precision: 0.53 BERT Recall: 0.16 BERT F1: 0.34

3.3 Pegasus

The next technique is Pegasus, in which most of the steps will be more or less the same as with BART, so some code will be redirected to the previous code result or sample. So, to start the fine-tuning of this pre-trained model, which is Pegasus, is to load the dataset. After loading the data, a chosen Pegasus model and its tokenizer will need to be defined for the fine-tuning process, where the chosen model and tokenizer can be seen below (Code Sample 18).

```
model_checkpoint = "google/pegasus-cnn_dailymail"
tokenizer = PegasusTokenizer.from_pretrained(model_checkpoint)
```

Code Sample 18 – Define Pegasus Model and Tokenizer

After defining the model and tokenizer that will be used, the next step is the same as BART, where defining a prefix to make Pegasus know this is a summarization task and making a function to preprocess the article will be defined like what has been shown before (Code Sample 7).

Then, continuing the preprocessing function (Code Sample 7), the task is to utilize that function to tokenize the dataset, and also to make it more effective, the padding will be done dynamically, and to do so can be seen the same as BART has shown (Code Sample 8).

After successfully preprocessing the dataset, the evaluation methods, ROUGE and BERT, will need to be loaded (Code Sample 9), and a function to calculate the ROUGE and BERT scores for Pegasus will also need to be defined, which will also use the same code as before (Code Sample 10).

Now, after all of those preparations, the next step will be loading the Pegasus model that has been defined before by utilizing a class that is suitable for loading the model, as can be shown below (Code Sample 19).

```
model = PegasusForConditionalGeneration.from_pretrained(model_checkpoint)
```

Code Sample 19 – Load Pegasus Model

After loading the model, the fine-tuning process of Pegasus with the dataset can now be concluded, and to do so, one must define the training hyperparameters and also the Trainer class (Code Sample 20), which will help to evaluate the ROUGE and BERT metrics.

```
args = Seq2SeqTrainingArguments(
    output_dir="pegasus-cnn",
    evaluation_strategy="epoch",
    learning_rate=1e-6,
    per_device_train_batch_size=4,
    per_device_eval_batch_size=4,
    weight_decay=0.01,
    save_total_limit=2,
    num_train_epochs=3,
    gradient_accumulation_steps=4,
    predict_with_generate=True,
    report_to="wandb",
    run_name = "pegasus-cnn-training",
)

trainer = Seq2SeqTrainer(
    model,
    args,
    train_dataset = tokenized_dataset["train"],
    eval_dataset = tokenized_dataset["test"],
    tokenizer = tokenizer,
    data_collator = data_collator,
    compute_metrics = compute_metrics
)
```

Code Sample 20 – Training Hyperparameters and Trainer Class for Pegasus

And after many trials and errors by assigning different values, as can be seen above (Code Sample 20), the final values have been assigned for the training hyperparameters that are best suited for Pegasus with the dataset and also with the personal device that has been used. Below are the personal reasons for choosing those values after all the experiments.

- Evaluation Strategy (*epoch*)

  Assigning *epoch* as the evaluation strategy is common in machine learning practices; with it, monitoring the training process of the model will be easier and can help determine the value of other parameters.

- Learning Rate (*1e-6*)

  Making the learning rate as *1e-6* is the most suitable value for this model. A smaller learning rate allows this model for finer adjustment of model parameters during training while also reducing the risk of overshooting that may lead to failure to convergence.

- Batch Size (*train* = *4* and *evaluation* = *4*)

  With *4* as the value for both the *train* and *evaluation* batch sizes, it is the most suitable value for this model. Many experiments with higher or lower batch sizes aren't really compatible with the fine-tuning process, but with *4* as the value, which may make the training process not too fast, it is also a good thing because less memory is used and the training process runs smoothly.

- Weight Decay (*0.01*)
  Assigning *0.01* as the value for weight decay suits the best for this model; it's the common and recommended value for weight decay, and with that value, it gives balance in the regularization technique that will help prevent overfitting during the training process.
- Save Total Limit (*2*)
  In the save total limit parameter, assigning *2* as the value is the most convenient way personally, because those values don't take up too much space, so the training process can run smoothly.
- Number of Epochs Trained (*3*)
  For the number of epochs, other values have been tried to be assigned, but when comparing it to this last value, assigning *3* as the value for how many epochs to be trained is the best for getting the ROUGE and BERT scores.
- Gradient Accumulation Steps (*4*)
  Assigning *4* as the value for this parameter suits the best; it also fits the batch size, which allows for a more stable training process, and this value also suits the memory that is being trained.
- Predict with Generate (*true*)
  For this parameter, assigning *true* is necessary, because this model will need to generate summary, and it fits with this value because it allows the model to generate summaries directly during the evaluation process.

With that, conclude the chosen parameters and their corresponding values to be trained or fine-tuned with Pegasus using the dataset. Where the fine-tuning process has given the results of the metrics that will be shown below (Code Sample 21), it can also be seen for the visualization of the process of training the epoch for ROUGE (Figure 17, Figure 18, and Figure 19) and BERT (Figure 20, Figure 21, and Figure 22).

Rouge1	Rouge2	Rougel	Bert Precision	Bert Recall	Bert F1
0.356300	0.161700	0.268900	0.539600	0.672100	0.597400
0.356800	0.161800	0.269400	0.540200	0.672200	0.597800
0.356800	0.161800	0.269400	0.540200	0.672200	0.597800

Code Sample 21 – Results of ROUGE and BERT Scores for Pegasus

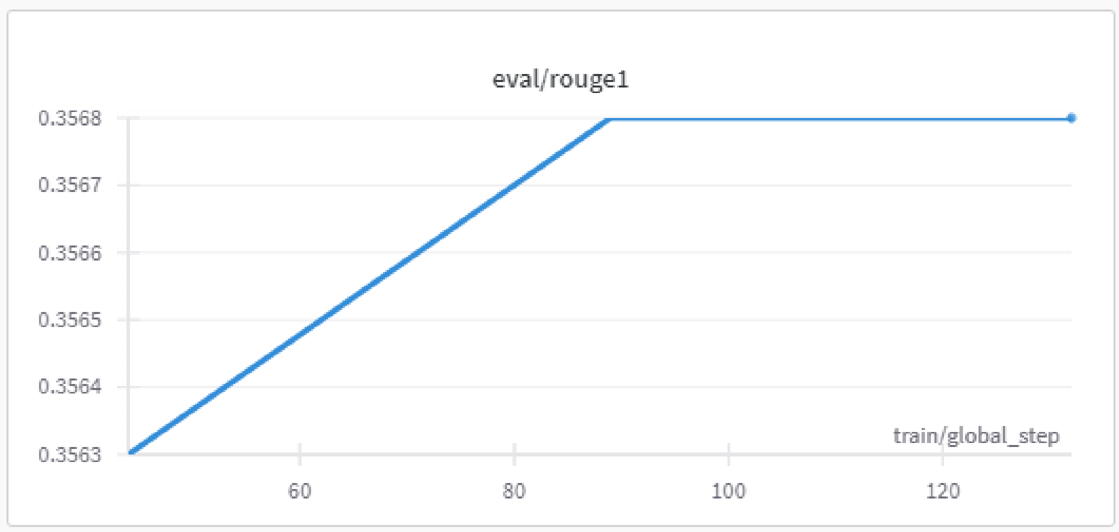


Figure 17 – Visualization of Epoch in ROUGE-1 for Pegasus

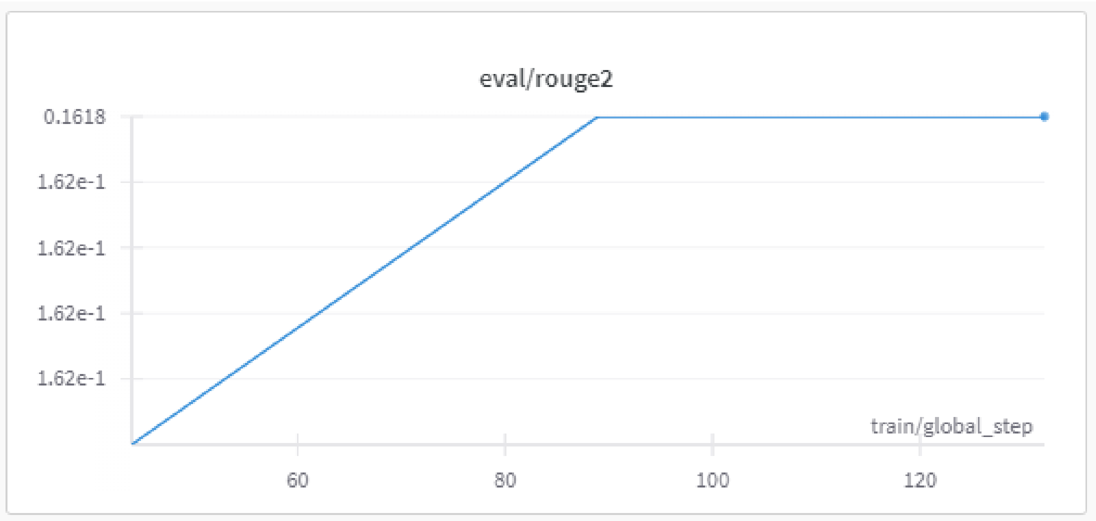


Figure 18 – Visualization of Epoch in ROUGE-2 for Pegasus

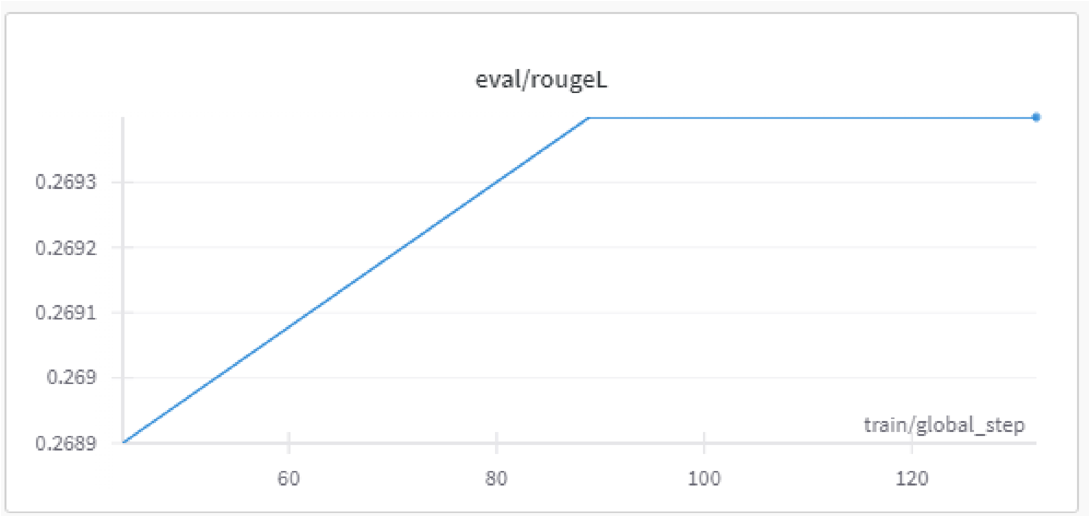


Figure 19 – Visualization of Epoch in ROUGE-L for Pegasus

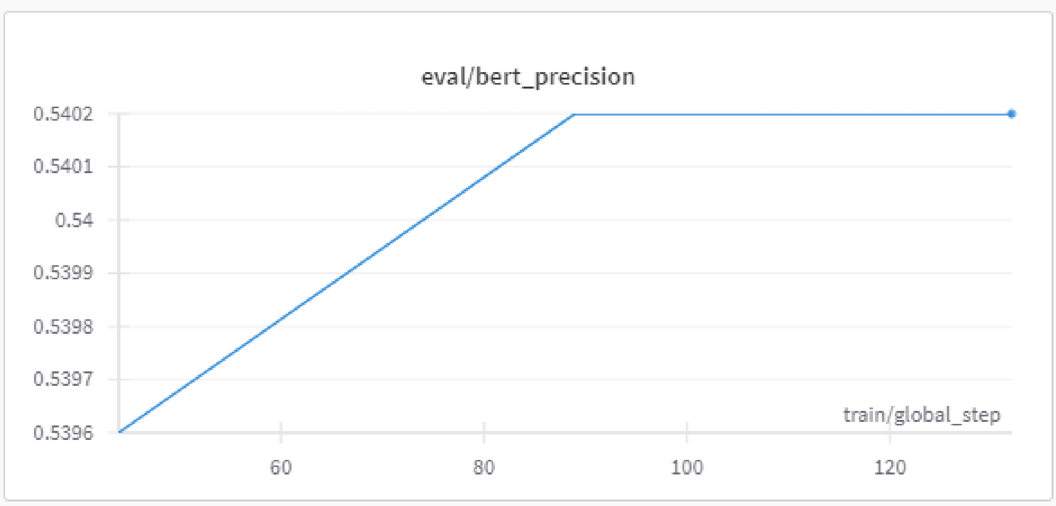


Figure 20 – Visualization of Epoch in Precision of BERT for Pegasus

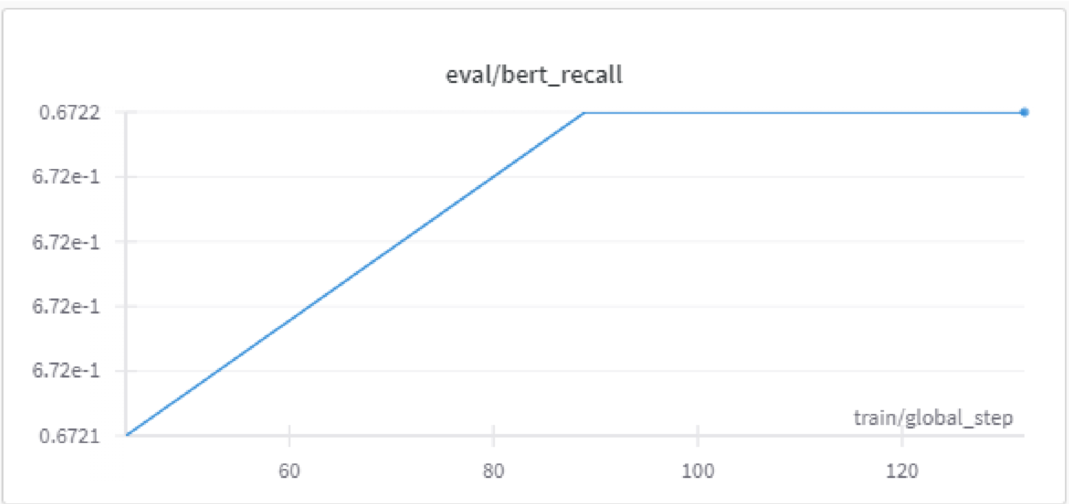


Figure 21 – Visualization of Epoch in Recall of BERT for Pegasus

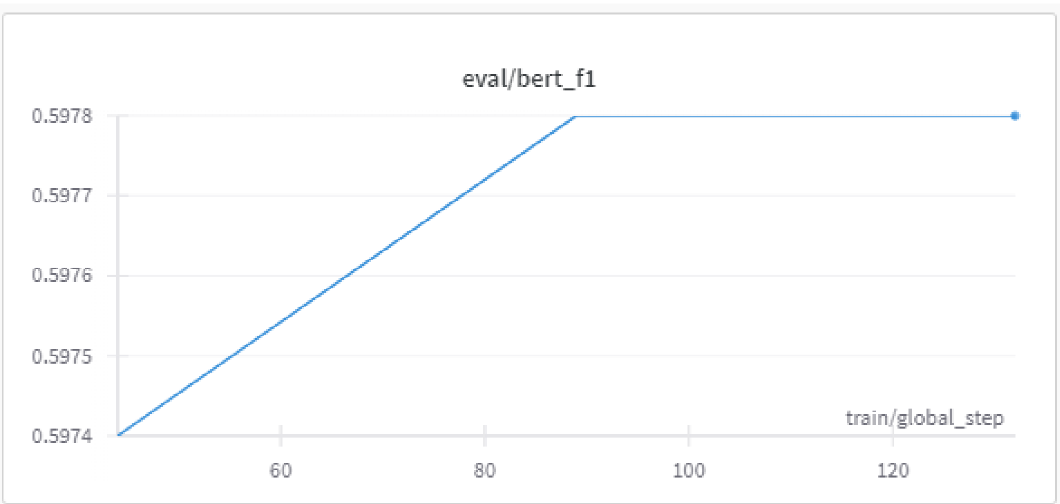


Figure 22 – Visualization of Epoch in F1-Score of BERT for Pegasus

After gaining the results from the modal that have been fine-tuned with BART using the dataset, it now can be saved and use to generate summaries. The first thing to do, is to load the best model that have been saved with the model loader and tokenizer that suits it like as shown below (Code Sample 22) and then make a function to generate summaries that will also utilize the what have been load before, and the function will look the same as what have been done in BART (Code Sample 15).

```
model_path = 'pegasus-cnn/checkpoint-1000'
model = PegasusForConditionalGeneration.from_pretrained(model_path)
tokenizer = PegasusTokenizer.from_pretrained(model_path)
```

Code Sample 22 – Load Fine-tuned Pegasus Model and Tokenizer

Continuing to the next step is to generate the summaries using the test set in the dataset, like what has been shown in BART (Code Sample 16), because it is used in the same way. Now, after it's done, the runtime for generating the summaries for the test set in the dataset can be shown below (Code Sample 23).

```
Total runtime: 1924.6472 seconds
```

Code Sample 23 – Total Runtime to Generate Summaries for Test Set using Pegasus Fine-tuned Model

Also, some summaries from the fine-tuned Pegasus model with its metrics scores can be shown (Table 3) for further comparison and for the original article that have been used can be seen in the Appendix (Table A.1).

Table 3 – Summary and Metrics Scores for Pegasus

Summary	ROUGE and BERT Scores
Kenyans use social media to share the victims' stories, hopes and dreams. Kenyan authorities have not released a list of the victims. The attack killed 142 students, three security officers and two university security personnel.	ROUGE-1: 0.46 ROUGE-2: 0.46 ROUGE-L: 0.31 BERT Precision: 0.74 BERT Recall: 0.20 BERT F1: 0.46
One of the five had a heart-related issue on Saturday and has been discharged. They were exposed to Ebola in Sierra Leone in March, but none developed the deadly virus. They all had contact with a colleague who was diagnosed with the disease.	ROUGE-1: 0.57 ROUGE-2: 0.55 ROUGE-L: 0.57 BERT Precision: 0.71 BERT Recall: 0.38 BERT F1: 0.54
Yahya Rashid is charged with engaging in conduct in preparation of acts of terrorism. He's also charged with engaging in conduct with the intention of assisting others to commit acts of terrorism. Rashid is due to appear in court on Wednesday.	ROUGE-1: 0.82 ROUGE-2: 0.79 ROUGE-L: 0.35 BERT Precision: 0.72 BERT Recall: 0.56 BERT F1: 0.64

End of Table 3

Summary	ROUGE and BERT Scores
Arsenal beat Burnley 1-0 to move to within four points of first placed Chelsea. Chelsea have two games in hand and play QPR on Sunday. Arsenal have won eight games in a row since the start of the year.	ROUGE-1: 0.38 ROUGE-2: 0.36 ROUGE-L: 0.37 BERT Precision: 0.61 BERT Recall: 0.02 BERT F1: 0.30
Katie, a giraffe at the Dallas Zoo, gives birth to a baby giraffe live on the Internet. There's no word on the baby's gender or condition. The giraffe's 15-month gestation period is an average for a giraffe.	ROUGE-1: 0.48 ROUGE-2: 0.47 ROUGE-L: 0.48 BERT Precision: 0.57 BERT Recall: 0.15 BERT F1: 0.35

3.4 Claude LLM

Lastly, the last text summarization technique that will be explored is Claude LLM, which can be said to be a newcomer but promising, and here the API from Claude LLM will be utilized to get summaries and also the ROUGE and BERT scores. Firstly, like all of the techniques before, loading the dataset will be the first step. Then, the next step will be defining the Claude API key so the script can interact with the Claude API. Also, the evaluation metrics will be loaded for further use, as shown below (Code Sample 24).

```
# Define Claude API key
API_KEY = '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'

# Initialize Anthropic client
client = Anthropic(api_key=API_KEY)

# Load BERT tokenizer
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

# Define BERT model
bert_model = "bert-base-uncased"

# Define evaluation method (ROUGE)
rouge = evaluate.load("rouge")

# Define BERT scorer
scorer = SentenceTransformer(model_name_or_path=bert_model)
```

Code Sample 24 – Initialize Claude API Key and Load Evaluation Metrics for Claude LLM

After that, some functions will need to be defined to be used in generating the summaries. Starting with the function below (Code Sample 25), which will interact with the Claude API, it will get the model of Claude and encapsulate the process of sending a user message to the Claude API, which generates a completion (referred to as an "article" in the context of the function). The completion contains a summary of the input text. The function then extracts the generated completion (generated summary) text from the response and returns it for further use.

```python
def get_completion(client, text):
    MODEL_NAME = "claude-3-opus-20240229"
    return client.messages.create(
        model=MODEL_NAME,
        max_tokens=2048,
        messages=[{"role": 'user', "content": text}]
    ).content[0].text
```

Code Sample 25 – Define Function to Interact with Claude API to Obtain a Completion (Summary)

Next, a function to calculate the ROUGE and BERT scores for the generated summaries by the Claude API will be made, as can be seen below (Code Sample 26), to then be used as evaluation metrics and in the metrics comparison with other text summarization models or techniques.

```python
def compute_metrics(completion_text, reference_highlight):
    # Calculate ROUGE scores
    rouge_results = rouge.compute(predictions=[completion_text], references=[reference_highlight], use_stemmer=True)

    # Calculate BERT scores
    bert_results = bert_score([completion_text], [reference_highlight], lang="en", model_type=bert_model)

    return rouge_results, bert_results
```

Code Sample 26 – Define Function to Calculate Evaluation Metrics for Claude LLM

Then, the last function that is needed to be made is a function to summarize the article and to evaluate the metrics by utilizing the function before (Code Sample 25 and Code Sample 26), where this function (Code Sample 27) will generate the completion text or generate the summary using the Claude API and also get the metrics that have been said before, as can be seen below.

```python
def summarize_and_evaluate(article):
    # Generate completion text using Claude API
    completion_text = get_completion(client, article['article'])

    # Get the reference highlight
    reference_highlight = article['highlights']

    # Calculate metrics
    rouge_scores, bert_scores = compute_metrics(completion_text, reference_highlight)

    return rouge_scores, bert_scores
```

Code Sample 27 – Define Function to Get the Summaries and Calculate the Evaluation Metrics for Claude LLM

Lastly, to fully utilize all of the functions that have been made, a variable to evaluate the metrics and get the summary will be made below (Code Sample 28), while also measuring the time it takes to generate the summaries in the test set of the dataset to then be compared with others, where the runtime, the metrics results, and also the visualization of the metrics results can also be seen below (Code Sample 29 and Figure 23).

```
# Start measuring time
start_time = time.time()

# Calculate evaluation metrics for the test dataset
evaluation_results = [summarize_and_evaluate(article) for article in test_news_dataset]

# End measuring time
end_time = time.time()

# Calculate runtime
runtime = end_time - start_time

# Format runtime
formatted_runtime = "{:.4f}".format(runtime)
print("Runtime:", formatted_runtime, "seconds")
```

Code Sample 28 – Define Function to Calculate the Evaluation Metrics for the Test Set in the Dataset for Claude LLM

```
Rouge1 Scores: 0.23911633401260848
Rouge2 Scores: 0.08771649441635669
RougeL Scores: 0.15320589575035298
BERT Precision Scores: 0.4940331
BERT Recall Scores: 0.65450925
BERT F1 Scores: 0.5617026
Runtime: 2163.9916 seconds
```

Code Sample 29 – Results of ROUGE and BERT Scores with its Runtime for Claude LLM

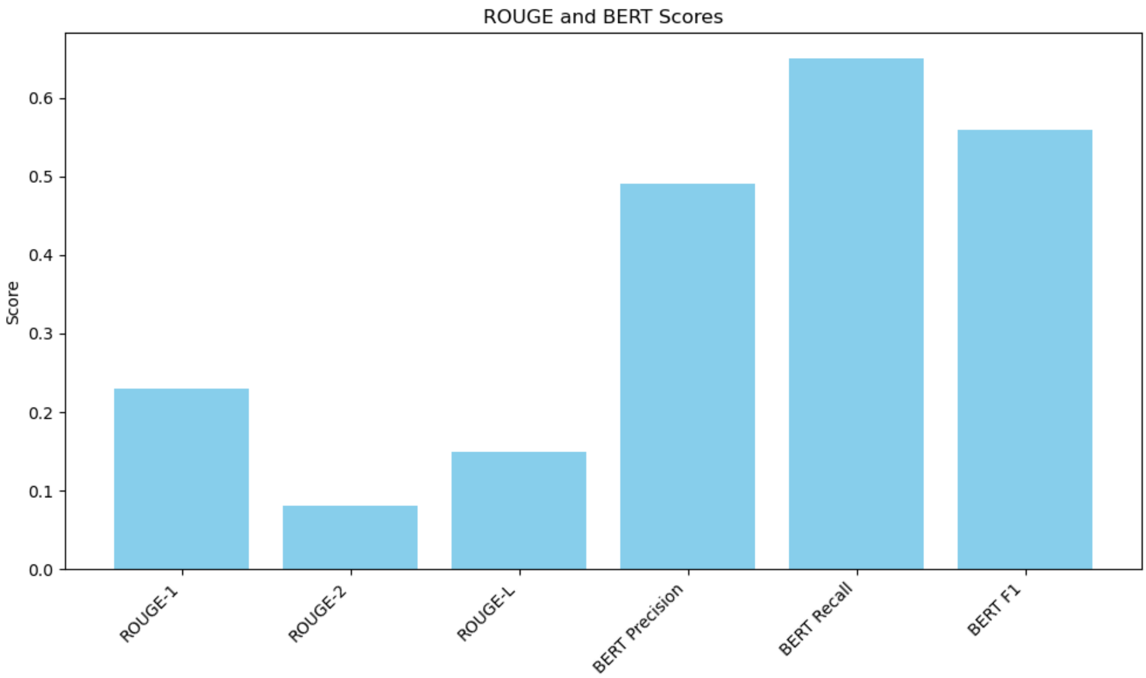


Figure 23 – Visualization of ROUGE and BERT Scores for Claude LLM

And with that, the results of the evaluation metrics for generated summaries by Claude LLM that utilize the Claude API can be shown. Some of the generated summaries can also be seen below (Table 4), which will be used for further comparison with other text summarization techniques or models and for the original article that have been used can be seen in the Appendix (Table A.1).

Table 4 – Summary and Metrics Scores for Claude LLM

Summary	ROUGE and BERT Scores
In a tragic terrorist attack at Garissa University College in Kenya, 147 people, mostly students, were killed by Al-Shabaab terrorists. Kenyans used the hashtag #147notjustanumber on social media to share stories and pictures of the victims, highlighting their hopes and dreams. The attack, which occurred on Thursday, was the deadliest in Kenya since the 1998 U.S. Embassy bombing. During the attack, the gunmen separated Muslims from Christians, killing the latter. Kenyan churches mourned the victims during emotional Easter services under the protection of armed guards. One of the attackers killed by security forces was identified as the son of a government official. Al-Shabaab, an Islamist extremist group based in Somalia, has carried out attacks in Kenya before, including the 2013 Westgate Mall attack in Nairobi that killed nearly 70 people.	ROUGE-1: 0.21 ROUGE-2: 0.20 ROUGE-L: 0.21 BERT Precision: 0.65 BERT Recall: 0.04 BERT F1: 0.29
Five American clinicians from Partners in Health were monitored for Ebola exposure at a Nebraska hospital for three weeks after contact with an infected colleague in Sierra Leone. None developed the virus and have been released, with one being treated for a heart issue before leaving the area. The CDC expects all 17 monitored patients to be released by Thursday. The Ebola epidemic in West Africa, primarily affecting Guinea, Liberia, and Sierra Leone, has claimed over 10,000 lives since December 2013.	ROUGE-1: 0.35 ROUGE-2: 0.34 ROUGE-L: 0.35 BERT Precision: 0.69 BERT Recall: 0.10 BERT F1: 0.38
A 19-year-old British man, Yahya Rashid, was arrested and charged with terror offenses upon his return from Turkey to the UK. The charges relate to his alleged involvement in preparing and assisting others to commit acts of terrorism between November 2014 and March 2015. Rashid is set to appear in court on Wednesday.	ROUGE-1: 0.51 ROUGE-2: 0.43 ROUGE-L: 0.47 BERT Precision: 0.69 BERT Recall: 0.18 BERT F1: 0.42

End of Table 4

Summary	ROUGE and BERT Scores
Arsenal kept their Premier League title hopes alive with a 1-0 win over Burnley, thanks to a goal from Aaron Ramsey. The victory took Arsenal to within four points of leaders Chelsea, who have two games in hand. Arsenal's good form is attributed to the introduction of Francis Coquelin in a defensive midfield role. The upcoming match between Arsenal and Chelsea is likely to decide the title race. In the relegation battle, Aston Villa continued their resurgence under Tim Sherwood with a 1-0 win over Tottenham, while bottom club Leicester City secured a dramatic 3-2 victory against West Brom. Sunderland suffered a heavy 4-1 defeat at home to Crystal Palace, with Yannick Bolasie scoring a hat trick for the visitors.	ROUGE-1: 0.15 ROUGE-2: 0.12 ROUGE-L: 0.14 BERT Precision: 0.55 BERT Recall: 0.24 BERT F1: 0.12
Katie, a giraffe at the Dallas Zoo, gave birth to a baby giraffe on Friday evening in a live-streamed event on Animal Planet. The hour-long labor was captured by 10 cameras, and the newborn, whose gender and condition were not immediately known, showed positive signs like moving ears, attempting to stand, and nursing from its mother. The baby, which is about 6 feet tall, joins a 4-year-old sister named Jamie. Katie, known as the "diva" among the zoo's herd of 12 giraffes, underwent a 15-month gestation period, which is average for giraffes. The birth was witnessed by other giraffes in an adjacent barn, including Katie's best friend, Jade.	ROUGE-1: 0.24 ROUGE-2: 0.15 ROUGE-L: 0.18 BERT Precision: 0.54 BERT Recall: 0.11 BERT F1: 0.20

3.5 Model Comparison

After exploring all of the selected techniques which are Luhn Summarization, BART, Pegasus, and Claude LLM, while also obtaining what is targeted in the requirements chapter, which are ROUGE scores, BERT scores, and the runtime for training if any and generating summaries, the comparison proses of the model can now be conducted, firstly let’s see on each of the summary generated by each of the text summarization technique or model that have also been provided before below, started with the first article (Table 5), second article (Table 6), third article (Table 7), fourth article (Table 8), and lastly the fifth article (Table 9).

Table 5 – Summary Results for the First Article

Model	Summary
Luhn Summarization	The attack in Kenya killed 142 students, three security officers and two university security personnel, and was the nation's deadliest since the bombing of the U.S. Embassy in 1998. As Kenyans mourned those killed last week in one of the deadliest terrorist attacks in the nation, citizens used social media to share the victims' stories, hopes and dreams. Using the hashtag #147notjustanumber -- a reference to the number of people, mostly students, killed at Garissa University College on Thursday -- Kenyans tweeted pictures of the victims in happier times.
BART	The attack in Kenya killed 142 students, three security officers and two university security personnel. Kenyans tweet pictures of the victims in happier times. The Kenyan Authorities have not released the Kenyan.
Pegasus	Kenyans use social media to share the victims' stories, hopes and dreams. Kenyan authorities have not released a list of the victims. The attack killed 142 students, three security officers and two university security personnel.
Claude LLM	In a tragic terrorist attack at Garissa University College in Kenya, 147 people, mostly students, were killed by Al-Shabaab terrorists. Kenyans used the hashtag #147notjustanumber on social media to share stories and pictures of the victims, highlighting their hopes and dreams. The attack, which occurred on Thursday, was the deadliest in Kenya since the 1998 U.S. Embassy bombing. During the attack, the gunmen separated Muslims from Christians, killing the latter. Kenyan churches mourned the victims during emotional Easter services under the protection of armed guards. One of the attackers killed by security forces was identified as the son of a government official. Al-Shabaab, an Islamist extremist group based in Somalia, has carried out attacks in Kenya before, including the 2013 Westgate Mall attack in Nairobi that killed nearly 70 people.

Table 6 – Summary Results for the Second Article

Model	Summary
Luhn Summarization	Five Americans who were monitored for three weeks at an Omaha, Nebraska, hospital after being exposed to Ebola in West Africa have been released, a Nebraska Medicine spokesman said in an email Wednesday. They all had contact with a colleague who was diagnosed with the disease and is being treated at the National Institutes of Health in Bethesda, Maryland. More than 10,000 people have died in a West African epidemic of Ebola that dates to December 2013, according to the World Health Organization.
BART	Five patients have been released from Omaha, Nebraska, hospital. One of the five had a heart-related issue on Saturday and has been discharged. The others have already gone home.
Pegasus	One of the five had a heart-related issue on Saturday and has been discharged. They were exposed to Ebola in Sierra Leone in March, but none developed the deadly virus. They all had contact with a colleague who was diagnosed with the disease.
Claude LLM	Five American clinicians from Partners in Health were monitored for Ebola exposure at a Nebraska hospital for three weeks after contact with an infected colleague in Sierra Leone. None developed the virus and have been released, with one being treated for a heart issue before leaving the area. The CDC expects all 17 monitored patients to be released by Thursday. The Ebola epidemic in West Africa, primarily affecting Guinea, Liberia, and Sierra Leone, has claimed over 10,000 lives since December 2013.

Table 7 – Summary Results for the Third Article

Model	Summary
Luhn Summarization	He's been charged with engaging in conduct in preparation of acts of terrorism, and with engaging in conduct with the intention of assisting others to commit acts of terrorism. Yahya Rashid, a UK national from northwest London, was detained at Luton airport on Tuesday after he arrived on a flight from Istanbul, police said. A 19-year-old man was charged Wednesday with terror offenses after he was arrested as he returned to Britain from Turkey, London's Metropolitan Police said.
BART	Yahya Rashid, 19, was detained at Luton airport on Tuesday. He's been charged with engaging in conduct in preparation of acts of terrorism.
Pegasus	Yahya Rashid is charged with engaging in conduct in preparation of acts of terrorism. He's also charged with engaging in conduct with the intention of assisting others to commit acts of terrorism. Rashid is due to appear in court on Wednesday.
Claude LLM	A 19-year-old British man, Yahya Rashid, was arrested and charged with terror offenses upon his return from Turkey to the UK. The charges relate to his alleged involvement in preparing and assisting others to commit acts of terrorism between November 2014 and March 2015. Rashid is set to appear in court on Wednesday.

Table 8 – Summary Results for the Fourth Article

Model	Summary
Luhn Summarization	Arsenal have been in superb form since the start of the year, transforming what looked to be another mediocre season struggling to secure fourth place -- and with it Champions League qualification -- into one where they at least have a shot at winning the title. After going ahead, Arsenal rarely looked in any danger of conceding, showing more of the midfield pragmatism epitomized by the likes of Francis Coquelin, who also played a crucial role in the goal. Belgian international Christian Benteke scored the only goal of the game, his eighth in six matches, to secure a vital three points to give the Midlands club breathing space.
BART	Arsenal kept their slim hopes of winning this season's English League title alive by beating Burnley 1-0 at Turf Moor, with the help from Welsh international Aaron Ramsey.
Pegasus	Arsenal beat Burnley 1-0 to move to within four points of first placed Chelsea. Chelsea have two games in hand and play QPR on Sunday. Arsenal have won eight games in a row since the start of the year.
Claude LLM	Arsenal kept their Premier League title hopes alive with a 1-0 win over Burnley, thanks to a goal from Aaron Ramsey. The victory took Arsenal to within four points of leaders Chelsea, who have two games in hand. Arsenal's good form is attributed to the introduction of Francis Coquelin in a defensive midfield role. The upcoming match between Arsenal and Chelsea is likely to decide the title race. In the relegation battle, Aston Villa continued their resurgence under Tim Sherwood with a 1-0 win over Tottenham, while bottom club Leicester City secured a dramatic 3-2 victory against West Brom. Sunderland suffered a heavy 4-1 defeat at home to Crystal Palace, with Yannick Bolasie scoring a hat trick for the visitors.

Table 9 – Summary Results for the Fifth Article

Model	Summary
Luhn Summarization	But there were good signs, as seen on the live stream and Dallas Zoo's Twitter feed -- like its ears moving, its efforts to stand, and its nursing (or at least trying to nurse) from mom. But the giraffe definitely did have watchers in the form of fellow giraffes who saw the scene unfold from an abutting barn, one of them being Katie's BFF Jade. The zoo describes her as the "diva" among a herd of 12 giraffes at the zoo who loves to "toss her head around" when she doesn't like something.
BART	Katie gave birth to a not-so-little baby on Friday at the Dallas Zoo. The giraffe is 6 feet tall, and has no immediate word on the gender or condition.
Pegasus	Katie, a giraffe at the Dallas Zoo, gives birth to a baby giraffe live on the Internet. There's no word on the baby's gender or condition. The giraffe's 15-month gestation period is an average for a giraffe.
Claude LLM	Katie, a giraffe at the Dallas Zoo, gave birth to a baby giraffe on Friday evening in a live-streamed event on Animal Planet. The hour-long labor was captured by 10 cameras, and the newborn, whose gender and condition were not immediately known, showed positive signs like moving ears, attempting to stand, and nursing from its mother. The baby, which is about 6 feet tall, joins a 4-year-old sister named Jamie. Katie, known as the "diva" among the zoo's herd of 12 giraffes, underwent a 15-month gestation period, which is average for giraffes. The birth was witnessed by other giraffes in an adjacent barn, including Katie's best friend, Jade.

Those are the summaries generated by each of the text summarization techniques or models; some of them are shorter and longer than each other; some of them only copy and paste from the article; there was also a summary that paraphrased it; there are many differences between each of them; it’s understandable because each of the text summarization models has its own way to summarize it. Now let’s see below (Table 10) the comparison of their respective evaluation metrics and also the runtime to make it easier to understand those summaries.

Table 10 – Metrics Comparison for All Text Summarization Techniques or Models

Metrics Results	**Luhn Summarization**	**BART**	**Pegasus**	**Claude LLM**
ROUGE-1	0.19	0.31	**0.35**	0.23
ROUGE-2	0.09	0.13	**0.16**	0.08
ROUGE-L	0.13	0.25	**0.26**	0.15
BERT Precision	0.46	**0.61**	0.54	0.49
BERT Recall	0.59	0.52	**0.67**	0.65
BERT F1	0.52	0.56	**0.59**	0.56
Runtime for Model Training	-	**7279 seconds ≈ 2 hours**	17702 seconds ≈ 5 hours	-
Runtime to Generate Summaries for Test Set	1900 seconds ≈ 31 minutes	**1079 seconds ≈ 17 minutes**	1924 seconds ≈ 32 minutes	2163 seconds ≈ 36 minutes

Finally, all the metrics that are targeted are obtained, from ROUGE, BERT, to the runtime for model training if it has and for generating the summaries for the test set, some notes for the table above (Table 10), for Luhn Summarization and Claude LLM, the runtime for model training is set to empty because there is no model training in there, and here is the summary or important note that can be obtained from the table (Table 10) that is the results of the research or exploration for each of the text summarization technique or model.

- Luhn Summarization

  The oldest technique in here and can be said to be one of the simplest in terms of the algorithm, which is also not being trained. It is normal that its ROUGE and BERT scores show as the lowest from the others; the only higher metric is BERT Recall; even then, it is not the highest and only slightly higher than BART.

- BART

  This pre-trained model, which is fine-tuned using the dataset, gets pretty good metrics scores compared to the others with its good architecture, which uses a bidirectional encoder and an autoregressive decoder. It gets the highest score for BERT Precision, which means the generated summaries are pretty accurate in the proportion of relevant information from the reference summaries. Its runtime for model training is also 3 hours faster than Pegasus, and the runtime for generating summaries in the test set of the dataset is the fastest at only 17 minutes.

- Pegasus

  Next, the other pre-trained model that is fine-tuned is Pegasus; this model could be the winner or the best for generating summaries for news articles. The metrics results of this model prove it. Its rough scores are the highest compared to the others, which means this model's summaries are more similar to the reference summaries in terms of unigram (ROUGE-1), bigram (ROUGE-2), and longest common subsequence overlap (ROUGE-L). It can also be seen that the BERT Recall and BERT F1 scores are the highest, meaning the model's generated summaries are comprehensive, capturing a substantial amount of relevant information while maintaining a large portion of the relevant information from the reference summaries that are included in the generated summaries. But this pre-trained model had some drawbacks, namely that the runtime for model training took longer, up to 5 hours, and the runtime to generate summaries for the test set of the dataset took up to 32 minutes.

- Claude LLM

  Lastly, for Claude LLM, exploring a language model that utilizes an API and did not do any training, unlike the two pre-trained models before, gives a new insight. It can give moderate ROUGE scores to be fair; it's not as good as BART and Pegasus, but it's slightly better than Luhn. The BERT scores for Claude LLM are also pretty good; they didn’t drop too far compared to the others. But as for the runtime to generate the summaries, it can be seen as the slowest, and it is possible because of the process of the interaction with its API.

Therefore, this concludes the exploration for all the chosen text summarization models, and to finish it, the conclusion of the exploration has been made. Of the experiments that have been carried out in the making of these text summarization models or techniques, Pegasus comes out as the most suitable model for news articles. And based on the implementation that has been conducted, Pegasus may win over the others because of the technique used by Pegasus, which is the masking technique. That technique is possibly the most suitable for making a good summarization compared to the Luhn technique by searching for the highest frequency word, the BART guessing game, and the Claude API. And when looking into the training hyperparameters that can be compared with BART, Pegasus uses a smaller learning rate that can help the model converge to a better minimum during training, potentially leading to improved generalization and less overfitting. With the batch size, Pegasus uses a smaller batch size, which can allow for more frequent updates during training, potentially leading to better adaptation to the training data. And lastly, the training epoch: while BART is trained for more epochs, it's possible that Pegasus found a good solution within 3 epochs. Which means overtraining on the training data can lead to worse performance on unseen data. As can be seen in the above table (Table 10), with the highest metrics score in almost all aspects, it shows that it's the best. It may have shortcomings, such as a longer time it took to train or fine-tune the model and also in the generation of the summaries, but here the main metrics to look at are ROUGE and BERT, which evaluate the summaries that have been generated, where Pegasus gets the highest score from almost all of the ROUGE and BERT metrics.

CONCLUSION

The objective of this bachelor thesis paper is to explore the capabilities of Luhn Summarization, BART, Pegasus, and Claude LLM using open-source news articles. The specific objectives are to provide a text summarization model or technique, to then obtain the evaluation metrics scores of each of the models or techniques, and to conclude by comparing and concluding the result of each of the evaluation metrics of the text summarization model or technique.

The techniques, technologies, and tools mentioned in this paper can be leveraged to great effect when exploring text summarization models and techniques for generating summaries of news articles.

This paper successfully explores the capabilities of Luhn Summarization, BART, Pegasus, and Claude LLM in generating summaries for news articles. It concludes that Pegasus is the most suitable text summarization model or technique compared to the other three for generating summaries for news articles.

REFERENCES

1. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension / M. Lewis, Y. Liu, N. Goyal [et al.] // arXiv preprint arXiv:1910.13461v1. – 2019.
2. What is a large language model (LLM)?. Cloudflare. [N.p.], n.d. – URL: https://www.cloudflare.com/learning/ai/what-is-large-language-model/ (access date 05.04.2024).
3. ROUGE: A Package for Automatic Evaluation of Summaries / C. Lin // Association for Computational Linguistics. – P. 74-81. – 2004.
4. What is an API?. MuleSoft. M. Frye, n.d. – URL: https://www.mulesoft.com/resources/api/what-is-an-api (access date 05.04.2024).
5. Transfer learning and fine-tuning. TensorFlow. [N.p.], n.d. – URL: https://www.tensorflow.org/tutorials/images/transfer_learning (access date 05.03.2024).
6. What is Tokenization?. DataCamp. [N.p.], n.d. – URL: https://www.datacamp.com/blog/what-is-tokenization (access date 05.03.2024).
7. Prefix. Computer Hope. [N.p.], n.d. – URL: https://www.computerhope.com/jargon/p/prefix.htm#:~:text=A%20prefix%20is%20an%20affix,cores%20is%20denoted%20by%20prefixes (access date 05.03.2024)
8. Model inference overview. Google Cloud. [N.p.], n.d. – URL: https://cloud.google.com/bigquery/docs/inference-overview#:~:text=Machine%20learning%20inference%20is%20the,machine%20learning%20model%20into%20production.%22 (access date 05.03.2024).
9. fine-tuning. Tech Target. L. Craig, 2024. – URL: https://www.techtarget.com/searchenterpriseai/definition/fine-tuning#:~:text=Fine%2Dtuning%20is%20the%20process,suit%20more%20specialized%20use%20cases. (access date 05.03.2024).
10. Parameters and Hyperparameters in Machine Learning and Deep Learning. Towards Data Science. K. Nyuytiymbiy, 2020. – URL: https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac (access date 05.03.2024).
11. Trainer. Hugging Face. [N.p.], n.d. – URL: https://huggingface.co/docs/transformers/en/main_classes/trainer (access date 05.03.2024).
12. Project Jupyter. [N.p.], n.d. – URL: https://jupyter.org/ (access date 05.01.2024).
13. PyTorch Community. [N.p.], n.d. – URL: https://discuss.pytorch.org/ (access date 05.01.2024).
14. Hugging Face. [N.p.], n.d. – URL: https://huggingface.co/ (access date 05.04.2024).
15. What is Epoch in Machine Learning?. U-next. UNext Editorial Team, 2022. – URL: https://u-next.com/blogs/machine-learning/epoch-in-machine-learning/#:~:text=An%20epoch%20in%20machine%20learning,learning%20process%20of%20the%20algorithm. (access date 05.04.2024).
16. What Is Learning Rate in Machine Learning?. Pure Storage. [N.p.], n.d. – URL: https://www.purestorage.com/fr/knowledge/what-is-learning-rate.html (access date 05.04.2024).
17. Relation Between Learning Rate and Batch Size. Baeldung. E. Zvornicanin, 2024. – URL: https://www.baeldung.com/cs/learning-rate-batch-size#:~:text=Batch%20size%20defines%20the%20number,to%20train%20a%20neural%20network (access date 05.04.2024).
18. Weight Decay in Machine Learning: Concepts. Analytics Yogi. A. Kumar, 2022. – URL: https://vitalflux.com/weight-decay-in-machine-learning-concepts/ (access date 05.04.2024).

19. Gradient Accumulation. Hopsworks AI. [N.p.], n.d. – URL: https://www.hopsworks.ai/dictionary/gradient-accumulation#:~:text=Gradient%20Accumulation%20is%20a%20technique,throughput%20(reduce%20training%20time (access date 05.04.2024).
20. Dataset Card for CNN Dailymail Dataset. Hugging Face. A. See, 2022. – URL: https://huggingface.co/datasets/abisee/cnn_dailymail (access date 05.04.2024).
21. Jupyter Notebooks—a publishing format for reproducible computational workflows / T. Kluyver, B. Ragan-Kelley, F. Pérez [et al.] // In Positioning and Power in Academic Publishing: Players, Agents and Agendas. – 2016.
22. Python for Data Analysis: Second Edition / P. McKinney // O'Reilly Media, Inc. – 2017.
23. A Primer on Scientific Programming with Python / K. Langtangen // Springer International Publishing. – 2016.
24. Deep Learning with Python / F. Chollet // Manning Publications Co. – 2017.
25. Data Visualization with Python and Matplotlib / P. McKinney // O'Reilly Media, Inc. – 2017.
26. Automatic differentiation in PyTorch / A. Paszke, S. Gross, S. Chintala [et al.] // NIPS Autodiff Workshop. – 2017.
27. PyTorch: A Primer / J. Goldie // Packt Publishing Ltd. – 2017.
28. HuggingFace's Transformers: State-of-the-art Natural Language Processing / T. Wolf, L. Debut, V. Sanh [et al.] // arXiv preprint arXiv:1910.03771v5. – 2020.
29. PyTorch: An Imperative Style, High-Performance Deep Learning Library / A. Paszke, S. Gross, F. Massa [et al.] // arXiv preprint arXiv:1912.01703v1. – 2019.
30. The Automatic Creation of Literature Abstracts / H. P. Luhn // IBM Journal of Research and Development. – Vol. 2, is. 2. – P. 159-165. – 1958.
31. Term-weighting approaches in automatic text retrieval / G. Salton, C. Buckley // Information Processing & Management. – Vol. 24, is. 5. – P. 513-523. – 1988.
32. Automatic Summarization / I. Mani // John Benjamins Publishing Company. – 2001.
33. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin, M. Chang, K. Lee, K. Toutanova // arXiv preprint arXiv:1810.04805. – 2018.
34. Improving Language Understanding by Generative Pre-Training / A. Radford, K. Narasimhan, T. Salimans, I. Sutskever // Preprint 2018. – 2018.
35. SpanBERT: Improving Pre-training by Representing and Predicting Spans / M. Joshi, D. Chen, Y. Liu, D. S. Weld // arXiv preprint arXiv:1907.10529. – 2019.
36. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization / J. Zhang, Y. Zhao, M. Saleh, P. J. Liu // arXiv preprint arXiv:1912.08777. – 2019.
37. Attention Is All You Need / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit // arXiv preprint arXiv:1706.03762. – 2017.
38. Introducing Claude. Anthropic. [N.p.], n.d. – URL: https://www.anthropic.com/news/introducing-claude (access date 05.01.2024).
39. BERTScore: Evaluating Text Generation with BERT / T. Zhang, V. Kishore, F. Wu [et al.] // arXiv preprint arXiv:1904.09675. – 2019.

APPENDIX A

Table A.1 – Original Article from CNN Dailymail Dataset [20]

No.	Article
1.	One hundred and forty-seven victims. Many more families affected. Even more broken hopes and dreams. As Kenyans mourned those killed last week in one of the deadliest terrorist attacks in the nation, citizens used social media to share the victims' stories, hopes and dreams. Using the hashtag #147notjustanumber -- a reference to the number of people, mostly students, killed at Garissa University College on Thursday -- Kenyans tweeted pictures of the victims in happier times. Kenyan authorities have not released a list of the victims. The posts provided heart-wrenching details on the victims, including one about an elderly man whose dreams died with his son. He had reportedly taken a loan to educate him at the university, where he was killed by Al-Shabaab terrorists. The attack in Kenya killed 142 students, three security officers and two university security personnel, and was the nation's deadliest since the bombing of the U.S. Embassy in 1998. Kenyan churches mourned the dead during Easter services Sunday as armed guards protected the congregations. In emotional services nationwide, churchgoers wept as they paid tribute to the victims of the massacre. The gunmen who attacked the university in the predawn hours separated Muslims from Christians and killed the latter. The extremist group has also killed Muslims in recent attacks. The Interior Ministry has identified one of the attackers killed by security forces as the son of a government official. The father of suspect Abdirahim Abdullahi is a chief in Mandera and had reported his son missing, officials said. The Islamist extremist group is based in Somalia, but it hasn't confined its terrorism to the nation that shares a border with Kenya. In 2013, militants attacked Nairobi's upscale Westgate Mall, killing nearly 70 people. [20]
2.	Five Americans who were monitored for three weeks at an Omaha, Nebraska, hospital after being exposed to Ebola in West Africa have been released, a Nebraska Medicine spokesman said in an email Wednesday. One of the five had a heart-related issue on Saturday and has been discharged but hasn't left the area, Taylor Wilson wrote. The others have already gone home. They were exposed to Ebola in Sierra Leone in March, but none developed the deadly virus. They are clinicians for Partners in Health, a Boston-based aid group. They all had contact with a colleague who was diagnosed with the disease and is being treated at the National Institutes of Health in Bethesda, Maryland. As of Monday, that health care worker is in fair condition. The Centers for Disease Control and Prevention in Atlanta has said the last of 17 patients who were being monitored are expected to be released by Thursday. More than 10,000 people have died in a West African epidemic of Ebola that dates to December 2013, according to the World Health Organization. Almost all the deaths have been in Guinea, Liberia and Sierra Leone. Ebola is spread by direct contact with the bodily fluids of an infected person. [20]

No.	Article
3.	A 19-year-old man was charged Wednesday with terror offenses after he was arrested as he returned to Britain from Turkey, London's Metropolitan Police said. Yahya Rashid, a UK national from northwest London, was detained at Luton airport on Tuesday after he arrived on a flight from Istanbul, police said. He's been charged with engaging in conduct in preparation of acts of terrorism, and with engaging in conduct with the intention of assisting others to commit acts of terrorism. Both charges relate to the period between November 1 and March 31. Rashid is due to appear in Westminster Magistrates' Court on Wednesday, police said. CNN's Lindsay Isaac contributed to this report. [20]
4.	Arsenal kept their slim hopes of winning this season's English Premier League title alive by beating relegation threatened Burnley 1-0 at Turf Moor. A first half goal from Welsh international Aaron Ramsey was enough to separate the two sides and secure Arsenal's hold on second place. More importantly it took the north London club to within four points of first placed Chelsea, with the two clubs to play next week. But Chelsea have two games in hand and play lowly Queens Park Rangers on Sunday, a team who are themselves struggling against relegation. Good form . Arsenal have been in superb form since the start of the year, transforming what looked to be another mediocre season struggling to secure fourth place -- and with it Champions League qualification -- into one where they at least have a shot at winning the title. After going ahead, Arsenal rarely looked in any danger of conceding, showing more of the midfield pragmatism epitomized by the likes of Francis Coquelin, who also played a crucial role in the goal. "He has been absolutely consistent in the quality of his defensive work," Arsenal coach Arsene Wenger told Sky Sports after the game when asked about Coquelin's contribution to Arsenal's current run. They have won eight games in a row since introducing the previously overlooked young Frenchman into a more defensive midfield position. "He was a player who was with us for seven years, from 17, he's now just 24," Wenger explained. "Sometimes you have to be patient. I am very happy for him because he has shown great mental strength." Now all eyes will be on next week's clash between Arsenal and Chelsea which will likely decide the title. "They have the games in hand," said Wenger, playing down his club's title aspirations. "But we'll keep going and that's why the win was so important for us today." Relegation dogfight . Meanwhile it was a good day for teams at the bottom of the league. Aston Villa continued their good form since appointing coach Tim Sherwood with a 1-0 victory over Tottenham, who fired Sherwood last season. Belgian international Christian Benteke scored the only goal of the game, his eighth in six matches, to secure a vital three points to give the Midlands club breathing space. Another Midlands club looking over their shoulder is West Brom, who conceded an injury time goal to lose 3-2 against bottom club Leicester City. But it was an awful day for Sunderland's former Dutch international coach Dick Advocaat, who saw his team lose 4-1 at home against form team Crystal Palace. Democratic Republic of Congo international Yannick Bolasie scored Crystal Palace's first ever hat trick in the Premier League to secure an easy victory. [20]

End of Table A.1

No.	Article
5.	Anyone who has given birth -- or been an observer of the event -- knows how arduous it can be. But to do it live on the Internet? With two hooves sticking out for several minutes in the midst of labor? Luckily, Katie -- a giraffe at the Dallas Zoo -- is a champ. In an hour-long labor captured by 10 cameras and streamed live by Animal Planet, Katie gave birth to a not-so-little baby (about 6 feet tall) early Friday evening. There was no immediate word on the newborn's gender or condition. But there were good signs, as seen on the live stream and Dallas Zoo's Twitter feed -- like its ears moving, its efforts to stand, and its nursing (or at least trying to nurse) from mom. "We're so proud," the zoo tweeted. The newcomer's debut was a long time coming, especially when you count for Katie's 15-month gestation period -- average for a giraffe, according to Animal Planet. The baby joins a sister, 4-year-old calf Jamie. It wasn't immediately known how many people online saw Katie go into labor and give birth. But the giraffe definitely did have watchers in the form of fellow giraffes who saw the scene unfold from an abutting barn, one of them being Katie's BFF Jade. The fact that the spunky Katie held up so well under the spotlight isn't a total shocker. The zoo describes her as the "diva" among a herd of 12 giraffes at the zoo who loves to "toss her head around" when she doesn't like something. As Animal Planet noted, "She's one of the only giraffes at the Dallas Zoo who can stick her long tongue out on cue." CNN's Justin Lear contributed to this report. [20]

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет предоставлен сервисом «Антиплагиат» - http://tsu.antiplagiat.ru

Отчет о проверке

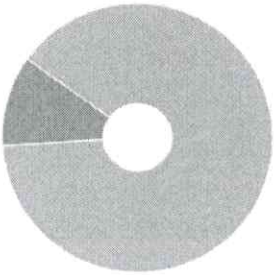
Автор: Arisudana Gede Yoga

Название документа: Arisudana Gede Yoga FW

Проверяющий: Политова Анастасия Михайловна

Организация: Томский Государственный Университет

РЕЗУЛЬТАТЫ ПРОВЕРКИ



Совпадения:
11,33%

Оригинальность:
88,67%

Цитирования:
0%

Самоцитирования:
0%

«Совпадения», «Цитирования», «Самоцитирования», «Оригинальность» являются отдельными показателями, отображаются в процентах и в сумме дают 100%, что соответствует проверенному тексту документа.

- **Совпадения** — фрагменты проверяемого текста, полностью или частично сходные с найденными источниками, за исключением фрагментов, которые система отнесла к цитированию или самоцитированию. Показатель «Совпадения» – это доля фрагментов проверяемого текста, отнесенных к совпадениям, в общем объеме текста.
- **Самоцитирования** — фрагменты проверяемого текста, совпадающие или почти совпадающие с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа. Показатель «Самоцитирования» – это доля фрагментов текста, отнесенных к самоцитированию, в общем объеме текста.
- **Цитирования** — фрагменты проверяемого текста, которые не являются авторскими, но которые система отнесла к корректно оформленным. К цитированиям относятся также шаблонные фразы; библиография; фрагменты текста, найденные модулем поиска «СПС Гарант: нормативно-правовая документация». Показатель «Цитирования» – это доля фрагментов проверяемого текста, отнесенных к цитированию, в общем объеме текста.
- **Текстовое пересечение** — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
- **Источник** — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
- **Оригинальный текст** — фрагменты проверяемого текста, не обнаруженные ни в одном источнике и не отмеченные ни одним из модулей поиска. Показатель «Оригинальность» – это доля фрагментов проверяемого текста, отнесенных к оригинальному тексту, в общем объеме текста.

Обращаем Ваше внимание, что система находит текстовые совпадения проверяемого документа с проиндексированными в системе источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности совпадений или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

Номер документа: 244

Тип документа: Выпускная квалификационная работа

Дата проверки: 14.06.2024 14:01:33

Дата корректировки: 14.06.2024 14:13:29

Количество страниц: 62

Символов в тексте: 103826

Слов в тексте: 15996

Число предложений: 3975

Комментарий: не указано

14.06.2024 /Прищепа В.В./

ПАРАМЕТРЫ ПРОВЕРКИ

Выполнена проверка с учетом редактирования: Да

Выполнено распознавание текста (OCR): Нет

Выполнена проверка с учетом структуры: Нет

Модули поиска: IEEE, Интернет Плюс*, Переводные заимствования*, Коллекция НБУ, ИПС Адилет, Переводные заимствования по коллекции Интернет в английском сегменте, СПС ГАРАНТ: аналитика, Переводные заимствования издательства Wiley, Перефразирования по СПС ГАРАНТ: аналитика, Шаблонные фразы, Библиография, Переводные заимствования IEEE, Перефразированные заимствования по коллекции Интернет в русском сегменте, Цитирование, Кольцо вузов, Сводная коллекция ЭБС, СПС ГАРАНТ: нормативно-правовая документация, Диссертации НББ, Перефразирования по коллекции издательства Wiley, Перефразирования по Интернету, Публикации РГБ, СМИ России и СНГ, Публикации eLIBRARY (переводы и перефразирования), Патенты СССР, РФ, СНГ, Издательство Wiley, Перефразирования по Интернету (EN), Перефразирования по коллекции IEEE, Медицина, Переводные заимствования по Интернету (EnRu), Переводные заимствования (RuEn), Перефразированные заимствования по коллекции Интернет в английском сегменте, Переводные заимствования по коллекции Интернет в русском сегменте, Переводные заимствования по коллекции Гарант: аналитика, Кольцо вузов (переводы и перефразирования), Публикации eLIBRARY, Собственная коллекция компании

ИСТОЧНИКИ

№	Доля в тексте	Доля в отчете	Источник	Актуален на	Модуль поиска	Комментарий
[01]	4,2%	3,77%	https://winnspace.uwinnipeg.ca/b... https://winnspace.uwinnipeg.ca	11 Июн 2024	Интернет Плюс*	
[02]	3,34%	1,34%	https://www.mdpi.com/books/pdf... https://mdpi.com	18 Апр 2023	Интернет Плюс*	
[03]	2,86%	2,34%	https://www.politesi.polimi.it/bitst... https://politesi.polimi.it	14 Мар 2023	Интернет Плюс*	
[04]	2,15%	0,9%	https://aclanthology.org/2021.bio... https://aclanthology.org	30 Мая 2022	Интернет Плюс*	
[05]	1,71%	0,58%	Dual encoding for abstractive text... https://core.ac.uk	20 Янв 2023	Перефразированные заимствования по коллекции Интернет в английском сегменте	
[06]	1,71%	0,66%	Dual encoding for abstractive text... https://core.ac.uk	20 Янв 2023	Интернет Плюс*	
[07]	1,71%	0%	https://uwe-repository.worktribe.c... https://uwe-repository.worktribe.com	01 Янв 2023	Интернет Плюс*	
[08]	1,52%	0%	https://mdpi-res.com/d_attachme... https://mdpi-res.com	11 Июн 2024	Интернет Плюс*	
[09]	1,39%	0,7%	https://uwe-repository.worktribe.c... https://uwe-repository.worktribe.com	01 Янв 2023	Перефразированные заимствования по коллекции Интернет в английском сегменте	
[10]	1,21%	0,42%	https://oa.upm.es/71381/1/TFM_D... https://oa.upm.es	11 Июн 2024	Интернет Плюс*	
[11]	1,2%	0,31%	https://theses.hal.science/tel-033... https://theses.hal.science	22 Дек 2022	Интернет Плюс*	
[12]	0,71%	0%	http://itnt-conf.org/images/docs/2... http://itnt-conf.org	08 Фев 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[13]	0,7%	0%	Баймурзина, Диляра Римовна Не... http://dlib.rsl.ru	04 Апр 2022	Публикации РГБ	Источник исключен. Причина: Маленький процент пересечения.
[14]	0,68%	0%	ISIS tells British-based fanatics to ... http://dailymail.co.uk	06 Янв 2018	Перефразирования по Интернету (EN)	Источник исключен. Причина: Маленький процент пересечения.
[15]	0,64%	0%	https://mipt.ru/upload/medialibra... https://mipt.ru	30 Июн 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[16]	0,63%	0%	alaev_e_n_avtomaticheskaya-sum...	13 Мая 2020	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[17]	0,63%	0%	kulpin_p_l_proekt-avtomaticheskо...	27 Фев 2023	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[18]	0,61%	0%	https://arxiv.org/pdf/2204.01849.p... https://arxiv.org	28 Апр 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[19]	0,61%	0%	Automated Handling of Anaphoric... https://ieeexplore.ieee.org	20 Июн 2022	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[20]	0,58%	0%	Log-based Anomaly Detection Wit... https://ieeexplore.ieee.org	20 Янв 2022	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[21]	0,58%	0%	http://itnt-conf.org/images/docs/2... http://itnt-conf.org	24 Апр 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.

[22]	0,57%	0%	susla_d_m_primenenie-predobuc...	05 Июн 2021	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[23]	0,56%	0%	https://arxiv.org/pdf/2108.01064.p... https://arxiv.org	11 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[24]	0,5%	0%	https://aclanthology.org/2021.sem... https://aclanthology.org	29 Мар 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[25]	0,49%	0%	https://arxiv.org/pdf/2305.06147.p... https://arxiv.org	26 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[26]	0,47%	0%	http://donnu.ru/public/journals/fil... http://donnu.ru	04 Мар 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[27]	0,45%	0%	Front Matter https://ieeexplore.ieee.org	24 Окт 2023	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[28]	0,44%	0%	Abstracts of BERT-Related Papers https://ayaka14732.github.io	14 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[29]	0,43%	0%	Куратов, Юрий Михайлович Спе... http://dlib.rsl.ru	12 Янв 2021	Публикации РГБ	Источник исключен. Причина: Маленький процент пересечения.
[30]	0,42%	0%	SCSE 2022 Conference Proceedings https://ieeexplore.ieee.org	04 Окт 2022	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[31]	0,41%	0%	Extractive Text Summarization https://ieeexplore.ieee.org	05 Июн 2024	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[32]	0,41%	0%	Infographics Generator: A Smart A... https://ieeexplore.ieee.org	21 Мар 2024	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[33]	0,41%	0%	Хаммуд Жаафар; [Место защиты:... http://dlib.rsl.ru	01 Янв 2023	Публикации РГБ	Источник исключен. Причина: Маленький процент пересечения.
[34]	0,39%	0%	ВКР_Гордиенко_НА_ПИ19-1м Опр...	02 Июн 2021	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[35]	0,39%	0%	ВКР. Кочеров Сергей. АБД19-1м. ...	02 Июн 2021	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[36]	0,39%	0%	http://itnt-conf.org/images/docs/2... http://itnt-conf.org	28 Апр 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[37]	0,38%	0%	Large Language Models: A Survey https://arxiv.org	08 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[38]	0,37%	0%	Extractive Summarization - A Com... https://ieeexplore.ieee.org	30 Мая 2023	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[39]	0,36%	0%	solomatin_r_i_razrabotka-sistemy-...	02 Июн 2023	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[40]	0,35%	0%	Янина, Анастасия Олеговна Тем... http://dlib.rsl.ru	01 Янв 2022	Публикации РГБ	Источник исключен. Причина: Маленький процент пересечения.
[41]	0,33%	0%	A Comprehensive Review on Auto... https://scielo.org.mx	11 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[42]	0,33%	0%	Fusion of Text and Graph Informa...	30 Апр 2022	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[43]	0,33%	0,33%	ROUGE Score Analysis and Perfor... https://ieeexplore.ieee.org	15 Мар 2024	Перефразирования по коллекции IEEE	
[44]	0,33%	0%	MIPRO'2015. 38th International Co... http://elibrary.ru	11 Июл 2015	Публикации eLIBRARY	Источник исключен. Причина: Маленький процент пересечения.
[45]	0,27%	0%	Дубинина, Екатерина Юрьевна д... http://dlib.rsl.ru	раньше 2011	Публикации РГБ	Источник исключен. Причина: Маленький процент пересечения.
[46]	0,26%	0%	Neural Abstractive Summarizatio... https://ieeexplore.ieee.org	04 Авг 2023	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[47]	0,26%	0%	Quality Enhancement of Abstracti... https://ieeexplore.ieee.org	04 Янв 2023	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[48]	0,25%	0%	nekrasova_p_a_analiz-tehnologiy-s...	09 Янв 2024	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[49]	0,25%	0%	Abstractive Text Summarization fo... https://ieeexplore.ieee.org	31 Окт 2023	IEEE	Источник исключен. Причина: Маленький процент пересечения.
[50]	0,25%	0%	http://paijournal.guiaidn.ru/down... http://paijournal.guiaidn.ru	11 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[51]	0,25%	0%	Implementing AI Vertical use case... https://doi.org	09 Авг 2023	Издательство Wiley	Источник исключен. Причина: Маленький процент пересечения.
[52]	0,24%	0%	Дударин, Павел Владимирович ... http://dlib.rsl.ru	27 Июн 2022	Публикации РГБ	Источник исключен. Причина: Маленький процент пересечения.
[53]	0,23%	0%	ArA*summarizer: An Arabic text s... https://doi.org	30 Апр 2020	Издательство Wiley	Источник исключен. Причина: Маленький процент пересечения.
[54]	0,21%	0%	https://pure.rug.nl/ws/portalfiles/... https://pure.rug.nl	30 Мар 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[55]	0,21%	0%	Crossing the "Cookie Theft" Corpu... https://frontiersin.org	16 Апр 2021	СМИ России и СНГ	Источник исключен. Причина: Маленький процент пересечения.
[56]	0,21%	0%	Text-based sentiment analysis in ... https://doi.org	25 Фев 2024	Издательство Wiley	Источник исключен. Причина: Маленький процент пересечения.
[57]	0,2%	0%	Alzheimer's disease recognition fr... https://doi.org	29 Янв 2024	Издательство Wiley	Источник исключен. Причина: Маленький процент пересечения.

[58]	0,2%	0%	litewi: A combined term extractio... https://doi.org	29 Фев 2016	Издательство Wiley	Источник исключен. Причина: Маленький процент пересечения.
[59]	0,19%	0%	https://www.bnl.gov/education/do... https://bnl.gov	08 Янв 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[60]	0,18%	0%	Creating clear and informative im... https://journals.plos.org	11 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[61]	0,17%	0%	Артемов Кирилл; [Место защиты... http://dlib.rsl.ru	01 Янв 2022	Публикации РГБ	Источник исключен. Причина: Маленький процент пересечения.
[62]	0,16%	0%	https://journal.tusur.ru/storage/1... https://journal.tusur.ru	29 Мая 2023	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[63]	0,16%	0%	Parameters, Hyperparameters, Ma... https://towardsdatascience.com	11 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[64]	0,16%	0%	Creating a summary having sente... http://freepatentsonline.com	09 Ноя 2016	Патенты СССР, РФ, СНГ	Источник исключен. Причина: Маленький процент пересечения.
[65]	0,16%	0%	Method and system for calculatin... http://freepatentsonline.com	06 Ноя 2016	Патенты СССР, РФ, СНГ	Источник исключен. Причина: Маленький процент пересечения.
[66]	0,15%	0%	https://aclanthology.org/S19-2.pdf https://aclanthology.org	14 Мар 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[67]	0,15%	0%	Novak's gynecology [Текст] http://emll.ru	21 Дек 2016	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[68]	0,15%	0%	Optimization of an Abstract Text S... https://ieeexplore.ieee.org	15 Мар 2024	Перефразирования по коллекции IEEE	Источник исключен. Причина: Маленький процент пересечения.
[69]	0,15%	0%	Habteab Asmerom - Masters Thesis	14 Июн 2021	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[70]	0,15%	0%	Master Thesis Document.pdf	17 Июн 2023	Собственная коллекция компании	Источник исключен. Причина: Маленький процент пересечения.
[71]	0,14%	0%	https://grattoncourses.wordpress.... https://grattoncourses.wordpress.com	11 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[72]	0,14%	0%	https://www.iae.nsk.su/images/st... https://iae.nsk.su	07 Мая 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[73]	0,14%	0%	TEXT PROCESSING APPARATUS, TE... http://freepatentsonline.com	09 Ноя 2016	Патенты СССР, РФ, СНГ	Источник исключен. Причина: Маленький процент пересечения.
[74]	0,13%	0%	don_s_v_proekt-sravnitelnyy-anali...	20 Фев 2020	Кольцо вузов	Источник исключен. Причина: Маленький процент пересечения.
[75]	0,11%	0%	Ситтхисон	15 Июн 2023	Собственная коллекция компании	Источник исключен. Причина: Маленький процент пересечения.
[76]	0,1%	0%	https://uust.ru/media/dc/2424790... https://uust.ru	31 Мая 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[77]	0,1%	0%	Биостратиграфия и фораминифе... https://ibooks.ru	01 Янв 2020	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[78]	0,1%	0%	Палеобиогеографическое район... https://ibooks.ru	01 Янв 2021	Сводная коллекция ЭБС	Источник исключен. Причина: Маленький процент пересечения.
[79]	0,08%	0%	https://www.diva-portal.org/smas... https://diva-portal.org	20 Мая 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[80]	0,07%	0%	Guide to Image Segmentation in C... https://encord.com	07 Мая 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[81]	0,07%	0%	Международная конференция п... http://emll.ru	21 Дек 2016	Медицина	Источник исключен. Причина: Маленький процент пересечения.
[82]	0,07%	0%	Future Internet | Free Full-Text | ... https://mdpi.com	25 Апр 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[83]	0,06%	0%	https://web.stanford.edu/~jurafsk... https://web.stanford.edu	14 Мая 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[84]	0,05%	0%	https://www.govinfo.gov/content/... https://govinfo.gov	26 Дек 2022	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.
[85]	0,05%	0%	https://www.ablaikhan.kz/images/... https://ablaikhan.kz	01 Июн 2024	Интернет Плюс*	Источник исключен. Причина: Маленький процент пересечения.

УТВЕРЖДАЮ
Проректор по научной и инновационной деятельности Томского государственного университета____________А.Б. Ворожцов
«______» _____________ 2024 г.

ЭКСПЕРТНОЕ ЗАКЛЮЧЕНИЕ

о возможности открытого опубликования

Экспертная комиссия научно-образовательного центра «Высшая ИТ школа» Федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский Томский государственный университет» Министерства науки и высшего образования Российской Федерации,
рассмотрев выпускную квалификационную работу обучающегося Томского государственного университета по направлению подготовки 09.03.04 «Программная инженерия», направленность «ITS/TSU Software Engineering», гр.972006

Арисудана Геде Йога

на тему «Exploring the Capabilities of Luhn Summarization, BART, Pegasus, and Claude LLM for news summarization / Изучение возможностей метода Луна, BART, Pegasus и Claude LLM в реферировании новостей» (Выпускная квалификационная работа студентов посвящена сравнению различных подходов машинного обучения к обобщению текста. Первая глава работы посвящена описанию общей логики работы системы, которая будет разработана и использована для проведения сравнительного анализа. Вторая глава посвящена упоминанию и подготовке всех используемых моделей машинного обучения, метрик и стека технологий. В третьей главе описываются результаты обобщения, достигнутые студентом, и сравнение моделей с использованием различных показателей и подходов.)
подтверждает, что в материале не содержится информация с ограниченным доступом (Закон РФ «О государственной тайне», Перечень сведений, подлежащих засекречиванию Минобрнауки РФ № 31с от 04.12.2023), , а также не содержится информация, подпадающая под Списки, контролируемых товаров, технологий, утверждённых постановлениями Правительства Российской Федерации: № 1299 от 19.07.2022 двойного назначения; № 1284 от 16.07.2022 химикатов, оборудования, технологий; № 1285 от 16.07.2022 ядерных, специальных неядерных материалов, соответствующих технологий; № 1286 от 16.07.2022 оборудование и материалы двойного назначения, применяемых в ядерных целях; № 1287 от 16.07.2022 микроорганизмов, токсинов, оборудования, технологий; № 1288 от 16.07.2022 оборудования, материалов, используемые при создании ракетного оружия).
Материалы выпускной квалификационной работы ранее не публиковались в российских печатных и интернет-источниках.

Заключение:

Разрешить открытую публикацию выпускной квалификационной работы на тему «Exploring the Capabilities of Luhn Summarization, BART, Pegasus, and Claude LLM for news summarization / Изучение возможностей метода Луна, BART, Pegasus и Claude LLM в реферировании новостей».

Председатель комиссии:
Академический руководитель НОЦ ВИТШ
доктор физ.-мат. наук, профессор __________________ О.А. Змеев «14» июня 2024г

Члены комиссии:
Доцент учебного офиса НОЦ ВИТШ
кандидат техн. наук __________________ Д.О. Змеев «14» июня 2024г
Доцент учебного офиса НОЦ ВИТШ
кандидат физ.-мат. наук __________________ К.С. Ким «14» июня 2024г

СОГЛАСОВАНО:
Начальник первого отдела __________________ А.М. Амельченко «12» 06 2024г

Специалист
по экспортному контролю ОНТИ НУ __________________ Н.Н. Ходанович «14» 06 2024г