Abstractive Text Summarization *- SimpleT5*

Gauravkumar Singh Gaharwar   
Professor, *Navrachana University* Vadodara, India  
gauravsinghg@nuv.ac.in

Megh Patel   
Student, *Navrachana University*   
 *Bachelors of Information Technology*Vadodara, India  
meghpatel005@gmail.com

Dhwani Modi   
Student, *Navrachana University*   
 *Bachelors of Information Technology*Vadodara, India  
dcmodi23052000@gmail.com

*Abstract*—As the trend of big data and data generation on the internet is rapidly rising, text summarization has become vital and an important aspect. The majority of modern recommender and text categorization systems necessitate a massive quantity of data analysis. In addition, creating exact and eloquent summaries of lengthy publications by hand is a time-consuming and exhausting endeavor. As a result, creating automatic data summaries and using them to train machine learning models will save both space and time. There are two different approaches for summarizing the text: Extractive and abstractive summarization. The extractive methodology extracts few apposite words and sentences from the input text. The summary is constructed after understanding the original text in abstractive summarization techniques, which makes it more sophisticated. In this report paper, we will be proposing an intensive report of the transformer architecture based pre-trained model: SimpleT5 for the purpose of text summarization. For analysis and comparison, we have used the news dataset that contains text data that can be used for the purpose summarization and human generated summaries for comparing and contrasting the summaries generated by machine learning model.

Keywords— (NLP) Natural Language Processing, Deep Learning, Summarization, Transformers, Abstractive, Gradio, Encoder-Decoder, (EDA) Exploratory Data Analysis, (ML) Machine Learning, (GloVe) Global Vectors.

# Introduction

* 1. *Need for Text Summarization*

The goal of text summarising is to construct a concise summary from a long document or text article while preserving all of the content. Deep Learning has gained notable importance in recent years for the purpose of computing the text summaries.

Automating summarization [1] would eliminate manual efforts. Shorter texts, which are summaries of longer texts, would reduce reading time. With the advent of big data generation, text summarization would optimize the size of files and hence resolve problem of data storage. A shorter text or summary would provide more significant insights. Furthermore, precise summaries are extremely important in text mining and data analysis.

*1.2 Summarization Techniques*

There are two general ways to text summarization: [2] –

**Extractive Summarization** - Extractive summarization is the mechanism of generating a summary from a given input text by picking a subset of the overall sentence base [3]. The most essential phrases or sentences from the text are discovered and chosen based on a score calculated from the words in the sentence.

**Abstractive Summary** - An interpretation is initially formed by analysing the text supplied in the abstractive summarization approach. Based on this knowledge, the system provides a summary. By paraphrasing portions of the original source, the text is altered.

When compared to extractive summarization, abstractive summarization is more efficient because it gathers information from several publications to build a precise summary of data. This has grown in popularity as a result of its capacity to create new sentences to convey vital information from text documents. An abstractive summarizer presents the summed data in a logical, grammatically correct, and easily understandable way. The generated summary's readability, or linguistic quality, is an important factor in its overall quality.

Diagram

Description automatically generated

Fig 1 – Overview of Abstractive Summarization

# Research Reviews

After studying and analyzing various research papers, we’ve understood how different algorithms are useful in summarizing the text along with their different application areas. [4][5]

**Extractive**:

* Algorithm: Ranking methods (TextRank, LexRank)
* Extracting sentences
* Relatively easier to implement
* Perfect for short documents

**Abstractive**:

* Algorithm: NLP Techniques (Seq-2-Seq Model)
* Generating sentences
* Complex implementation
* Perfect for long documents

**Combined Approach:**

There are mixed approaches, in which an abstractive generator is used taking as input a text coming from an extractive summarizer. In this way the abstraction/generation process is more efficient because it works on a text already purged of all redundancies and irrelevant information.

**Evaluation Systems:**

ROUGE, GLUE, Race, SQuaD

**Automatic text summarization algorithms:**

Here is the list of some of the most important algorithms [5] that are used significantly in the recent times. Moreover, these are highly trained ones, so they are very accurate in summarizing even very large documents.

1. GPT (generative pretrained transformer)
2. BERT (bidirectional encoder representation for transformers)
3. BART (bidirectional autoencoder representation for transformers)
4. XLNET
5. UNILM (Unified Language Model Pre-training for Natural Language Understanding and Generation)
6. PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization)

# Application Areas

Text Summarization is a pioneering method that has found applications [6] in Natural Language Processing (NLP) use cases such as Question Answering and Text Classification, as well as other computer science fields such as Information Retrieval. These systems might incorporate the creation of summaries as a step in the process, with the goal of lowering document length, because Geographical Information Retrieval is considered as an add-on to the Information Retrieval discipline. In a variety of commercial applications, text summarization can be extremely useful, which includes:

**Newsletter**: Many weekly newsletters begin with an introduction and then feature a handpicked selection of related content. Summarization would allow companies to supplement newsletters with a stream of summaries (rather than a list of links), which is a more mobile-friendly format.

**Search marketing and Search Engine Optimization (SEO):** It's vital to have a thorough understanding of what your competitors are talking about in their content when evaluating search queries for SEO. This is especially crucial now that Google has altered its algorithm and moved its focus to topical authority (versus keywords). Multi-document summary is a useful method for swiftly analyzing many search results, identifying common themes, and skimming the most relevant aspects.

**Financial research:** Investment banks spend a lot of money on research to help them make decisions, which includes computerized stock trading. If you're a financial analyst who spends every day reading market reports and news, you'll eventually reach a wall and won't be able to read everything. Financial document summarization systems, such as earnings reports and financial news, can aid analysts in quickly extracting market signals from information.

**Legal contract analysis:** Similar to internal document workflow, more precise summarizing systems for analyzing legal documents could be developed. In this situation, a summarizer could be useful in compressing a contract down to the hazardous provisions or assisting you in comparing contracts.

**Social Media Marketing:** Companies who create long-form information, such as whitepapers, e-books, and blogs, may be able to use summary to break down this content and make it shareable on social media sites like Twitter and Facebook. Companies would be able to reuse old content even more effectively as a result of this.

**Question answering and bots:** Personal assistants are taking over the workplace and the smart home with question answering and bots. On the contrast, most bot assistants are constrained to particular set of responsibilities. Summarization on a large scale could be a useful strategy for addressing questions. A summarizer could create a cohesive answer in the form of a multi-document summary by collecting the most relevant documents for a certain issue.

# Proposed System

Abstractive Summarization includes interrogative approaches to upskill the system to acknowledge the whole context and give a well-structured summary based on the fundamental understanding. This is a more human-like way of generating summaries and these summaries are more effective as compared to the extractive approaches.

*4.1 Abstract Model*

Diagram

Description automatically generated

Fig 2- Proposed Abstract Model

The goal is to summarize text with the use of Abstractive technique. Firstly, the input will be either in the form of text or the article link. These data will be preprocessed and cleaned so that the unwanted data from the dataset using EDA (Exploratory Data Analysis) is eliminated. Further, analysis of attribute will be performed using Seq2Seq model and then the model will be trained using BART transformers. After successfully deploying the model, the metrics are evaluated using the ROUGE technique.

**Input**: This application will accept input as text or the article’s URL from which the text is extracted and is sent to the next processing phase.

**Text-preprocessing**: It cleans the text data and make it ready for the model to accept it by using various EDA methods.

**Seq2Seq Model**: It transfers a sequence input to a sequence output with a tag and attention value. Seq2Seq model aids in predicting the words for the desired summary.

**GloVe**:

GloVe (Global Vectors) is a representation for the distributed words, measured in vectors. It is an unsupervised learning algorithm used in NLP techniques for measuring vectors of words. This is accomplished by mapping words into a meaningful space in which word distance is proportional to semantic similarity.

A word embedding is a learnt text representation in which words with related meanings are represented similarly. One of the significant achievements of deep learning on tough natural language processing problems may be this way to expressing words and documents.

Chart, line chart

Description automatically generated

Fig 3- Abstractive model graph (1)

A picture containing chart

Description automatically generated

Fig 4 - Abstractive model graph (2)

From fig.3 it can be inferred how the model measures vectors and provides a attention tag to the words. The value is determined based on their meaning, word length and their similarity. Like the word ‘Verizon’ and ‘Vodafone’ are placed next to each other in graph based on their similarities.

However, from the fig.4 it can be inferred that how related words but opposite to each other like ‘Men’ and ‘Women’ are placed in the graph.

Likewise, other words are placed accordingly with their measured tags. This technique of measuring the words helps the model in predicting the words while generating human-like summary of the text.

**BART Transformers**: BART is a de-noising autoencoder for the sequence-to-sequence model that has been pre-trained. It learns by changing the text input with a random noising function and then teaching a model to rebuild the original text.

**ROUGE Algorithm**: ROUGE metrics compares and contrasts an autonomously generated text summary by the model to a referenced human-based summary for the sake of evaluation.

**Output**: The trained model will generate summarized output and will be displayed on the user interface in text format.

*4.2 Proposed System Architecture*

Diagram

Description automatically generated

Fig 5- Proposed System Architecture

The input will be in the form of text or in as the article link. At the data gathering layer, the article data will be fetched from the provided input. Secondly, EDA will be performed at the Data Preprocessing Layer for ensuring the integrity of data. Next, at Feature calculation layer, lexical analysis will be performed to predict the words and their rankings.

For model training layer, the dataset will be spitted into the ratio of 80:20. Finally, the trained model will be evaluated on the factors such as accuracy, precision and recall at Performance evaluation layer.

**Data Gathering Layer**: This layer fetches the text data from the user input whether it might in the text format or in the URL format.

**Data Preprocessing Layer**: In this phase, the extracted text is cleaned with the help of EDA methods, so that the unwanted disturbances in the raw data can be removed.

**Feature Calculation Layer**: This layer makes use of encoder-decoder model with the help of which the model can perform necessary lexical and semantic analysis.

**Model Training Layer**: In this phase, the original data is spitted into training and testing data respectively for the purpose of training the model. This further aids in recognizing of contextual relations between the words.

**Performance Evaluation Layer**: This layer performs necessary metric calculations to predict the accuracy, precision and recall of the resultant output.

*4.3 Proposed Flow chart*

Diagram

Description automatically generated

Fig 6- System Flow Chart

The proposed flow chart makes use of various python libraries to extract and clean the data for the model to be trained. This pre-processed data is then divided into training and testing sets and are fed to the training model.

Transformer model makes use of encoder-decoder model so as to train the layers, this will further help the model to predict words based on the lexical and semantic analysis performed.

*4.4 Model Explanation*

## **T5**: Google's answer to the world for open source language models was the "Text-to-Text Transfer Transformer." The T5 paper demonstrates that employing the entire encoder-decoder design (of the transformer) is superior to using only the decoder (like the GPT series does), thus they stay loyal to the original transformer architecture.

**Transformer Model**: The encoder and decoder layers are included in the transformer model [7]. Each layer feeds the forward network levels and is coupled to a multi-head attention layer. The model uses cosine and sine functions to recall the location and sequence of words, resulting in positional encoding. The encoder and decoder stages of the multi-head attention layer employ the self-attention mechanism. The input is sent via three linked layers to create query (Q), key (K), and value (V) vectors. These vectors are separated into n vectors in total.

Text

Description automatically generated

**Encoder-Decoder Model:**

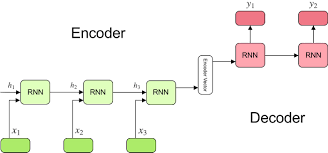


Fig 7- Attention Mechanism

The encoder-decoder paradigm is a method of solving the sequence-to-sequence forecast issue using recurrent neural networks [8].

**Encoder**: A stack of cyclic units of: LSTM or GRU cells for higher performance, that takes a single input sequence element, assemble the information for that particular element, and propagates it further.

**Decoder**: A set of recurrent units, each of which forecasts an output y t at a given time step t. Each repetitive unit collects a secret state from the previous unit and builds its own hidden state.

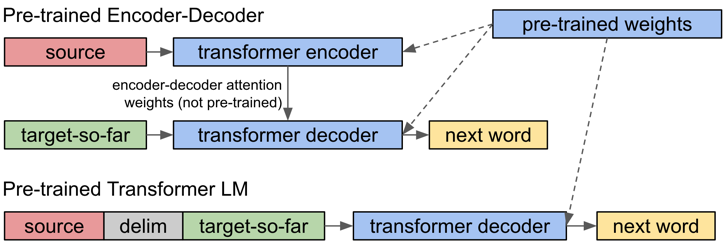


Fig 8- Trained encoder-decoder model

This model uses the same network of encoder-decoder to predict the source and target.

The attention mechanism of transformer uses words from training, weigh them and helps it to predict the future words.

**Attention Mechanism:**

The attention mechanism [9] is a part of a neural architecture that allows users to dynamically highlight significant portions of incoming data, which in NLP processing is often a collection of textual components. It can be used to either raw data or a higher-level representation of it.

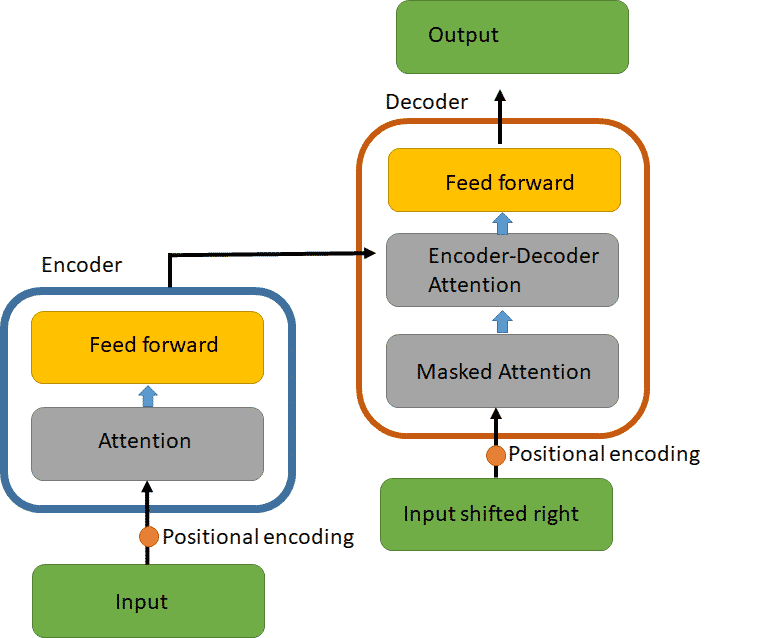


Fig 9- Attention Mechanism

## 4.5 User Interface

Gradio is a Python toolkit that lets you easily generate customisable UI components for your machine learning model, any API, or any random function in minimum lines of code. You may either embed the GUI into your Python notebook or send the URL to anyone.

It's quick, easy to set up, and ready to use, and it's shareable as a public connection that anyone can use to run the model remotely and concurrently on your system. Gradio works with a diverse variation of media like: text, images, video, and sound. Apart from machine learning models, it may also be used for python code embeddings.

## 4.6 Model Evaluation

ROUGE, or Recall-Oriented Understudy for Evaluation [10], is a collection of metrics that is used in NLP processing to evaluate the automatically generated summary and the machine translated technologies. The metrics contrast between an autonomously produced summary or translation against a human-authored summary or translation.

* **ROUGE-N-** It measures the overlap [11] among **unigram, bigram, trigram, and higher order n-grams**
* **ROUGE-L** - Using LCS, the longest matching sequence of words is established.
* **ROUGE-S-** The overlap of word pairs [12] with a maximum of two gaps between them is measured by skip-bigram.

# Implementation

The basic experimental technique will be presented in this part, followed by a discussion of the evaluation metrics and a description of several Transformer models that can be used. Then we'll put our findings together and explain how the model performed [13].

*5.1 Application specifications*

**Language used**: Python 3

**IDE**: Google Colab

**Browser**: Chrome, Safari

**GPU**: required (Training Model)

**RAM**: min 8GB

**Storage**: min 15GB

**Libraries**:

1. TensorFlow - 1.15
2. Pandas - 1.3.4
3. Sklearn - 0.22
4. Matplotlib – 3.5.0
5. ROUGE – 1.0.1

*5.2 Dataset*

For the data, we used a dataset that consist of news articles along with their summarized text for the text classification. It consists of 95,000 rows of data including the text article along with its summarized data.

Total Data: 95000 rows

Trained Data: 16000 rows

Testing Data: 4000 rows

Splitting Proportion: 80:20

*5.3 Pre-processing*

This dataset had long news snippets along with their short summaries for the purpose of comparison [14]. The raw dataset was then cleaned and trimmed using several pre-processing techniques like:

**Lower casing** - To convert the input text into the same casing format so that all capital, lower case and mixed case are treated similarly.

**Eliminate Punctuation** - HTML tags and links- Removal of punctuations, links and tags [15] that do not add meaning to the text such as “!"#$%&\'()\*+,-./:;<=>?@[\\]^\_{|}~`” to standardize the text.

**Eliminate Stop words and frequently occurring** words – The elimination of common repetitive words such as ‘the’, ‘a’, etc that are frequently used in a text but do not provide valuable information for downstream analysis.

**Targeting Columns** - Including the necessary data columns for training purpose.

*5.4 Libraries Used*

1. **simpleT5** – Transformer model is used for summarizing the text using Natural language processing (NLP).

The T5 Transformer is an encoder-decoder composition with text sequences as both the input and output. This allows it to do any Natural Language Processing task without having to make any changes to the model design. It also implies that a single T5 model can be taught to execute several tasks at the same time.

1. **TensorFlow** – For training and analysing the built models. TensorFlow is the prominent and widely used open-source deep learning framework. It's a machine learning and artificial intelligence software library. It can be used for a variety of applications, but it focuses on deep neural network training and inference.
2. **Pandas** – For exploratory data analysis (EDA)

Pandas is a NumPy-based open-source Python library. It's a Python module for manipulating numerical data and time series with a range of data formats and methods. The basic purpose is to simplify data input and analysis at greater extent. It is based on matplotlib for data visualization and NumPy for mathematical calculations, which are both essential Python modules.

1. **Sklearn** – Machine learning library for statistical models.

Scikit-learn is a free Python machine learning library. It supports Python numerical and scientific libraries like NumPy and SciPy, as well as algorithms like support vector machine, random forests, and k-neighbours.

1. **Matplotlib** – Visualizing the data [16]

One of the prime benefits of visualization is that it gives us visual access to large volumes of data in simple images. Line, bar, scatter, histogram, and more graphs are available in Matplotlib.

1. **Gradio** – Python based GUI library tool

It is based on python’s flask framework which helps in creating user interface especially for machine learning model with minimum ease with the assistance of API.

1. **Rouge** - Recall-Oriented Understudy for Evaluation

It is an evaluation algorithm that calculated metrics such as f2 score, precision and recall values that helps in understanding the efficiency of model.

*5.5 Visualizing text parameters*

*Chart

Description automatically generated*

Fig 10- Word counts

In order to understand the model hyperparameters, text and summary fields of the dataset are visualized with the help of matplotlib library.

From the graphs, it can be interpreted that the average word length of text is 60 words and that of summary is around 10 words.

This can help in determining the model parameters such as max and min token length while training the custom simpleT5 model.

*5.6 Training the model*

*Text

Description automatically generated*

Fig 11 – Training Model

**Model Hyperparameters:**

Data is splitted into training and testing data frames using panda’s library and then it is assigned to their respective variables in the train method.

As visualized in the figure 5, max and min token length are defined to 120 and 15 respectively. Source and target token lengths are set a little higher so as to accept and deliver user inputs.

The batch size is set to 16 as per the GPU limitations of the system.

Epochs are set to 15 so as to train the model more accurately with more iterations with the help of GPU. Moreover, the early stopping variable is set to 0 as the model needs to be trained on large set of data.

*5.6 Trained Model Analysis*

**TensorBoard** is a platform that provides the measurements and visualisations required for machine learning. TensorBoard lets you keep track of experiment parameters like loss and accuracy, see the model graph, and project embeddings to a lower-dimensional space.

**Epochs**: In machine learning, an epoch is the number of passes the machine learning algorithm has performed through the whole training dataset.

In out project we have defined epochs to 15 so as to minimize the training loss and determine the value gained by the model. As a result, a total of 20000 rows are fed into the model.

Chart, line chart

Description automatically generated

Fig 12 – Epochs

**Train Loss**: It is the amount of the objective function that you're decreasing. This number could be positive or negative depending on the specific objective function of your training data. The training loss is measured over the entire training dataset.

Chart, line chart

Description automatically generated

Fig 13 – Training loss

As the model has been trained for 15 epochs, the train loss is observed to be decreasing gradually, reaching a loss of around 0.2605 at the 15th epoch.

**Value Loss**: The loss amount depicts how well or how poorly the model performs after each optimization is performed.

Chart, line chart

Description automatically generated

Fig 14 – Value metric

From the graph, it can be interpreted that the model has been in the most efficient stage between the 2nd and the 7th epochs.

# Results

## **Text as an input**

This technique accepts text input from the user, this text is then loaded into the specified summarization model using Gradio user interface.

*Example 1*:

Input:

Graphical user interface, text

Description automatically generated

Output:

Graphical user interface, text, application

Description automatically generated

*Example 2*:

Input:

Text

Description automatically generated

Output:

Text

Description automatically generated

## **URL as an input**

*Example 1*:

Input:

A picture containing graphical user interface

Description automatically generated

Output:

Graphical user interface, text, application

Description automatically generated

*Example 2*:

Input:

Graphical user interface

Description automatically generated with medium confidence

Output:

Graphical user interface, text, application

Description automatically generated

*6.1 Qualitative Analysis:*

To create summaries, we fine-tuned the following transformer-based pre-trained language models [17] from the Hugging face library. The News Dataset was used to create summaries that included text as well as human created summaries, which are summaries written by humans.

**Example Summary***:* [18] *“An incident of robbery that occurred at the shopping complex last night was reported at the local police station this morning. A lot of valuables were stolen and multiple such robberies have been reported in that area. The people have been asked to stay alert and notice any suspicious activity. A CCTV camera from a nearby house captured the incident and there were a total four robbers who can be seen carrying bags.* *The merchant has sustained a loss and is hoping that the police will apprehend the perpetrators.”.*

**T5 Model**:

The following table contains the output of summarized text from the simpleT5 model trained in this project.

1. summary generated by simpleT5 model

| Models | Summaries generated by models |
| --- | --- |
| T5 | “Shopkeeper suffered loss and hopes police catch the culprits. four robbers can be seen carrying bags at the shopping complex.” |

The following table compares the output from different summarization models from the transformer library.

1. summary generated by different models

| Models | Summaries generated by models |
| --- | --- |
| Pipeline-BART | “Last night, an incident of a robbery was reported at the shopping complex. According to the shopkeeper, as he reached his shop in the morning, he found that the door was already open and many valuable items were stolen from the shop. They broke into the shop and within 30-40 min came back with bags full of valuable items.” |
| BART modified | “Last week an incident of a robbery was reported at the shopping complex. The shopkeeper who has a shop in the complex said this incident to the local police station in the morning. The police suspected that these burglaries are somehow show that the shopkeeper’s association has issued a notice” |
| PEGASUS | “The shopkeeper who has a shop in the complex said this incident to the local police station in the morning as they found out about the robbery in the according to the as he reached his shop in the he found that the door was already open and many important products were taken from the business, and he immediately phoned the local police station and filed a report. The shopkeeper is hopeful that the police will find the thieves as soon as possible so that he can recover the items that were stolen from his there have been multiple such complaints in the past few weeks in this and the police suspect that these robberies are somehow the shopkeepers association has issued a notice and asked everyone in the area.” |

**Pipeline**: The pipeline model created summaries that concentrated on irrelevant sentences and contained sentences that varied the greatest from the original reference summary.

**BART**: In the pipeline method, BART outperformed the pre-trained Bart model. The generated summaries were fluid, accurate, and included supporting information from the input source. As a result, the pre-trained BART model's summaries show that the BART model is useful for text interpretation.

**T5**: The T5 model produces good outcomes and has a higher Rogue score F- value than the other models. The summaries produced are accurate and coherent. The meaning of the text was kept in these summaries, which were well aligned with the original summary.

**PEGASUS**: We achieved fluent and coherent summaries after fine-tuning and utilising this model for our dataset. The linguistic quality of these summaries was excellent, and they closely matched the style of ground truth reports. However, the results suggest that the generated summaries were too brief and incomplete.

*6.2 Quantitative Analysis:*

1. COmparison of ROUGE Scores

| Models | Evaluation Metrics | | |
| --- | --- | --- | --- |
| ROUGE - 1 | ROUGE - 2 | ROUGE - L |
| Pipeline - BART | 0.38 | 0.28 | 0.38 |
| BART modified | 0.40 | 0.28 | 0.40 |
| **T5** | **0.47** | **0.33** | **0.42** |
| PEGASUS | 0.42 | 0.29 | 0.40 |

Overall it can be seen that the T5 model has performed well in comparison to other summarization models.

T5 model outperformed all other models in evaluation of ROUGE metrics [19].

# Evaluation

Evaluated f1 score, precision and recall by comparing several human-written summaries to our model outputs.

**Recall**: It calculates the words extracted from the reference summary [20].



**Precision**: Calculates the relevant words.

A picture containing text

Description automatically generated

**F1 Score**: It calculates the balance between precision and recall.

Text

Description automatically generated

1. ROUGE scores

| Test Cases | ROUGE Parameters | | |
| --- | --- | --- | --- |
| F1 score | P | R |
| 1 | 0.2650 | 0.3928 | 0.2000 |
| 2 | 0.4742 | 0.4565 | 0.4509 |
| 3 | 0.2857 | 0.3478 | 0.2424 |
| 4 | 0.3551 | 0.4750 | 0.2835 |
| 5 | 0.3098 | 0.3666 | 0.2682 |
| 6 | 0.1684 | 0.2352 | 0.1311 |
| 7 | 0.3260 | 0.4054 | 0.2727 |
| 8 | 0.1897 | 0.3611 | 0.1287 |
| 9 | 0.2905 | 0.4250 | 0.2077 |
| 10 | 0.2831 | 0.4571 | 0.2654 |

# Literature Riview

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No** | **TITLE** | **AUTHOR** | **DATASET** | **METHOD** | **OUTCOME** |
| **1** | Semantic graph reduction approach for abstractive Text Summarization [21] | Ibrahim F. Moawad, Mostafa Aref | Scientific Document, Thesis paper. | Semantic Graph Reduction | The method summarises the input document by constructing an abstractive summary from the reduced graph after establishing a rich semantic graph for the original material. |
| **2** | Automated Text Summarization in SUMMARIST [22] | Eduard Hovy and Chin-Yew Lin | 1000 Documents from: Wall Street Journal, Associated Press, and Federal Register | Graph based approach | Abstractive Summary generation of redundant data |
| **3** | Automatic Text Summarisation [23] | Tarun Aggarwal, Vishal Tyagi, Utkarsh Dwivedi, Yash Kumar Sharma | News articles, and Magazines articles. | NLP | Extracting only meaningful excerpts from a large document or website. |
| **4** | Document Summarization and Keyword Generation using Graph-Based Ranking Algorithm [24] | [M. Senthilraja](https://www.researchgate.net/profile/M-Senthilraja-2?_sg%5B0%5D=IGS7w25dT0goi1Dt7Ahg6OJZHcAW0LlWZwznT8ROVuHdeWisUzGywaTP1YDLR44zy2yMVJI.MaY7hljwzxlIq7UpSf1QFdeYY0C6vU2ZNBoaQpCyyIFx-qqqw5KauP0Bdyik2sJWtCRgSs99iAB3BP4g3dythA&_sg%5B1%5D=mypDtbQk4ndJRFbbKzpuf81NJ1foSsM9wuIVOF-eT0DD0xuePXzAI1j12dlR0GQ7AAdTTcw.YuIn7eFiF8bLE7rj7JTQt2cDzTQll7k_rAQBHZoIxLuQvAliUss_imYUJOoVYp204sEjMntrdIf1xzYE_obFIQ), [Srinivasan Rajendran](https://www.researchgate.net/profile/Srinivasan-Rajendran?_sg%5B0%5D=IGS7w25dT0goi1Dt7Ahg6OJZHcAW0LlWZwznT8ROVuHdeWisUzGywaTP1YDLR44zy2yMVJI.MaY7hljwzxlIq7UpSf1QFdeYY0C6vU2ZNBoaQpCyyIFx-qqqw5KauP0Bdyik2sJWtCRgSs99iAB3BP4g3dythA&_sg%5B1%5D=mypDtbQk4ndJRFbbKzpuf81NJ1foSsM9wuIVOF-eT0DD0xuePXzAI1j12dlR0GQ7AAdTTcw.YuIn7eFiF8bLE7rj7JTQt2cDzTQll7k_rAQBHZoIxLuQvAliUss_imYUJOoVYp204sEjMntrdIf1xzYE_obFIQ), [S Iniyan](https://www.researchgate.net/scientific-contributions/S-Iniyan-2121378813?_sg%5B0%5D=IGS7w25dT0goi1Dt7Ahg6OJZHcAW0LlWZwznT8ROVuHdeWisUzGywaTP1YDLR44zy2yMVJI.MaY7hljwzxlIq7UpSf1QFdeYY0C6vU2ZNBoaQpCyyIFx-qqqw5KauP0Bdyik2sJWtCRgSs99iAB3BP4g3dythA&_sg%5B1%5D=mypDtbQk4ndJRFbbKzpuf81NJ1foSsM9wuIVOF-eT0DD0xuePXzAI1j12dlR0GQ7AAdTTcw.YuIn7eFiF8bLE7rj7JTQt2cDzTQll7k_rAQBHZoIxLuQvAliUss_imYUJOoVYp204sEjMntrdIf1xzYE_obFIQ) | News articles | Graph-Based Ranking Algorithm | The summarizer considers information recursively extracted from the full text, which is iterated until convergence is attained. |
| **5** | Deep Learning Based Abstractive Text Summarization [25] | Arafat Awajan, Dima Suleiman | CNN/Daily Mail datasets | Deep Learning | The most common strategies for abstractive text summarization, according to this study, are recurrent neural networks with an attention mechanism and long short-term memory (LSTM). |
| **6** | Automatic Text Summarizations [26] | Nesreen Alsharman | Document understanding workshop, Text Analysis Conference, Computational Linguistics Scientific Document Summarization Shared Task Corpus, Opinosis | NLP | A final summary is constructed using extracted unigrams and bigrams, as well as other attributes. |
| **7** | Text Summarization Techniques and Applications [27] | Bhavya Joshi, Pawan Kartik,  Gaurav Aggarwal, Virender Dehru and Pradeep Kumar Tiwari | Kaggle: News Summary & Food reviews from Amazon | Text Rank Algorithm, LSTM Method | This research shows the statistical-based algorithms can be used to generate fast and decent summaries. |
| **8** | Synthesis Abstractive Text Summarization [28] | Amine Benkhouya, Hemza Rahmani | CNN/Daily mail corpus and Gigaword corpus, DUC corpus | Sequence-to-Sequence RNNs | The architecture of an attentional encoder-decoder for an abstractive text summarization job is summarised. |
| **9** | Automatic Text Summarization of Articles in Wikipedia [29] | Srividhya Vasudevan | Wikipedia Articles | Sentence extraction method. | The extractive approach involves choosing the top N most sentences that best spread the entire information uttered by the original source content. |
| **10** | Deep Extractive Text Summarization [30] | Rupal Bhargava, Yashvardhan Sharma | Para Multiling 2015 dataset | Machine Learning and Deep Learning Approach. | In this study, paraphrase detection is utilised to determine whether or not a statement belongs in an extracted summary. |
| **11** | SummerTime: Text Summarization Toolkit for Non-experts [31] | Dragomir Radev, Yusen Zhang, Tao Yu, Zhangir Azerbayev, Ansong Ni, Ahmed Hassan Awadallah, Troy Feng, Mutethia Mutuma | News Articles from CNN and Daily Mail. | *Single-document:* TextRank, LexRank, BART  *Multi-document:* Query based, TF-IDF | It can automatically identify the best models or pipelines for a specific dataset and task, and visualize the differences between the model outputs and performances. |
| **12** | Extractive Text Summarization System for News Texts [32] | Fahrettin Horasan, Burhan Bilen | Business, politics and sports documents | ROUGE-N, TextRank | The paper work can be integrated to search engines to gather summaries of news or any text. |
| **13** | Topic Modeling Based Extractive Text Summarization [33] | Kalliath Abdul Rasheed Issam, Shivam Patel, Subalalitha C. N | WikiHow and CNN/Daily Mail datasets. | ROUGE and LDA methods. | Topic modelling was used to identify important subjects in a document that needed to be summarised, and text clusters were created around those topics. The final summary of the input document is formed by combining the summaries obtained for each of these text clusters. |
| **14** | Text Summarization in the Biomedical Domain [34] | Milad Moradi, Nasser Ghadiri | NLM WSD, MSH WSD dataset | NLP, Text Mining | To summarise biomedical content using the most important research approach. |
| **15** | Comparative Study on Abstractive Text Summarization [35] | Md Ashraful Islam Talukder, Sheikh Abujar, Sharmin Akter | Newspaper articles | Word graph methodology, semantic graph reduction, Markov clustering | All three methods work with different algorithms and different mechanism. |
| **16** | Keyphrase Generation: A Text Summarization Struggle [36] | Erion C ̧ ano, Ondˇrej Bojar | Scientific papers | encoder-decoder variants based on LSTMs | Experimented various un-supervised, supervised, deep supervised and abstractive text summarization models for predicting key phrases of scientific articles. |
| **17** | Text summarization from legal documents: a survey [37] | Ambedkar Kanapala, Sukomal Pal, Rajendra Pamula | Legal documents | Latent semantic analysis, Graph based approach, ROUGE | Experimented various summarization techniques on legal documents. |
| **18** | Text summarization using Wikipedia [38] | Sankar Subramaniam, K. Ramanathan, S. Ghosh | Wikipedia articles | Graph based ranking,  Query-focused summarisation | Demonstrating how utilising Wikipedia can improve summary quality significantly. |
| **19** | The method of multi-dimensional approach to text summarization [39] | Piotr Janaszkiewicz, Przemysław Różewski | DBpedia, WikiData | LSA, TextRank | Semantic relationship was used to create a relationship graph. |
| **20** | Compressive Cross-Language Text Summarization [40] | Elvys Linhares Pontes | English and French articles | NLP | The proposed system examines documents in both languages in order to extract all essential data. Then it compresses phrases using two different approaches. |
| **21** | Text Summarization using Centrality Concept [41] | Ghaleb Al\_Gaphari, Fadl M. Ba-Alwi, Aimen Moharram | Arabic NEWSWIRE, GIGAWORD | Graph based centrality algorithm | For Arabic Text Summarization, the Centroid-Based Algorithm (CBA) is used. |
| **22** | Abstractive Text Summarization using Transfer Learning [42] | Ekaterina Zolotareva, Tsegaye Misikir Tashu, Tomáš Horváth | BBC News dataset provided by Kaggle | Transformer model architecture | Used a newly introduced approach, the Transformer or T5 framework, to create a multi-sentence summary. |
| **23** | Text Summarization using Abstract Meaning Representation [43] | Shibhansh Dohare, Harish Karnick | CNN-Dailymail | Abstract Meaning  Re-presentation (AMR). | Observed the problems with the existing long text dataset and the evaluation metric for summarization through AMR. |
| **24** | Text Summarization Using Thematic Hierarchy Algorithm [44] | Aparna Khare, Prof. Lalji Prasad | Wikipedia articles | Thematic Hierarchy Algorithm | This algorithm includes the detailed relations in the chaining process and extract the text on the basis of user’s choice. |
| **25** | Automatic Text Summarization Using Fuzzy Inference [45] | Mehdi Jafari, Amir Shahab Shahabi, Xiaohui Tao | News articles | Fuzzy logic | For text summarization, fuzzy logic is used with standard extractive and abstractive techniques. |
| **26** | An approach to automatic text summarization using WordNet [46] | Alok Ranjan Pal, Diganta Saha | Wordnet dictionary | Lesk algorithm | Using the Simplified Lesk algorithm, this method assesses the weights of all the sentences in a text independently and organises them in decreasing order based on their weights. |
| **27** | Query-based Text Summarization using Averaged Query [47] | Abhinandh Ajay, Shravan V, R. Srinivasan | DUC dataset,  (TD-QFS) dataset | AQSum Algorithm | The goal was to offer a succinct overview of a topic that was linked to the user's query. The resulting summary is ten lines long and contains the most relevant sentences to the query. |
| **28** | Text summarization using Latent Semantic Analysis [48] | Makbule Gulcin Ozsoy, Ferda Nur Alpaslan | Duc2002, Duc2004, and Summac datasets. | LSA Algorithm | The results reveal that among all the LSA-based techniques, the cross method outperforms them all. Another significant benefit of this method is that it is unaffected by various input matrix generation methods. |
| **29** | An Efficient Text Summarizer Using Lexical Chains [49] | H. 'gregory Silber, Kathleen F Mccoy | Wordnet | Lexical chain algorithm | They have described a domain independent summarization engine which allows for efficient summarization of large documents. |
| **30** | Text Summarization using Random Indexing and PageRank [50] | Par Gustavsson, Arne Jonsson | News texts | PageRank, vector space technique | It is possible that PageRank works better for texts with more sentences as a larger number of sentences can be used to strengthen the mutual effect. |

##### References

1. Ramezani, Majid, and Mohammad-Reza Feizi-Derakhshi. "Automated text summarization: An overview." *Applied Artificial Intelligence* 28.2 (2014): 178-215.
2. Allahyari, Mehdi, et al. "Text summarization techniques: a brief survey." *arXiv preprint arXiv:1707.02268* (2017).
3. Ambedkar Kanapala,Srikanth Jannu,Rajendra Pamula. “Passage-Based Text Summarization for Legal Information Retrieval”. Arabian Journal for Science and Engineering, 2019,11
4. Yande, Shrutika S. "Text Summarization for Aspect-Polarity Extraction." INTERNATIONAL JOURNAL 2.11 (2017).
5. Gupta, Vishal, and Gurpreet Singh Lehal. "A survey of text summarization extractive techniques." *Journal of emerging technologies in web intelligence* 2.3 (2010): 258-268.
6. Perea-Ortega, José M., et al. "Application of text summarization techniques to the geographical information retrieval task." *Expert systems with applications* 40.8 (2013): 2966-2974.
7. Xiao, Ya, et al. "Joint entity and relation extraction with a hybrid transformer and reinforcement learning based model." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. No. 05. 2020.
8. Zhang, Xinyu, and Zhiqiang Ge. "Automatic deep extraction of robust dynamic features for industrial big data modeling and soft sensor application." *IEEE Transactions on Industrial Informatics* 16.7 (2019): 4456-4467.
9. Galassi, Andrea, Marco Lippi, and Paolo Torroni. "Attention in natural language processing." *IEEE Transactions on Neural Networks and Learning Systems* (2020).
10. Yuan Jianglin, G Zhingang, C Gang. “Recurrent convolution attention model (RCAM) for text generation based on title”. Journal of Physics, 2019
11. Kaur, Manpreet, and Dipti Srivastava. "Text Summarization using Partial Textual Entailment based Graphs." *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. IEEE, 2019.
12. Bahloul, Belahcene, Hassina Aliane, and Mohamed Benmohammed. "ArA\* summarizer: An Arabic text summarization system based on subtopic segmentation and using an A\* algorithm for reduction." *Expert Systems* 37.2 (2020): e12476.
13. Yuan, Chengzhe, et al. "Incorporating word attention with convolutional neural networks for abstractive summarization." *World Wide Web* 23.1 (2020): 267-287.
14. Krishnamurthi, Karthik, Vijayapal Reddy Panuganti, and Vishnu Vardhan Bulusu. "Understanding document semantics from summaries: a case study on Hindi texts." *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 16.1 (2016): 1-20.
15. Embarak, Ossama. "Data Gathering and Cleaning." *Data Analysis and Visualization Using Python*. Apress, Berkeley, CA, 2018. 205-241.
16. Melek, Zeki, et al. "Visualization of fibrous and thread-like data." *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006): 1165-1172.
17. Mishev, Kostadin, et al. "Evaluation of sentiment analysis in finance: from lexicons to transformers." *IEEE Access* 8 (2020): 131662-131682.
18. Gupta, Anushka, Diksha Chugh, and Rahul Katarya. "Automated News Summarization Using Transformers." *arXiv preprint arXiv:2108.01064* (2021).
19. Saggion, Horacio, and Thierry Poibeau. "Automatic text summarization: Past, present and future." *Multi-source, multilingual information extraction and summarization*. Springer, Berlin, Heidelberg, 2013. 3-21.
20. Khaing, Phyu Phyu, and Than Nwe Aung. "Real Time Semantic Events Detection from Social Media Stream." *International Conference on Big Data Analysis and Deep Learning Applications*. Springer, Singapore, 2018.
21. I. F. Moawad and M. Aref, "Semantic graph reduction approach for abstractive Text Summarization," 2012 Seventh International Conference on Computer Engineering & Systems (ICCES), 2012, pp. 132-138, doi: 10.1109/ICCES.2012.6408498.
22. Hovy, Eduard, and Chin-Yew Lin. "Automated text summarization in SUMMARIST." Advances in automatic text summarization 14 (1999): 81-94.
23. Vishal Tyagi, Utkarsh Dwivedi, Yash Kumar Sharma, Tarun Agarwal. "Automatic Text Summarization". IJIRT Volume 8, Issue 1.
24. Senthilraja, M. & Rajendran, Srinivasan & Iniyan, S. (2017). Document Summarization and Keyword Generation using Graph-Based Ranking Algorithm. International Journal of Control Theory and Applications. 10. 333.
25. Suleiman, Dima, and Arafat Awajan. "Deep learning based abstractive text summarization: Approaches, datasets, evaluation measures, and challenges." Mathematical Problems in Engineering 2020 (2020).
26. Al-Sharman, Nesreen. Automatic Text Summarizations. New Mexico State University, 2018.
27. Dehru, Virender, et al. "Text Summarization Techniques and Applications." IOP Conference Series: Materials Science and Engineering. Vol. 1099. No. 1. IOP Publishing, 2021.
28. Benkhouya, Amine, and Hemza Rahmani. "Abstractive Text Summarization."
29. Vasudevan, Srividhya. (2021). Automatic Text Summarization of Articles in Wikipedia.
30. Bhargava, Rupal, and Yashvardhan Sharma. "Deep extractive text summarization." Procedia Computer Science 167 (2020): 138-146.
31. Ni, Ansong, et al. "SummerTime: Text Summarization Toolkit for Non-experts." arXiv preprint arXiv:2108.12738 (2021).
32. HORASAN, Fahrettin, and Burhan BİLEN. "Extractive Text Summarization System for News Texts." International Journal of Applied Mathematics Electronics and Computers 8.4 (2020): 179-184.
33. Issam, Kalliath Abdul Rasheed, and Shivam Patel. "Topic Modeling Based Extractive Text Summarization." arXiv preprint arXiv:2106.15313 (2021).
34. Moradi, Milad, and Nasser Ghadiri. "Text Summarization in the Biomedical Domain." arXiv preprint arXiv:1908.02285 (2019).
35. Talukder, Md Ashraful Islam, et al. "Comparative Study on Abstractive Text Summarization." 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2020.
36. Cano, Erion, and Ondřej Bojar. "Keyphrase generation: A text summarization struggle." arXiv preprint arXiv:1904.00110 (2019).
37. Kanapala, Ambedkar, Sukomal Pal, and Rajendra Pamula. "Text summarization from legal documents: a survey." Artificial Intelligence Review 51.3 (2019): 371-402.
38. Sankarasubramaniam, Yogesh, Krishnan Ramanathan, and Subhankar Ghosh. "Text summarization using Wikipedia." Information Processing & Management 50.3 (2014): 443-461.
39. Janaszkiewicz, Piotr, and Przemysław Różewski. "The method of multidimensional approach to text summarization." Procedia Computer Science 159 (2019): 2189-2196.
40. Pontes, Elvys Linhares. Compressive Cross-Language Text Summarization. Diss. Université d'Avignon, 2018.
41. Algaphari, Ghaleb, Fadl M. Ba-Alwi, and Aimen Moharram. "Text summarization using centrality concept." International Journal of Computer Applications 79.1 (2013).
42. Zolotareva, Ekaterina, Tsegaye Misikir Tashu, and Tomás Horváth. "Abstractive Text Summarization using Transfer Learning." ITAT. 2020.
43. Dohare, Shibhansh, Harish Karnick, and Vivek Gupta. "Text summarization using abstract meaning representation." arXiv preprint arXiv:1706.01678 (2017).
44. Khare, Aparna, and Lalji Prasad. "TEXT SUMMARIZATION USING THEMANTIC HIERARCHY ALGORITHM." (2016).
45. Jafari, Mehdi, et al. "Automatic text summarization using fuzzy inference." 2016 22nd International Conference on Automation and Computing (ICAC). IEEE, 2016.
46. Pal, Alok Ranjan, and Diganta Saha. "An approach to automatic text summarization using WordNet." 2014 IEEE international advance computing conference (IACC). IEEE, 2014
47. Ajay, Abhinandh & Rajendran, Srinivasan & Shravan, Venkatesh. (2020). Query-based Text Summarization using Averaged Query. International Journal of Psychosocial Rehabilitation. 24. 2020.
48. Ozsoy, Makbule Gulcin, Ferda Nur Alpaslan, and Ilyas Cicekli. "Text summarization using latent semantic analysis." Journal of Information Science 37.4 (2011): 405-417.
49. Silber, H. Gregory, and Kathleen F. McCoy. "An efficient text summarizer using lexical chains." INLG’2000 Proceedings of the First International Conference on Natural Language Generation. 2000.
50. Gustavsson, Pär, and Arne Jönsson. "Text summarization using random indexing and pagerank." Proceedings of the third swedish language technology conference (sltc-2010), linköping, sweden. 2010.