NEWS ARTICLE SUMMARIZATION USING NLP

**K. Sudeeptha1, G. Sathvika2, A. Tejeswi3**

1Department of Computer Science and Engineering, SR University, Warangal, Telangana, India. 2Department of Computer Science and Engineering, SR University, Warangal, Telangana, India. 3Department of Electrical and Electronics Engineering SR University, Warangal, Telangana, India.

**Abstract:** In this new era, where tremendous information is available on the Internet, it is most important to provide the improved mechanism to extract the information quickly and most efficiently. It is very difficult for human beings to manually extract the summary of a large documents of text. There are plenty of text material available on the Internet. So there is a problem of searching for relevant documents from the number of documents available, and absorbing relevant information from it. In order to solve the above two problems, the automatic text summarization is very much necessary. Text summarization is the process of identifying the most important meaningful information in a document or set of related documents and compressing them into a shorter version preserving its overall meanings. We have developed using BART.

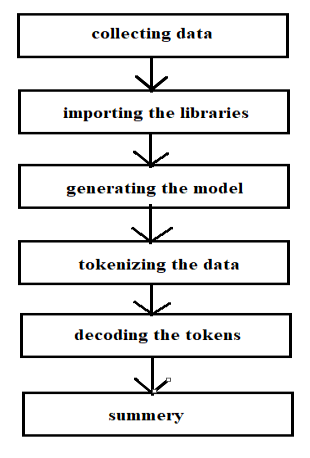
**Keywords: BART.**

# INTRODUCTION

Artificial Intelligence (AI) is the part of computer science that focuses on designing intelligent computer systems that show the traits we re-late with human intelligence like comprehending languages, learning problem-solving, decision making, etc. One of the significant contributions of AI has remained in Natural Language Processing (NLP), which glued together linguistic and computational techniques to assist computers in understanding human languages and facilitating human-computer interaction. Machine Translation, Chat bots or Conversational Agents, Speech Recognition, Sentiment Analysis, Text summarization, etc., fall under the active research areas in the domain of NLP. However, in the past few years, Sentiment analysis has become a demanding realm. Nowadays, Artificial Intelligence has spread its wings into Thinking Artificial Intelligence and Feeling Artificial Intelligence (Huang and Rust 2021). Figure 1 shows the sub domain of artificial intelligence. Thinking AI is designed to process information in order to arrive at new conclusions or decisions. The data are usually unstructured. Text mining, speech recognition, and face detection are all examples of how thinking AI can identify patterns and regularities in data. Machine learning and deep learning are some of the recent approaches to how thinking AI processes data. AI has made a big impact on the globe. AI was reintroduced in a significant manner in the twentieth century, and it inspired researchers to perform in-depth studies in domains like NLP, and machine learning. However, the domains of NLP remain ambiguous due to its computational methodologies, which assist computers in understanding and producing human-computer interactions in the form of text and voice. Text summarization refers to the technique of shortening long pieces of text. The intention is to create a coherent and fluent summary having only the main points outlined in the document. Automatic text summarization is a common problem in machine learning and natural language processing (NLP). Text Summarization is one of those applications of Natural Language Processing which is bound to have a huge impact on our lives. It is a process of generating a concise and meaningful summary of text from multiple text resources such as books, articles etc. applying text summarization reduces reading time, accelerates the process of researching for information, and increases the amount of information that can fit in an area. Text summarization is an interesting machine learning field that is increasingly gaining traction. As research in this area continues, we can expect to see breakthroughs that will assist in fluently and accurately shortening long text documents. Extractive text summarization involves the selection of phrases and sentences from the source document to make up the new summary. Techniques involve ranking the relevance of phrases in order to choose only those most relevant to the meaning of the source. Abstractive text summarization involves generating entirely new phrases and sentences to capture the meaning of the source document. This is a more challenging approach, but is also the approach ultimately used by humans. Classical methods operate by selecting and compressing content from the source document.

# PROBLEM DEFINITION:

Summarization helps in understanding a huge information within less time and helps in saving time. The summarization involves few steps mentioned below.



**Figure 2.** Steps for Text Summarization using NLP.

# DATASET AND ATTRIBUTES:

The dataset contains

1. Article
2. Summary
3. Id

The data is of three types

1. Training Data
2. Testing Data
3. Validation Data

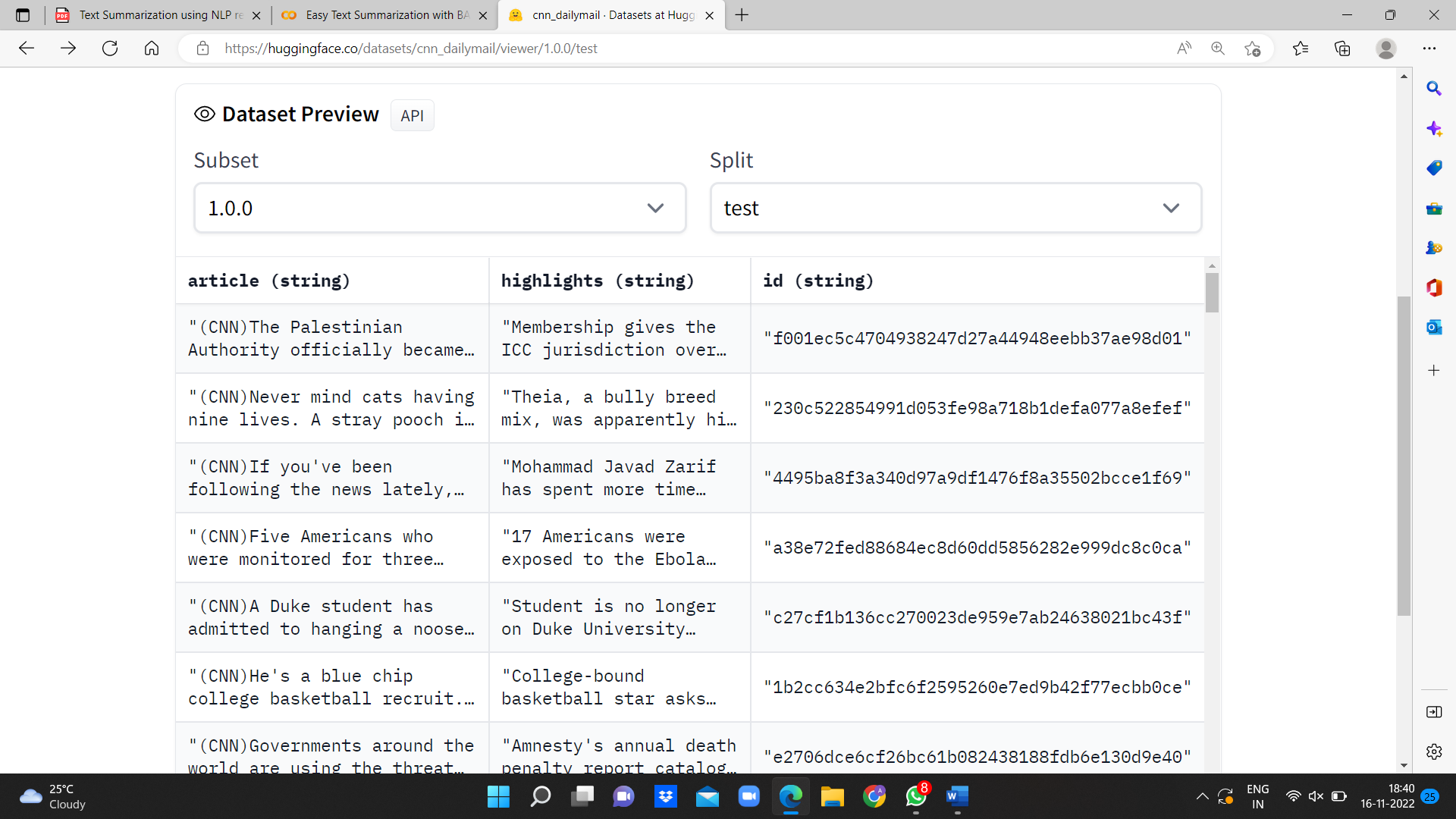


Fig.2. Test data

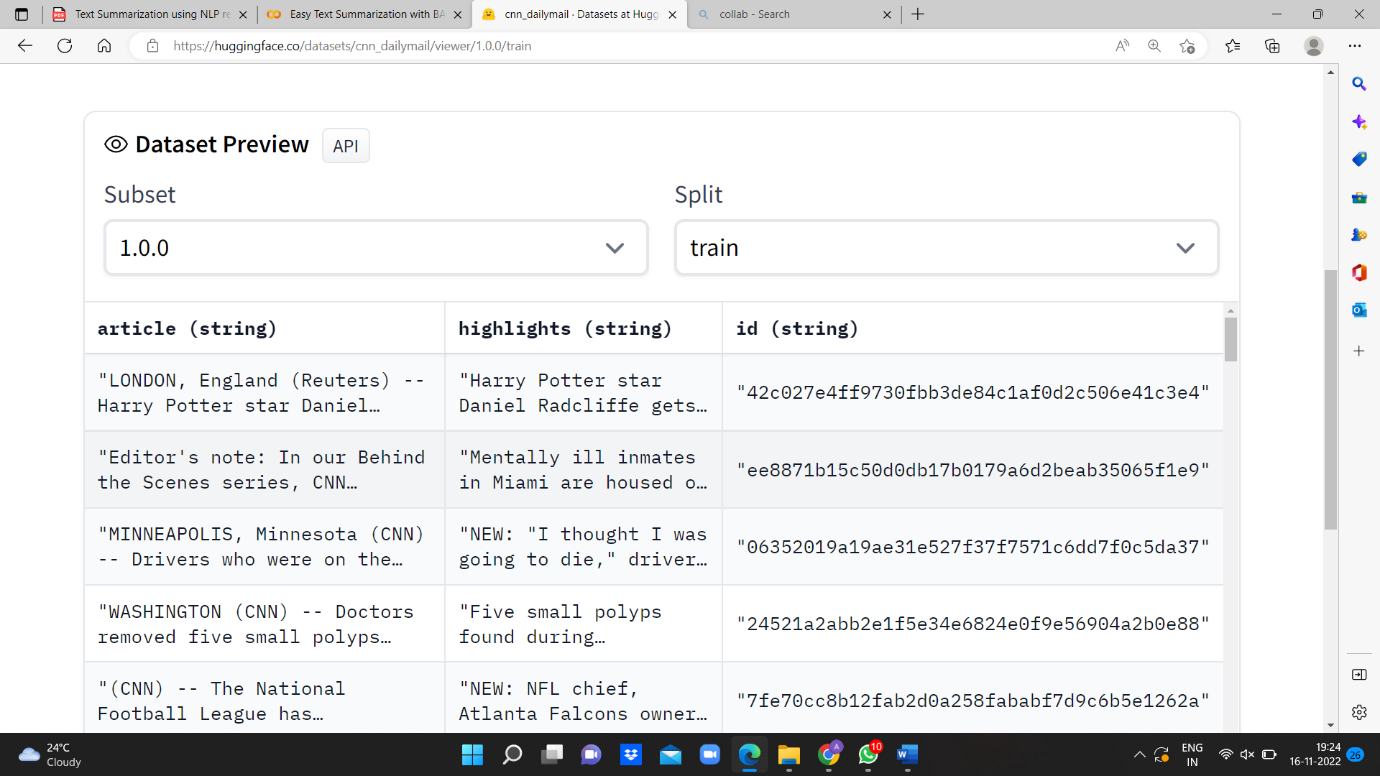


Fig.3. Training Data

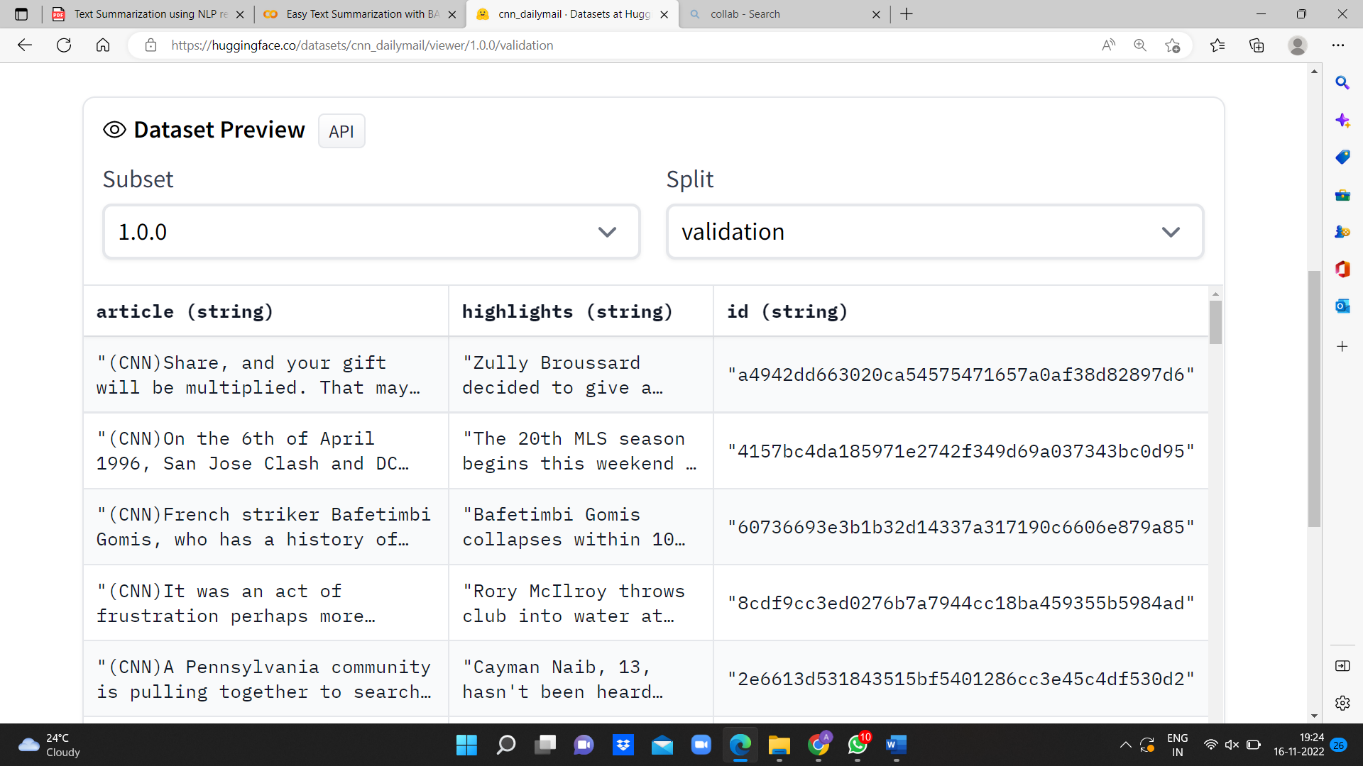


Fig.4. Validation Data

**Fig 3 :** Data-set.

### DATA PRE-PROCESSING :

* + - Real-world data collection has its own set of problems. It is often very messy which includes missing data, presence of outliers, unstructured manner, etc. Before looking for any insights from the data, we have to first perform preprocessing tasks which then only allow us to use that data for further observation and train our machine learning model. We use missing values treatment, outliers detection, normalization and data split to process our data before feeding it to the machine learning model.

***Data info:***



**Fig 4 :** Data preprocessing

## Missing values treatment:

The real world’s dataset often has many missing values which can be treated by using certain methods. But in our data, there are no missing values, because we collected the data manually through google forms survey and made sure not to miss out any data. To treat the missing values, we generally use the following strategies:

* Remove the entire row (If missing values are less in number)
* Replace the missing value with either mean or median
* Replace the missing value with most frequent value in the column (This is generally used only for large dataset)

## Normalization:

Normalization is a technique for organizing data in a database. Data normalization is the process of rescaling one or more attributes to the range of 0 to 1. This means that the largest value for each attribute is 1 and the smallest value is 0. It is important that a database is normalized to ensure only related data is stored in each table and to avoid biasing towards huge values. When we normalize the data while feeding it to the model, we also have to de-normalize it. This process can be done using the formulas below:

* = −

−

* = − +

## Data split:

To train any machine learning model irrespective what type of dataset is being used, we have to split the dataset into training and testing data. The reason to split the data is to give the machine learning model an effective mapping of input to outputs and to evaluate the model performance. We pass the training data to train our machine learning model and then test the model on testing data.We can do the data split using train\_test\_split module in python.

1. **ALGORITHMS:**

This section talks about the algorithms used for the project. We included algorithms like BART and BERT

### BERT:

Summarization refers to extracting (summarizing) out the relevant information from a large document while retaining the most important information. BERT (Bidirectional Encoder Representations from Transformers) introduces rather advanced approach to perform NLP tasks. BERT (Bidirectional transformer) is a transformer used to overcome the limitations of RNN and other neural networks as Long term dependencies. It is a pre-trained model that is naturally bidirectional. This pre-trained model can be tuned to easily to perform the NLP tasks as specified, Summarization in our case. Being trained as a masked model the output vectors are tokened instead of sentences. Unlike other extractive summarizers it makes use of embeddings for indicating different sentences and it has only two labels namely sentence A and sentence B rather than multiple sentences. These embeddings are modified accordingly to generate required summaries. refers to the representation of words in their vector forms. It helps to make their usage flexible. Embedding even the Google utilizes the this feature of BERT for better understanding of queries. It helps in unlocking various functionality towards the semantics from understanding the intent of the document to developing a similarity model between the words.

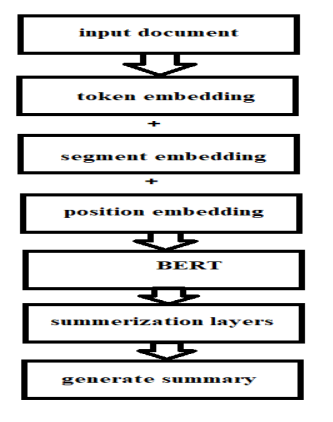


Fig.4. BERT flow chart

There are three types of embeddings applied to our text prior to feeding it to the BERT layer, namely: • Token Embeddings –

Words are converted into a fixed dimension vector. [CLS] and [SEP] is added at the beginning and end of sentences respectively.

• Segment Embeddings –

It is used to distinguish or we can say classify the different inputs using binary coding.

• Position Embeddings –

BERT can support input sequences of 512. Thus the resulting vector dimensions will be (512,768). Positional embedding is used because the position of a word in a sentence may alter the contextual meaning of the sentence and thus should not have same representation as vectors.

There are following two bert models introduced:

1. BERT base

In the BERT base model we have 12 transformer layers along with 12 attention layers and 110 million parameters.

2. BERT Large

In BERT large model we have 24 transformer layers along with 16 attention layers and 340 million parameters.

Transformer layer- Transformer layer is actually a combination of complete set of encoder and decoder layers and the intermediate connections. Each encoder includes Attention layers along with a RNN. Decoder also has the same architecture but it includes another attention layer in between them as does the seq2seq model. It helps to concentrate on important words.

### BART:

BART is a sequence-to-sequence model trained as a denoising autoencoder. This means that a fine-tuned BART model can take a text sequence (for example, English) as input and produce a different text sequence at the output (for example, French). This type of model is relevant for machine translation (translating text from one language to another), question-answering (producing answers for a given question on a specific corpus), text summarization (giving a summary of or paraphrasing a long text document), or sequence classification (categorizing input text sentences or tokens). Another task is sentence entailment which, given two or more sentences, evaluates whether the sentences are logical extensions or are logically related to a given statement. Since the unsupervised pretraining of BART results in a language model, we can fine-tune this language model to a specific task in NLP. Because the model has already been pre-trained, fine-tuning does not need massive labelled datasets (relative to what one would need for training from scratch). The BART model can be fine-tuned to domain-specific datasets to develop applications such as medical conversational chatbots, converting natural text to programming code or SQL queries, context-specific language translation apps, or a tool to paraphrase research papers. BART was trained as a denoising autoencoder, so the training data includes “corrupted” or “noisy” text, which would be mapped to clean or original text. BART is constructed from a bi-directional encoder like in BERT and an autoregressive decoder like GPT. BERT has around 110M parameters while GPT has 117M, such trainable weights. BART being a sequenced version of the two, fittingly has nearly 140M parameters. Many parameters are justified by the supreme performance it yields on several tasks compared to fine-tuned BERT or its variations like RoBERTa, which has 125M parameters in its base model.

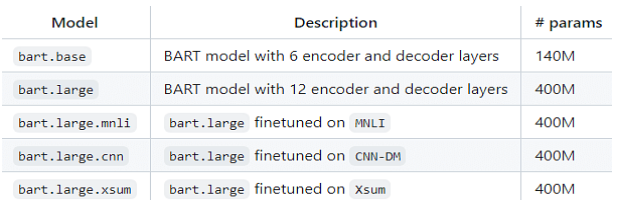


Fig.5.Parameters for different BART model

# RESULTS:

We used BART pretrained model for text summarization. This model has more efficiency when compared to the other existing models.

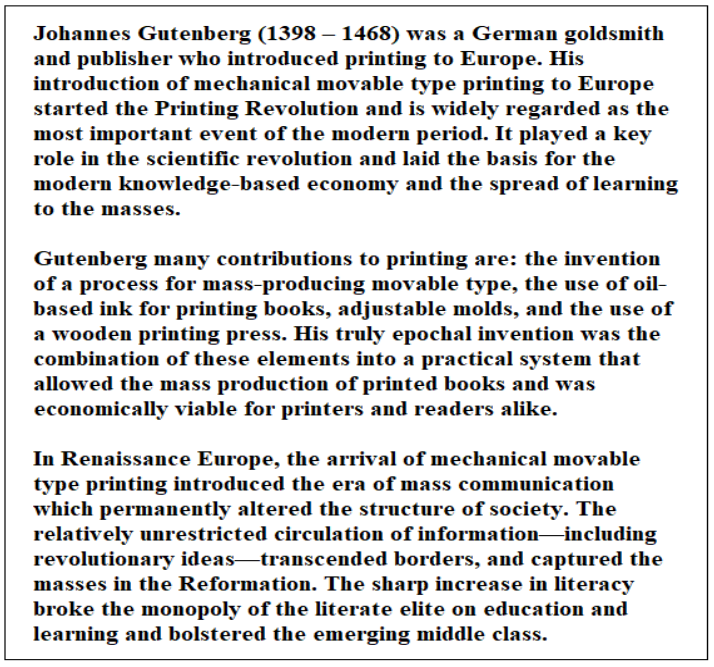


Fig.6.Text to be summarized

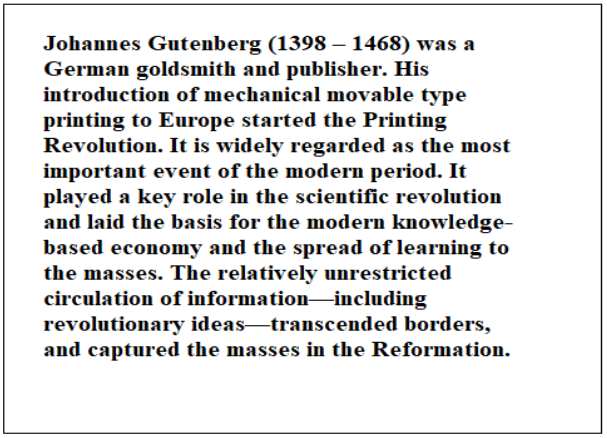


Fig.7.Summarized text

# CONCLUSION:

# From this we can conclude that the process of text summarization can be made simpler with the help of the Artificial Intelligence and the Natural Language Processing models. We did this by using the BART models. This can also be developed by using various methods like Seq2Seq, LSTM, and any other upcoming new models. We can also make this language specific by training the model in the required way and getting the summery only for the specific language texts.

# REFERENCES

[1] Ramesh Nallapati, Bowen Zhou et al., "Abstractive Text Summarization Using Sequence-to- Sequence RNNs and Beyond", The SIGNLL Conference on Computational Natural Language Learning (CoNLL), Aug 2016.

[2] Sutskever et al., "Sequence to Sequence Learning with Neural Networks", *Conference on Neural Information Processing Systems (NIPS 2014)*.

[3] Cho, B. van Merrienboer, D. Bahdanau and Y. Bengio, "On the Properties of Neural Machine translation: Encoder Decoder Approaches", *Workshop on Syntax Semantics and Structure in Statistical Translation (SSST-8)*, Oct 2014.

[4] Peter J. Liu et al., "Generating Wikipedia by Summarizing Long Sequences", International Conference on Learning Renresentation (ICLR), 2018.

[5] K. Chen et al., "Extractive Broadcast News Summarization Leveraging Recurrent Neural Network Language Modeling Techniques," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 8, pp. 1322-1334, Aug. 2015.

[6] M. Steinert, R. Granada, J. P. Aires and F. Meneguzzi, "Automating News Summarization with Sentence Vectors Offset," 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), 20.

[7] S. Huang, R. Wang, Q. Xie, L. Li and Y. Liu, "An ExtractionAbstraction Hybrid Approach for Long Document Summarization," 2019 6th International Conference on Behavioral, Economic and Socio- Cultural Computing (BESC), 2019.

[8] Rahul, S. Adhikari and Monika, "NLP based Machine Learning Approaches for Text Summarization," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020, pp. 535-538,