News article Summarization

Bansi Mehta
Department of Computer Science
Dalhousie University
bansi.mehta@dal.ca

Deep Patel
Department of Computer Science
Dalhousie University
deep.patel@dal.ca

Janvi Patel
Department of Computer Science
Dalhousie University
janvi.patel@dal.ca

Sanket Shah
Department of Computer Science
Dalhousie University
sanket.shah@dal.ca

1 Abstract

In the recent time, there is a wide availability of information on the internet. It is caused by many factors, including rapid digitization of paper documents, rapid internet expansion based on Web 2.0. This has raised the question and necessity for alternative ways for displaying some selected textual and multimedia content in such a way that only the most important information is highlighted. This would help the users to make decision weather to read and further investigate the document or move forward with something else. This problem is far more adverse in journalism. Leading print media channel like BBC news alone generates around 200 new articles each day on weekdays. This number further goes up on weekends. Now the situation is such that rather than quantity, relevance of the content is a big concern. User need some tool that can summarize the information for them. In this paper the emphasis is on exploring the means to generate abstracting text summary with a major focus on models such as BART and PEGASUS. This project implements both the models and evaluates the result. It considers execution speed and precision to determine the performance. The end result is a tool that can take input from the user to generate a summarized version of the content.

2 Introduction

Nowadays there is a plethora of information available on a variety of topics on the internet. Due to excessive contribution by the people, this information does not necessarily serve the needs of a user. News articles are one of the major part of this information on the Internet. In today's busy life, people do not spend a lot of their time to read news articles. Often these articles tend to be a click-bait and unnecessarily lengthy which ultimately wastes user's time. Summarization is the task of compressing a piece of text to a shorter version, reducing the size of the original text while at the same time preserving key informational elements and the meaning of content [1]. Manual summarization is time expensive and laborious task, so the automatization of the task is gaining increasing popularity. There are various applications for text summarization in various NLP related tasks such as text classification, question answering, legal texts summarization, news summarization, and headline generation. However, the summary generation is the intermediate stage that helps to reduce the length of the document. It is very challenging, because when we as humans summarize a piece of text, we usually read it entirely to develop our understanding, and then write a summary highlighting its main points. Since computers lack human knowledge and language capability, it makes automatic text summarization a very difficult task. Reading a lot of text to get the gist of any news is also a problem which needs to be resolved in the effective and efficient manner. To provide meaningful and abstractive summary the models trained on dataset CNN Dailymail [21] [22] are being used. The implementation starts with data-preprocessing, model generation, summarization and evaluation. For evaluation, RougeL and RougeS score has been calculated focusing mainly on precision.

3 Problem statement

It is becoming important to get precise and concise news nowadays due to lack of time that is needed to go through the entire article. This generates the need for "Abstractive text summary of news articles" for reducing the time that is needed to read the article. Summarizing the news will provide readers to have succinct overview of interesting details and important information.

4 List of possible approaches

There are two broad concepts to text summarization [2]:

1. Extractive: From the original content, sentences which seems most relevant and important are identified and extracted to generate the summary.

2. Abstractive: In contrast to extractive approach, new sentences are generated from original content maintaining and relevance and meaning same as of the original content.

The task before summarization is the preprocessing of the original text, Following steps would be performed as a part of text preprocessing [2][3]:

1. Remove non-alphabet characters.
2. Remove punctuation.
3. Remove special characters.
4. Convert words to lower case.
5. Remove excessive white space.
6. Remove stop words.

The text generated after preprocessing will pass through summarizer to get summarized text.
There are many approaches that could be adopted to generate text summary:

4.1 Seq2seq

This approach is mostly used where the input data is sequential [2]. It is mostly used in areas of natural language translation (input is in one language and output is in another language), named entity recognition (input is text and output is tags). This Seq2Seq approach can also be used in text summarization where to original text is the input (long text) and output is the summarized version of it. This approach as two main components: encoder and decoder which are set in training and testing phase.
This method will not work if the input sentences are very long. Reason being it would be difficult for encoder to convert long sentences into fixed length vectors.

4.2 Attention Mechanism

Instead of focusing on entire sentence, attention mechanism only focuses on the parts of it which is then used to generate the output (summarized text) [4].
There are two types of approaches in attention mechanism: Local Attention, Global Attention
This method has few issues like: Semantic irrelevance, Grammatical error, and Loss relevance of main idea.

4.3 T5

T5 is the abbreviation for "Text-to-Text Transfer Transformer" [5]. It uses sequence-to-sequence generation method that includes cross-attention layers to the decoder and generates the decoder output autoregressive [5]. One of the approaches mentioned in the research papers [3], is to provide the input to encoder as a series of tokens which are in sequence of embeddings. A self attention layer and feed forward network are the two subcomponents. Same as encoder, decoder follows similar structure with a modification of generalized attention mechanism after every self attention layer. The final layer is a dense layer and uses softmax as activation function [3].

4.4 GPT

GPT-2 models have over 1.5 billion parameters with a large transformer base [6]. The model is used to perform several tasks such as predicting the next word and modeling the language. Basically, the predicting task is used to predict the next choice as a gold state summary and using the modeling task can predict and model the next word [7]. In the process each task has their individual loss. The NEWS summarization tasks can be tuned with the model by passing the token (—summarize—) as a parameter. The token is used to differentiate between golden summary and tokens. Apart from that, all the tokens are padded with 1024 tokens and also inputs more than 1024 are truncated. As GPT-2 is an auto regressive model the nth regressive output is generated as a summarized text based on all n-1 inputs [7].

4.5 BERT

BERT stands for Bidirectional Encoder Representations from Transformers. BERT uses a set of transformer encoders and provides the text and sentence level summary [6]. Those granular levels of text can be generated with the help of unsupervised learning techniques such as masked language modeling and next sentence prediction technique. BERT is trained with a huge dataset of 3300 M [5].Bert includes two separate mechanisms – an encoder that takes text as input and a decoder that produces a prediction for the task. It is implemented as a sequence-to-sequence model with a bidirectional encoder over corrupted text and a left-to-right autoregressive decoder [8]. It is a pre-trained model that is being used as a transformer encoder to provide a sentence level understanding. To provide the sentence and word level understanding a contextual relationship BERT uses Transformer Encoder, but as per the need of making an abstract summary the transformer decoder is also being used to provide some properly structured and meaningful output [5].

4.6 BART

BART uses two different architectures of two different models. It uses a bidirectional encoder of BERT and left to right decoder architecture of GPT with a proper seq2seq to generate proper summarization [6]. BART is a diagnosing autoencoder that works with data and parallel maps the original data (reference summary) to the corrupted data (improper system summary). The bidirectional encoder of BERT is used upon corrupted text and a left to right decoder is also implemented with a standard seq2seq transformer architecture [9]. BART provides very generous results when it is fine tuned for text generation for prediction tasks but the model also provided promising results while used for comprehension and summarization tasks as well. By matching and comparing the predicted results it can generate the state of the art results for summarization tasks [9]. BART provides an innovative approach of fine tuning with additional layers of transformers and machine translations. BART uses 6 layered encode-decoder architecture for its base model and 12 layered architecture for larger models [9].

4.7 PEGASUS

Working algorithm of PEGASUS model does not depend on the randomly selection of sentences. Mainly, it removes significant lines from the provided input text, and this will be treated as separate output [4]. The popularity of the PEGASUS model is because of its working algorithm as it only chooses the relevant sentences by eliminating rest of the part. To train a transformer model, PEGASUS model uses its self-supervised objective GSG [10]. The whole explained process improves model's fine-tuning performance significantly on automatic text summarization. As the name of the PEGASUS stands for Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-Sequence [4], this is pre-trained Sequence-to-Sequence model which can be used to train transformer model for interpretation of the entire digital document or article.

From above explained models, it can be said that the abstractive summarization models work on machine learning and deep learning models which can generate the summary in a meaningful way. However, this summary's context may defer from the actual context or article given but this model tries to cover the main aspect or the dialogues as the gist of the article. For example, any news article related to several questions and answers can be further summarized by the abstractive summarization which may just contain some dialogue content of the article and may ignore the core approach of the entire article [11]. The BART model is proposed in this paper which is the abstractive summarization model and may generate the summary based on its own interpretation to make it more suitable and reflect the idea of the article as a whole.

5 Problem Definition and Methodology

5.1 Dataset

Dataset on which models (BART and PEGASUS) were trained and fine-tuned:

- google/pegasus-cnn-dailymail (Pegasus) [12][21].
- facebook/bart-large-cnn (Bart) [13][22].

The dataset has following characteristics:

- It has an information rich column called "Text" which contains the actual news.
- It has a column called 'Highlight" which contains the main highlight of the news. This column can be used as a summary reference.
- It has roughly around 300K articles (31% from CNN and 69% from daily mail).
- Length of the article is around 766 words [14].

Data that has been used for input in the tool:

- BBC News [23].

5.2 Model implemented

5.2.1 BART

Bidirectional and auto-regressive transformer (BART) is a de-noising auto encoder for the pre-training of sequence-to-sequence models for natural language generation, translation and comprehension [15]. The model is trained by altering the text with an arbitrary noising function, and to rebuild the original text [12]. BART uses the transformer based neural machine translation architecture with a bidirectional- BERT like encoder and GPT like decoder [12][17].

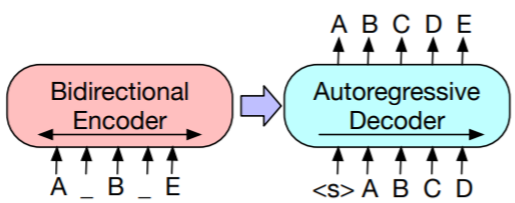

Figure 1: BART Architecture

If the single directional sequence performance is being used then the degradation can be possible as the complete context of one token contains preceding and following tokens, thus feeding only come before in words decoded to the decoder which can be turned into the unnatural sequences [16].

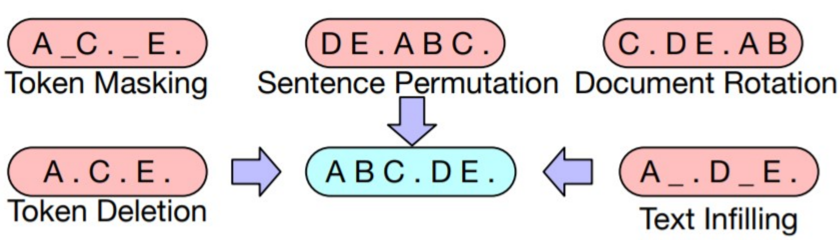

Figure 2: Multi-spectral Image

The model uses a 6 layered encoder-decoder. The pre-training techniques compared in the experiments can be divided between those that work at the token level and those that work at the sentence level [17][9]:

- **Token Masking** random tokens are sampled and replaced with [MASK]

- **Token Deletion** is the same as masking but the sampled token is deleted and the model will add a new token at those places.
- **Token Infilling** a number of text spans, i.e. contiguous group tokens, are sampled, and then they are replaced by the [MASK] token.
- **Sentence Permutation** random shuffling of the sentences from text.
- **Document Rotation** a token is chosen randomly to be the start of the document, the section before the starting token is appended at the end.

5.2.2 PEGASUS

Pre-training with Extracted Gap-Sentences for Abstractive Summarization is used for Gap Sentence Generation. For the encoding module, the random mask words are generated and use the other words from the sequence to predict the masked words [12][18].

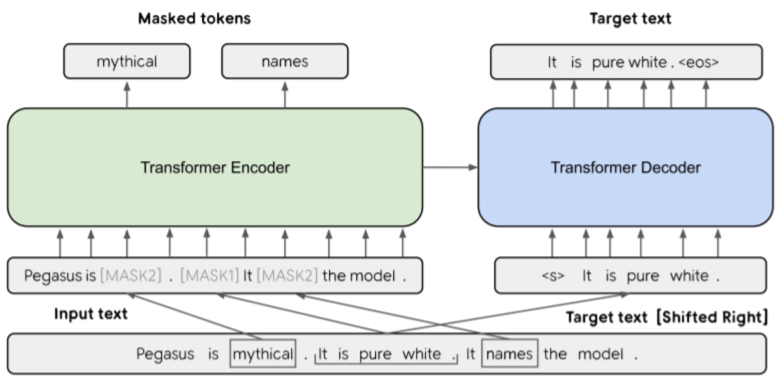

Figure 3: GAP

For PEGASUS, encoder (MLM) and decoder (GSG) train parallely. For example, from 3 sentences one sentence is masked with [MASK1] and used as target generation text (GSG). The other two sentences remain in the input, but some words are randomly masked by [MASK2] (MLM) as shown in below figure [12].

5.3 Implementation

The implementation is divided into three main sections:

1. Data preprocessing.
2. Model generation.
3. Defining summary length.
4. Model tuning paramters
5. Summarization

5.3.1 Data Preprocessing

This phase includes tasks performed on the data to clean it. In order to achieve good results with the model and an understandable summary it is essential to clean the data before hand.

Data preprocessing would include the following steps:

- Remove Non-Ascii characters.
  - Input: ”àA string asfec withé fuünny charactersß.”
  - Output: ”A string asfec with funny characters.”

- Convert everything to lower case.
  - Input: ”A string asfec with funny characters.”
  - Output: ”a string asfec with funny characters.”
- Removing all non-english words.
  - Input: ”a string asfec with funny characters.”
  - Output: ”a string with funny characters.”
- Remove all special characters.
  - Input: ”a string with funny characters.”
  - Output: ”a string with funny characters.”
- Remove excessive spaces.
  - Input: ”a string with funny characters.”
  - Output: ”a string with funny characters.”

There are two more steps in data cleaning which is not required when using BART or PEGASUS.

- Remove stopwords.
- Tokenize the input.

5.3.2 Model generation

This phase includes generating models for Bart and Pegasus. It uses a pre-trained data set mentioned above to generate the model. This phase also includes generation of a tokenizer.

5.3.3 Defining summary length

After the model generation step, next is defining the length of summarized output. This step calculates the minimum length and maximum length of the summarized text based on the initial document length.
If, d is the document length than,

- Minimum length = d / 6
- Maximum length = Minimum length + 200

Currently the system has two hard coded values 6 and 200. These values can be controlled by the end user to define the length of the summary.

5.3.4 Model tuning parameters

Table 1: Model tuning parameters:

TUNING PARAMETERS	BART	PEGASUS
Num of beams	4	6
Num of return sequences	1	1
Num of repeat ngram size	2	2
Length penalty	1	1
Minimum length	12	30
Maximum length	128	128
Early stopping	True	True

Length penalty is a method that prevents the probability value from decreasing when a long sentence is translated. Early stopping is used to monitor the generalization error when the model starts degrading. [19]

Early stopping is used to monitor the generalization error when the model starts degrading. [19]

5.3.5 Summarization

Using all the following values derived till now, the system summarizes the text:

- Model.
- Tokenizer.
- Mimimum length.
- Maximum length.

As mentioned in 4.2.1 and 4.2.2, using the tokenizer first encoder is generated which encodes the input string. Using the BART/ PEGASUS model and the encoded string, a model is generated. The output returned from the model is decoded and the final summarized text is returned.

5.4 Evaluation

Rouge Score is the acronym for Recall Oriented Understudy for Gisting Evaluation. Rough score is a combined value in terms of a set of matrices to evaluate the system generated summary. It uses the original text as a gold standard data or reference summary to compare and generate the outcome as a rouge score. Rouge score is computed through various parameters such as precision, recall and F1-Measure [20].

5.4.1 Precision Recall

Precision and recall in terms of rouge score can be calculated in a slightly different way as represented below:

Precision can be used to evaluate the amount of summary that is actually useful based on the system generated summary. The higher score in precision can suggest that the summary is accurate and can be precise in terms of relevance. The formula for precision in the context of Rouge Score is represented in given formula.

$$Precision = \frac{Number \quad of \quad Overlapping \quad words}{Total \quad words \quad in \quad system \quad summary}$$

Recall provides the information on the generated summary that actually contains the portions of the reference summary. Over here each individual word is precisely captured to check if they are overlapping for both reference and system summaries based on the reference summary. The formula to calculate the recall in context of Rogue score is represented in given formula.

$$Recall = \frac{Number \quad of \quad Overlapping \quad words}{Total \quad words \quad in \quad reference \quad summary}$$

Based on the precision and recall F1 measure can be calculated and can be used to present as a measurement for the generated summary evaluation.

5.4.2 Rouge Score Types

In the summarization of NEWS articles, the crucial information can be represented as long sentences and paragraphs. There can also be various gapping words in reference summary comparatively with system generated summary. Such measurements must be neglected to evaluate the generated summary to get the proper results in terms of Rouge score matrices. To avoid those inaccuracies in calculation here 3 types of Rouge scores are being calculated [20].

Rouge-N: There are various unigrams, bigrams and higher N-grams which can be used to evaluate further with more proper results. Here Rouge-1 is used to evaluate the generated summary in terms of overlapping unigrams and produce the results by comparing the granular level words.

Rouge-L: The get the results based on the sentence level evaluation the Rouge-L can be calculated which uses Longest Matching Subsequence (LMS). The LMS has its predefined n-grams therefore the parameters are not required to be set for n-grams. LMS matches consecutive sentences based on word by word from the reference summary and therefore through Rouge-L the matching sequence for the sentence can be helpful to determine whether the meaning of the sentence remains the same in the generated summary based on the sentence structure [20].

5.5 Experiment and Result

5.5.1 Execution Time

Same input was taken to generate the summary using BART and PEGASUS. Keeping the length of summary approximately same, this experiment was executed with an objective to find out the faster among the two implementation. Summary length were considered 1/8th, 1/6th and 1/4th of the original text and execution time was noted as shown in Table 2. Plotting a graph for the above representation, it can be noted that BART is way faster than PEGASUS in all the three cases.

Table 2: Summary Length Vs. Execution Time (Minutes)

Summary Length	BART	PEGASUS
1/8th	2.138337	8.695859
1/6th	2.051748	6.693776
1/4th	1.900121	6.583237

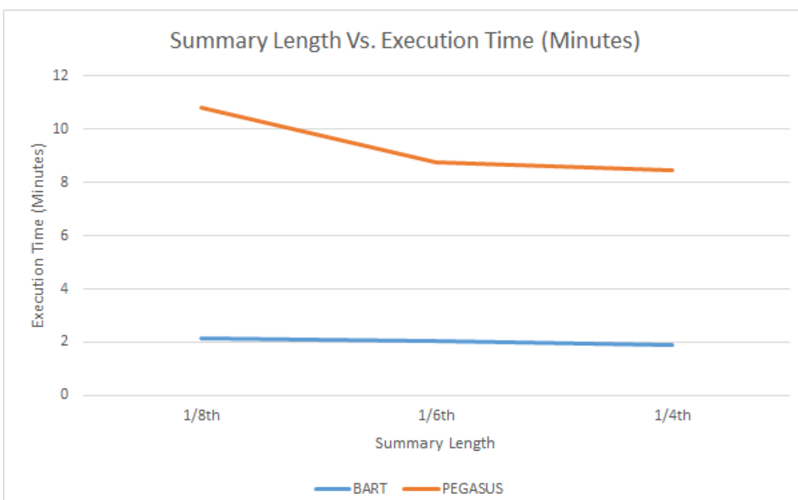

Figure 4: Execution Time

5.5.2 Rouge Score

When same input taken to generated the summary of varying lengths using BART and PEGASUS we can note the difference in Rouge1 score for both the methods as shown in Table 3. It can be noted that as summary length increases, (1/4th of original is longer than 1/8th of original) precision value decreases. The same can be noted for RougeL score (from Table 4). But coming to a general conclusion, BART has a great overall precision in most of the cases.

Table 3: Summary Length Vs. Rouge1 (Precision)

Summary Length	BART	PEGASUS
1/8th	1	1
1/6th	0.98	1
1/4th	0.97	0.93

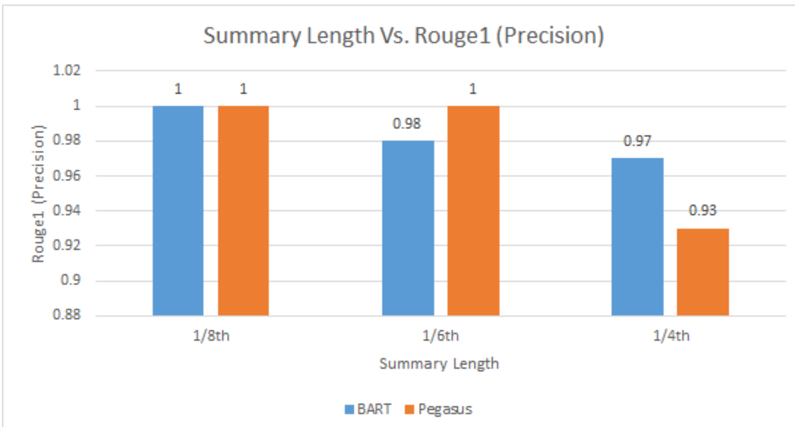

Figure 5: Rouge-1 score

Table 4: Summary Length Vs. RougeL (Precision)

Summary Length	BART	PEGASUS
1/8th	1	0.95
1/6th	0.95	0.95
1/4th	0.82	0.78

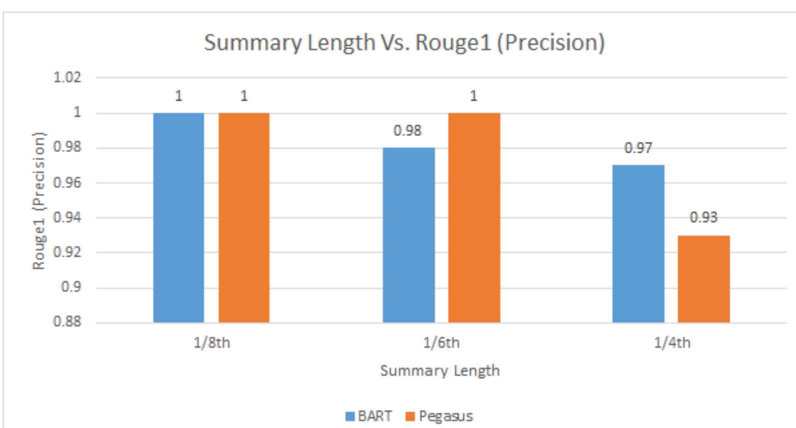

Figure 6: Rouge-L score

6 Our System

The tool that we have created provides user with three options to get the text summarized.

1. URL: URL provides the user with an option to enter the URL of the text that they want to summarize. The system fetches the HTML content of that URL and extract all the content between <P> </ P>. This content is used to generate the summary.
2. Text: Text provides the user with an option to directly copy paste the text which the system summarizes and returns the result.
3. Browse: Browse provides the user with an option to Browse and upload text file. The system reads the file and uses the file content to generate summary.

As shown in Figure 7, if the user uses ”Text” option to generate the summary the result would look as shown in the Figure. User copies the text and clicks on ”Summarize”.The system will summarize it using BART and PEGASUS and return back the output which is printed on the screen.

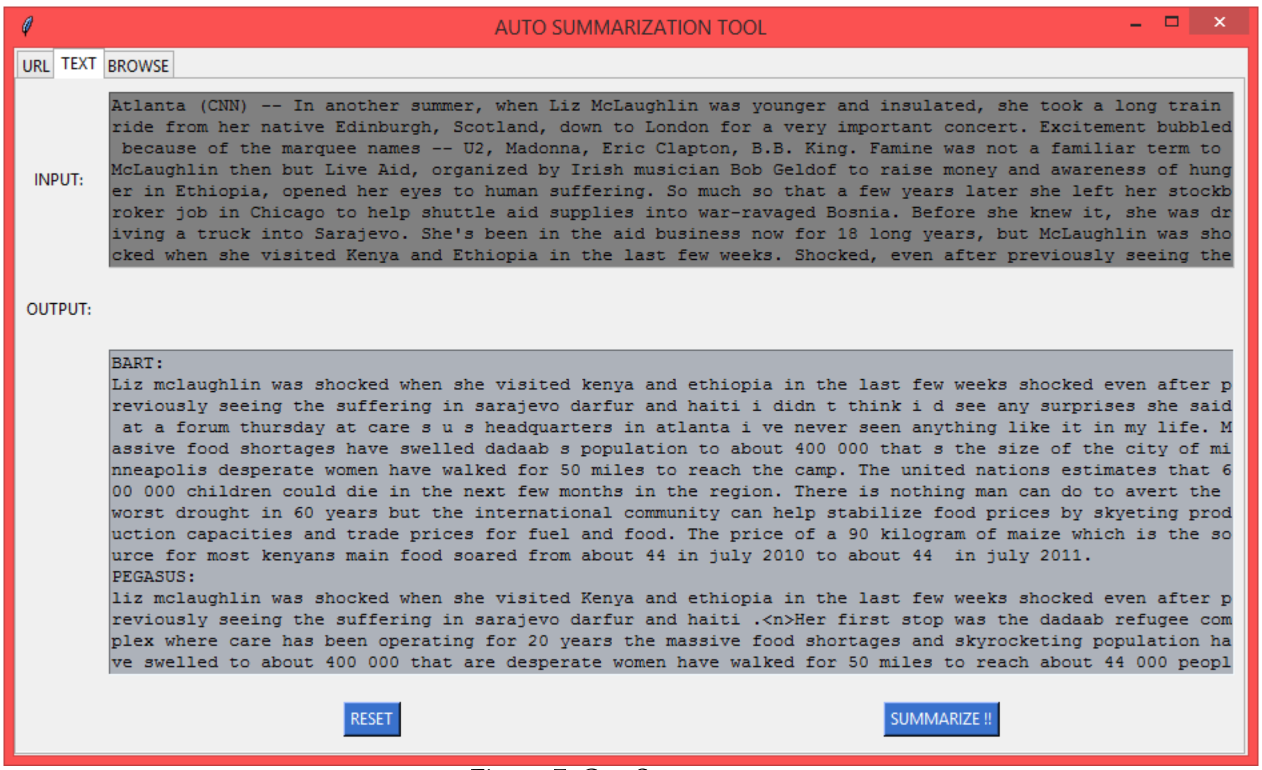

Figure 7: Our System

7 Conclusion

There are plenty of models and algorithms in the field of automatic text summarizaiton. After research, we found out that BART and PEGASUS are the most used models right now. This project was aimed at understanding the model architecture, implementing them to create a tool that would provide summarized text of the input text, understanding the evaluation techniques and using evaluation techniques to compare the performance of the two models - BART and PEGASUS.

When comparing BART and PEGASUS through the means of rouge score and execution time, it was observed that BART out-performed PEGASUS in almost all the cases. Execution time of BART of was almost always faster than PEGASUS when the summary length was configured to be 1/8th, 1/6th and 1/4th of the original text. Along with it, when comparing the rouge score (Rouge1 and RougeL) for the value of precision, BART had better value than PEGASUS.

Through this we came to conclusion that although, BART and PEGASUS both follow transformer encode and decoder model, BART is specifically designed for text summarization at it's architecture level and thus, provides better performance.

References

[1] Brownlee, "A Gentle Introduction to Text Summarization", Machine Learning Mastery, 2021. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-text-summarization/[Accessed: 04- Nov- 2021].

[2] Arvind Pai; "Text Summarization — Text Summarization Using Deep Learning", Analytics Vidhya, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/[Accessed: 04- Nov- 2021].

[3] Anushka Gupta1 , Diksha Chugh2 , Anjum3 , Rahul Katarya4; "Automated News Summarization Using Transformers", 2021. [Online]. Available: https://www.researchgate.net/publication/353653704_Automated_News_Summar ization_Using_Transformers [Accessed: 04- Nov- 2021].

[4] Luís Gonçalves; "Automatic Text Summarization with Machine Learning—An overview", Medium, 2021. [Online]. Available: https://medium.com/luisfredgs/automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25 [Accessed: 04- Nov- 2021].

[5] M. Ramina, N. Darnay, C. Ludbe and A. Dhruv, "Topic level summary generation using BERT induced Abstractive Summarization Model," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 747-752, doi: https://ieeexplore.ieee.org/document/9120997

[6] "Summarize Reddit Comments using T5, BART, GPT-2, XLNet Models", Medium, 2021. [Online]. Available: https://towardsdatascience.com/summarize-reddit-comments-using-t5-bart-gpt-2-xlnet-models-a3e78a5ab944. [Accessed: 04- Nov- 2021].

[7] V. Kieuvongngam, B. Tan, en Y. Niu, "Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2", arXiv [cs.CL]. 2020. [Accessed: 04- Nov- 2021]

[8] ZhengzhiLou,JuZhang;"AbstractiveSummarizationonCOVID-19Pub-lications-PDFFreeDownload",Docplayer.net,2021.[Online].Available:https://docplayer.net/201739488-Abstractive-summarization-on-covid-19-publications.html. [Accessed: 04- Nov- 2021]

[9] Research.fb.com, 2021. [Online]. Available: https://research.fb.com/wp-content/uploads/2020/06/BART-Denoising-Sequence-to-Sequence-Pre-training-for-Natural-Language-Generation-Translation-and-Comprehension.pdf. [Accessed: 05- Nov- 2021].

[10] J. Zhang, Y. Zhao, M. Saleh, en P. J. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization", arXiv [cs.CL]. 2019.

[11] Y. Chen and Q. Song, "News Text Summarization Method based on BART-TextRank Model," 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2021, pp. 2005-2010, doi: 10.1109/IAEAC50856.2021.9390683.

[12] Medium. 2021. Automatic text summarization system using Transformers—Are you tired of reading a long paper?. [online] Available at: https://neurondai.medium.com/automatic-text-summarization-system-using-transformers-are-you-tired-of-reading-a-long-paper-22d2cd9f5260 [Accessed 13 December 2021].

[13] "Hugging Face" [online] Available at: https://huggingface.co/google/pegasus-cnn$_dailymail$ [Accessed 13 December 2021].

[14] "Hugging Face." [online] Available at: https://huggingface.co/facebook/bart-large-cnn [Accessed 13 December 2021].

[15] J. Xu, "Abstractive Summarization on COVID-19 Publications with BART," [online]. Available at: http://cs230.stanford.edu/projects$_spring_2020/reports/38866168.pdf$ [Accessed 13 December 2021].

[16] Aclanthology.org. 2021. [online] Available at: https://aclanthology.org/K19-1074.pdf [Accessed 13 December 2021].

[17] Medium. 2021. "BART: Are all pretraining techniques created equal?." [online] Available at: https://medium.com/dair-ai/bart-are-all-pretraining-techniques-created-equal-e869a490042e [Accessed 13 December 2021].

[18] Arxiv.org. 2021. [online] Available at: https://arxiv.org/pdf/1912.08777.pdf [Accessed 13 December 2021].

[19] Brownlee, J., 2021. A Gentle Introduction to Early Stopping to Avoid Overtraining Neural Networks. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/: :text=Earlytoppingisdesignedto,generalizationerrorbeginstodegrade [Accessed 13 December 2021].

[20] "Ccs.neu.edu, 2021." [Online]. Available: https://www.ccs.neu.edu/home/vip/teach/DMcourse/5-topicmodel-summ/notes-slides/What-is-Rouge [Accessed: 13- Dec- 2021].

[21] "google/pegasus-cnn-dailymail · Hugging Face" Huggingface.co, 2021. [Online]. Available: https://huggingface.co/google/pegasus$_cnn_dailymail$ [Accessed: 13- Dec- 2021].

[22] "facebook/bart-large-cnn · Hugging Face", Huggingface.co, 2021. [Online]. Available: https://huggingface.co/facebook/bart-large-cnn. [Accessed: 13- Dec- 2021]

[23] "BBC News Classification — Kaggle", Kaggle.com, 2021. [Online]. Available: https://www.kaggle.com/c/learn-ai-bbc/data [Accessed: 13- Dec- 2021]