MULTIMEDIA SUMMARIZATION USING BERT

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

Submitted by

TANISHK VENKAT MAHESH BABU GALI	**318126510018**
JYOTHI SREEMAYEE VUPPALA	**318126510023**
VANNE KALYAN	**318126510057**
PRANAV CHANDRA PARUCHURI	**318126510044**

Under the guidance of

Dr.K.Selvani Deepthi (M.Tech,Ph.D)
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (UGC AUTONOMOUS)
(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)
Sangivalasa, Bheemili mandal, Visakhapatnam district (A.P)

2018-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (UGC AUTONOMOUS)

(***Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade***) **Sangivalasa, Bheemili mandal, Visakhapatnam district (A.P)**

BONAFIDE CERTIFICATE

This is to certify that the project report entitled "**MULTIMEDIA SUMMARIZATION USING BERT"** submitted by **Jyothi Sreemayee.V (316126510023), V.Kalyan (316126510057), P.Pranav Chandra(316126510044)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science Engineering** of Anil Neerukonda Institute of technology and sciences (A), Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

PROJECT GUIDE	**HEAD OF THE DEPARTMENT**
Dr. K. Selvani Deepthi (M.Tech,Ph.D)	Dr. R. Sivaranjini(M.Tech,Ph.D)
Computer Science Engineering	Computer Science Engineering
ANITS	ANITS

DECLARATION

We, **Jyothi Sreemayee.V (318126510023), V.Kalyan (318126510057), P.Pranav Chandra (318126510044)** of final semester B.Tech., in the department of Computer Science and Engineering from ANITS, Visakhapatnam, hereby declare that the project work entitled **MULTIMEDIA SUMMARIZATION USING BERT** is carried out by us and submitted in partial fulfillment of the requirements for the award of **Bachelor of Technology in Computer Science Engineering** , under Anil Neerukonda Institute of Technology & Sciences(A) during the academic year 2018-2022 and has not been submitted to any other university for the award of any kind of degree.

JYOTHI SREEMAYEE VUPPALA	318126510023
V.KALYAN	318126510057
P.PRANAV CHANDRA	318126510044

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of a task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement always boosted morale. We take a great pleasure in presenting a project, which is the result of a studied blend of both research and knowledge.

We first take the privilege to thank the Head of our Department, DR.R.SIVARANJANI, for permitting us in laying the first stone of success and providing the lab facilities, we would also like to thank the other staff in our department and lab assistants who directly or indirectly helped us in successful completion of the project.

We feel great to thank DR.K.SELVANI DEEPTHI , who is our project guide and who shared their valuable knowledge with us and made us understand the real essence of the topic and created interest in us to work day and night for the project.

PROJECT STUDENTS:

JYOTHI SREEMAYEE VUPPALA
(318126510023)

V. KALYAN
(318126510057)

P. PRANAV CHANDRA
(318126510044)

PROBLEM STATEMENT:

In recent years, the majority of people have become more reliant on internet material known as OTT (Over to Top), which comprises more video and audio content on a single platform. Because of the high demand for multimedia content, numerous massive models have emerged. Multimedia Summarization was one of them, and it was used by a variety of systems. The goal of video summarization is to speed up the search for large collections of video data for efficient access and display of video content. Viewing the overview allows users to quickly determine the usefulness of the video. The end-user cannot decide whether the video/ audio is informative or not so we made a project for making the end user flexible. Our project is to develop a model using deep learning which helps the user to show the gist of the video/audio so that the end-user may get a quick overview of the material.

ABSTRACT:

Multimedia summarization primarily seeks to condense any type of multimedia, such as audio, video, text, and so on, into a concise summary. Using the current BERT (Bi-directional Encoder Representational Transformer) technology, we employ extractive summarisation to provide a summary. We first convert the video to text using nlp, which includes capabilities such as quiet audio and other options. The text is then summarized using BERT. When it comes to accuracy we use a ROUGE score to analyze the summary. With the tvsum50 dataset, we're creating a pre-trained model. Prior to its introduction, BERT has previously been trained on a variety of datasets. BERT makes use of a transformer to assist it understand the context of a word in a phrase. Unlike directed models, the transformer scans all of the words at once with the speciality of getting the right meaning of the word. The proposed model has achieved 0.62 precision, 0.87 recall, and 0.68 fmeasure i.e., 68% accuracy on Rouge1 evaluation, 0.51 precision, 0.55 recall, and 0.52 fmeasure i.e., 52% accuracy on Rouge2 evaluation, and 0.58 precision, 0.77 recall, and 0.63 fmeasure i.e., 63% accuracy on RougeL evaluation.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

SYMBOL	MEANING
Σ	Summation(Uppercase Sigma)
α	Alpha
Φ	Phi
tanh	Hyperbolic tangent function
σ	Sigmoid Function(Lowercase Sigma)

CONTENTS:

Introduction:

With the advancement of technology, the internet is accessible through various devices like smartphones, smartwatches and within the reach of common people. That leads to the accessibility of a lot of information through the world wide web (WWW). More information on the internet is in text form so sometimes it becomes so difficult to select only required information from large texts. Due to the large volume of information, manual summarization of information is very challenging and also a time-consuming task. Thus, we need an automatic text summarization system. The process of producing summaries from the huge sets of information while maintaining the actual context of information is called Text Summarization. The summary should be fluent and concise throughout.Google uses featured snippets to show the summary of the article or the answer for the user's query. These snippets are basically extracted from webpages.

Summaries make the task of understanding the meaning of text easier. Text summarization helps users to manage vast amounts of information by condensing documents and including more relevant facts into them . Text summarization process contains three steps: analysis, transformation, and synthesis . The Analysis step analyzes and shows the general steps of text summarization or ATS. The input of the system can be single or multiple documents. It depends on the user requirement. The next step is Preprocessing in this step stop words removed and tokenization performed.

Sentence Analysis step includes sentence scoring and sentence ranking to rank the sentence. From this, the final summary is generated which is the final and the last step of the system.

The evaluation of a summary quality is a very ambitious task. Serious questions remain concerning the appropriate methods and types of evaluation. There are a variety of possible bases for the comparison of summarization systems performance. We can compare a system summary to the source text, to a human generated summary or to another system summary. Summarization evaluation methods can be broadly classified into two categories. In extrinsic evaluation, the summary quality is judged on the basis of how helpful summaries are for a given task, and in intrinsic evaluation, it is directly based on analysis of the summary. The latter can involve a comparison with the source document, measuring how many main ideas of the source document are covered by the summary or a content comparison with an abstract written by a human. The problem involves a comparison with the source document, measuring how many main ideas of the source document are covered by the summary or a content comparison with an abstract written by a human. The problem of matching the system summary against an “ideal summary” is that the ideal summary is hard to establish. The human summary may be supplied by the author of the article, by a judge asked to construct an abstract, or by a judge asked to extract sentences. There can be a large number of abstracts that can summarize a given document. The intrinsic evaluations can then be broadly divided into content evaluation and text quality evaluation. Whereas content evaluations measure the ability to identify the key topics, text quality evaluations judge the readability, grammar and coherence of automatic summaries.

Automatic text summarization technology is also able to summarize multiple documents and then presents the summary of all multiple documents into one summary . Such summaries are used for summarizing the multiple news from different sources and reviewing products. Search engines can also take the advantage of automatic text summarization to provide the summarized information of large web pages to users. The rest of the paper is organized as follows: It contains the different challenges in text summarization. Summary contains the classification of text summarization techniques. Summary contains literature review as we discussed in a final conclusion.

Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning.

There are broadly two different approaches that are used for text summarization:

- Extractive Summarization
- Abstractive Summarization

Extractive Summarization: In this process, we focus on the vital information from the input sentence and extract that specific sentence to generate a summary. There is no generation of new sentences for summary, they are exactly the same that are present in the original group of input sentences.

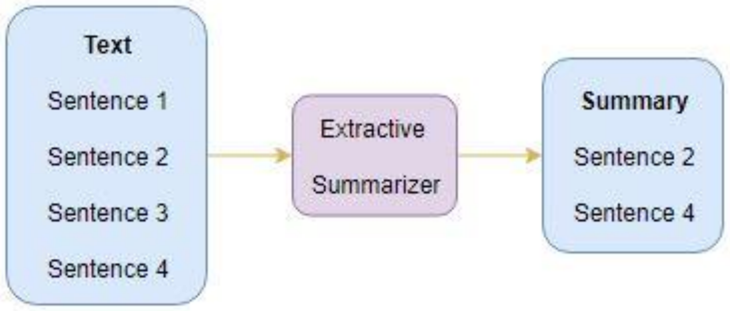


Fig-1

Example :

Source text: DataFlair is an online, immersive, instructor-led, self-paced technology school for students around the world. DataFlair offers lifetime support, quizzes to sharpen student's knowledge, and various live project participation. DataFlair machine learning projects are best for students to gain practical knowledge for real-world problems.

Summary: DataFlair is an online school for students around the world. DataFlair offers lifetime support, quizzes, and live projects. DataFlair machine Learning projects are best to gain knowledge for real-world problems.

Abstract Summarization: This is the opposite of Extractive summarization where it takes an exact sentence to generate a summary. Abstract Summarization focuses on the vital information of the original group of sentences and generates a new set of sentences for the summary. This new sentence might not be present in the original sentence.

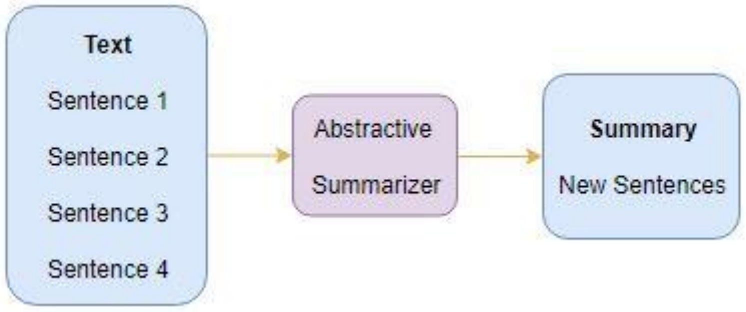


Fig-2

Example :

Source text: DataFlair is an online, immersive, instructor-led, self-paced technology school for students around the world. DataFlair offers lifetime support, quizzes to sharpen student's knowledge, and various live project participation. DataFlair machine learning projects are best for students to gain practical knowledge for real-world problems.

Summary: DataFlair is an online school where students are offered various quizzes and projects including machine learning to solve real-world problems.

Deep Learning

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks.The term "deep" usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150.Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.

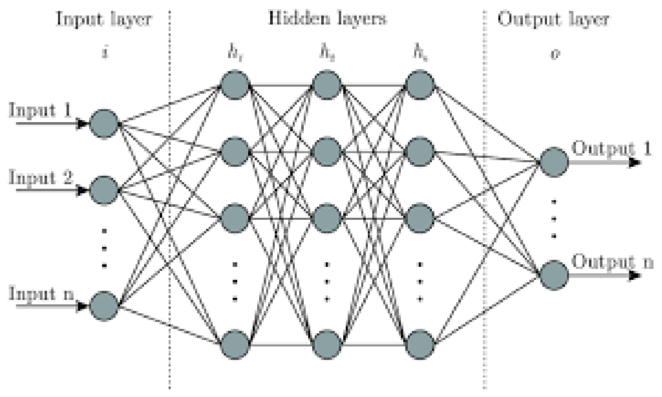


Fig-3

One of the most popular types of deep neural networks is known as convolutional neural networks (CNN or ConvNet). A CNN convolves learned features with input data, and uses 2D convolutional layers, making this architecture well suited to processing 2D data, such as images.CNNs eliminate the need for manual feature extraction, so you do not need to identify features used to classify images. CNN works by extracting features directly from images. The relevant features are not pretrained; they are learned while the network trains on a collection of images. This automated feature extraction makes deep learning models highly accurate for computer vision tasks such as object classification.

Neural Networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots inartificial intelligence, is swiftly gaining popularity in the development of trading systems.Neural networks, in the world of finance, assist in the development of such processes as time-series forecasting,algorithmic trading, securities classification, credit risk modeling, and constructing proprietary indicators and price derivatives.A neural network works similarly to the human brain's neural network. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis.

ANN

ANN are artificially intelligent structures that simulate the fabrication of a human brain. The neural network takes in an input and processes them one by one in layers. Each layer is a mathematical formula that takes the output from the previous layer and profoundly processes it for make it ideal for use within the next layer. At heart, this network is made up of neurons where each neuron represents a core idea or factor that affects surrounding neurons. ANNs make it possible for computers to mimic the behaviors of the human brain very closely — something that is sure to significantly increase their accuracy and productivity as humans develop more complex computing environments.

CNN

Deep Learning, a subset of Machine Learning, consists of algorithms that are inspired by the functioning of the human brain or the neural networks.These structures are called Neural Networks. It teaches the computer to do what naturally comes to humans. In Deep learning, there are several types of models such as the Artificial Neural Networks (ANN), Autoencoders, Recurrent Neural Networks (RNN) and Reinforcement Learning. But there has been one particular model that has contributed a lot in the field of computer vision and image analysis which is the Convolutional Neural Networks (CNN) or the ConvNets. CNNs are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analyzing visual images. Their applications range from image and video recognition, image classification, medical image analysis, computer vision and natural language processing.

ANN vs CNN:

There are two major foundations to AI in the name of Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). While both of them are revolutionary in their own sense and source from somewhat similar concepts, they do tend to accommodate entirely different characteristics — a fact that becomes a source of massive distinction when it comes to their capabilities, shortcomings, and possible applications. As a result, it becomes imperative for the tech geeks to discern their definitions and carve out the distinctions so that these can be put to use in the most ideal way. Simply put, comprehension of their theoretical prowess allows for informed decisions when developing AI products.

MoviePy:

MoviePy is a Python module for video editing, which can be used for basic operations (like cuts, concatenations, title insertions), video compositing (a.k.a. non-linear editing), video processing, or to create advanced effects. It can read and write the most common video formats, including GIF.

Speech Recognition:

Speech recognition, or speech-to-text, is the ability of a machine or program to identify words spoken aloud and convert them into readable text. Rudimentary speech recognition software has a limited vocabulary and may only identify words and phrases when spoken clearly. More sophisticated software can handle natural speech, different accents and various languages.Speech recognition uses a broad array of research in computer science, linguistics and computer engineering. Many modern devices and text-focused programs have speech recognition functions in them to allow for easier or hands-free use of a device.

Pydub:

Audio files are a widespread way of transferring information. So let's learn how to work with audio files using python. Python provides a module called Pydub to work with audio files. Pydub is a library to work with only .wav files . By using this library we can play, split, merge, edit our .wav audio files .

Audio Segment:

Audio segmentation refers to the class of theories and algorithms designed to automatically reveal semantically meaningful temporal segments in an audio signal, also referred to as auditory scenes . These scenes can be seen as equivalents of paragraphs in text, and can serve as input into audio categorization processes, either supervised (audio classification) or unsupervised (audio clustering). Through these processes, semantically similar auditory scenes can be grouped together and/or labeled using semantic indexes to provide multi-level, non-linear content-based access to large audio documents and collections.

Split on Silence:

We have defined the audio_chunk variable and by using the split_on_silence function we are splitting the audio file. This function takes sound as a parameter which is our audio file next it takes min_silence_len by default it is 1000. The minimum length for silent sections is in milliseconds. If it is greater than the length of the audio segment an empty list will be returned. Here we are giving it as 500 milliseconds. This function returns a list of audio segments.

OS:

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The os and os.path modules include many functions to interact with the file system.The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

Matplot:

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB.Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.A Python matplotlib script is structured so that a few lines of code are all that is required in most instances to generate a visual data plot. The matplotlib scripting layer overlays two APIs:

- The pyplot API is a hierarchy of Python code objects topped by matplotlib.pyplot
- An OO (Object-Oriented) API collection of objects that can be assembled with greater flexibility than pyplot. This API provides direct access to Matplotlib's backend layers.

Pandas:

Pandas is mostly used for handling the missing data , inserting and deleting the columns,in data alignment,conversions,slicing. fancy indexing ,merging and joining the datasets and reshaping the datasets. It allows us to analyze big data and make conclusions based on statistical theories. It can clean messy data sets and make them readable and relevant.

NumPy:

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. NumPy is a Python package. It stands for 'Numerical Python'**.**It is the fundamental package for scientific computing in python. It provides a multidimensional array object, various derived objects for fast operations on arrays like sorting, selecting, shape, manipulation, basic linear algebra, random simulation, I/O and many more.

Spacy:

A Python-based open source library programme for sophisticated natural language processing. Spacy is a current and decisive NLP framework that is the traditional source for conducting NLP with Python and includes outstanding characteristics such as speed, accuracy, and flexibility. It immediately became an important part of the NLP production pipeline. Spacy outperforms NLTK in terms of performance since it has its own statistical method and models.

Torch:

PyTorch is an optimized tensor library primarily used for Deep Learning applications using GPUs and CPUs. It is an open-source machine learning library for Python, mainly developed by the Facebook AI Research team. It is one of the widely used Machine learning libraries, others being TensorFlow and Keras. Here is the Google Search Trends which shows that the popularity of the PyTorch library is relatively higher compared to TensorFlow and Keras.

PyTorch is built in a python and torch library which supports computations of tensors on Graphical Processing Units. Currently is the most favored library for the deep learning and artificial intelligence research community.

Tqdm:

tqdm is a library in Python which is used for creating Progress Meters or Progress Bars. tqdm got its name from the Arabic name *taqaddum* which means ‘progress’.

Implementing tqdm can be done effortlessly in our loops, functions or even Pandas. Progress bars are pretty useful in Python because:

1. One can see if the Kernel is still working

2. Progress Bars are visually appealing to the eyes

3. It gives Code Execution Time and Estimated Time for the code to complete which would help while working on huge datasets

Auto Tokenizer:

The tokenize module provides a lexical scanner for Python source code, implemented in Python. The scanner in this module returns comments as tokens as well, making it useful for implementing “pretty-printers”, including colorizers for on-screen displays
The tokenize() generator requires one argument, readline, which must be a callable object which provides the same interface as the io.IOBase.readline() method of file objects. Each call to the function should return one line of input as bytes.
The generator produces 5-tuples with these members: the token type; the token string; a 2-tuple (srow, scol) of ints specifying the row and column where the token begins in the source; a 2-tuple (erow, ecol) of ints specifying the row and column where the token ends in the source; and the line on which the token was found. The line passed (the last tuple item) is the physical line. The 5 tuple is returned as a named tuple with the field names: type string start end line.

Auto Model:

A generic model class that will be instantiated as one of the base model classes of the library when created with the
AutoModel.from_pretrained(pretrained_model_name_or_path) or the
AutoModel.from_config(config) class methods.

This class cannot be instantiated using _init_()
Instantiates one of the base model classes of the library from a configuration

Cuda:

The NVIDIA® CUDA® Toolkit provides a development environment for creating high performance GPU-accelerated applications. With the CUDA Toolkit, you can develop, optimize, and deploy your applications on GPU-accelerated embedded systems, desktop workstations, enterprise data centers, cloud-based platforms and HPC supercomputers. The toolkit includes GPU-accelerated libraries, debugging and optimization tools, a C/C++

compiler, and a runtime library to build and deploy your application on major architectures including x86, Arm and POWER.

Using built-in capabilities for distributing computations across multi-GPU configurations, scientists and researchers can develop applications that scale from single GPU workstations to cloud installations with thousands of GPUs.

Limitations of the Encoder – Decoder Architecture

As useful as this encoder-decoder architecture is, there are certain limitations that come with it.

- The encoder converts the entire input sequence into a fixed length vector and then the decoder predicts the output sequence. This works only for short sequences since the decoder is looking at the entire input sequence for the prediction
- Here comes the problem with long sequences. It is difficult for the encoder to memorize long sequences into a fixed length vector

A potential issue with this encoder-decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector. This may make it difficult for the neural network to cope with long sentences. The performance of a basic encoder-decoder deteriorates rapidly as the length of an input sentence increases.

Neural Machine Translation by Jointly Learning to Align and Translate

So how do we overcome this problem of long sequences? This is where the concept of attention mechanism comes into the picture. It aims to predict a word by looking at a few specific parts of the sequence only, rather than the entire sequence. It really is as awesome as it sounds!

Literature Survey:

Derek Miller (2019) [1] research on extractive text summarization empowering BERT proposed a technique for extractive summarization. In this study, a RESTful service built in Python that uses the BERT model for text embeddings and for summary selection, K-Means clustering was used to find sentences that were closest to the centroid. The research's goal was to give students a lecture summary tool which summarizes based on the minutes they wanted to spend on it. In addition to summary work, the research also provides administration of lectures and summaries, as well as content storage. The proposed model was not able to provide accurate extractive text summarization.

Bowen Tan et al. (2020) [2] presented a technique using BERT and GPT-2 which are used to automatically summarize COVID-19 medical research articles. The dataset used for this research is taken from the Open Research Dataset Challenge (COVID-19). The authors used recent breakthroughs in NLP pre-trained models, such as OpenAI GPT-2, and BERT and ROUGE score as evaluation metric. The main disadvantage of this model is in order to perform well, the model needs more computational power.

Dmitrii Aksenov et al. (2020) [3] in their study using BERT to present Abstractive Text Summarization based on Locality Modeling and Language Model Conditioning. The BERT language model is used to train the decoder and encoder of a Transformer-based neural network. The authors present a new BERT-windowing method for chunk-wise processing of texts longer than the BERT window size. The encoder's first layers incorporate convolutional self-attention having 2-dimensions to achieve this. The models' results are compared to baseline and state-of-the-art models on the CNN/Daily Mail dataset. In order to demonstrate applicability in German, the authors trained the model on the SwissText dataset.

Eylampios Apostolidis et al. (2021) [4] provided a survey on Deep learning techniques which are used to summarize videos. In this survey, current developments in the field are focussed and provides a detailed overview of available deep-learning-based algorithms for video summarization. After presenting the reason behind the advancement of technologies summarizing videos, the authors define the video summarization problem and go over the key aspects of a typical analysis deep-learning-based pipeline.

G. Vijay Kumar et al. (2021) [5] demonstrated a technique for text summarization using NLP. The authors used a Text Rank algorithm which uses weighted, undirected graphs. The algorithm is implemented using the Gensim Library in NLP. The disadvantages of the proposed technique are the model was not evaluated using any performance metrics and the architecture of the proposed model is not clear.

Sanjana R et al. (2021) [6] created a technique for Video Summarization using NLP. The authors proposed NLP-based techniques which are used to create an automatic video summarization algorithm to provide a brief and simple video description of various YouTube videos. The suggested method starts by summarizing YouTube video transcripts, from which a summary video is created. The drawback of this model is the model is not able to perform well on short videos.

Vishal Pawar et al. (2019) [7] proposed a methodology for Abstractive Text Summarization of Multimedia News Content using RNN. The methodology consists of a technique that combines discourse management, PC vision, NLP, and Recurrent Neural Network (RNN) strategies to investigate the rich data included in various types of data and gain a better understanding of news content. The drawback of this corpus. The problems encountered during this research were generation of text while repetition of specific phrases and it is not performing well while finding out-of-vocabulary words.technique is, no usage of the modern text summarization algorithms and not able to generate a comprehensive summary.

Pankaj Gupta et al. [8] In this paper the author has reviewed different techniques of Sentiment analysis and different techniques of text summarization. Sentiment analysis isa machine learning approach in which a machine learns and analyzes the sentiments, emotions present in the text. Machine learning methods like Naive Bayes Classifier and Support Machine Vectors (SVM) are used.these methods are used to determine the emotions and sentiments in the text data like reviews about movies or products. Text summarization, uses the natural language processing (NLP) and linguistic features of sentences are used for checking the importance of the words and sentences that can be included in the final summary. In this paper, a survey has been done of previous research work related to text summarization and Sentiment analysis, so that new research areas can be explored by considering the merits and demerits of the current techniques and strategies.

Chin-Yew Lin [9] In this paper the author introduced Recall- Oriented Understudy for Gisting Evaluation ROUGE. That is an automatic evaluation package for text summarization. The paper also introduced four different measures of ROUGE: - ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-S. It measures the quality of summary by comparing the generated summary with other ideal summaries that are created by humans. These methods are efficient for automatic evaluation of single document summary as well as multi-document summaries.

Akshil Kumar [10] In this paper the author has analyzed and compared the performance of three different algorithms. Firstly, the different text summarization techniques explained. Extraction based techniques are used to extract important keywords to be included in the summary. For comparison three comparison three keyword extraction algorithms namely TextRank, LexRank, Latent Semantic Analysis were used. Three algorithms are explained and implemented in the python language. The ROUGE 1 is used to evaluate the effectiveness of the extracted keywords. The results of the algorithms are compared with the handwritten summaries and evaluate the performance. In the end, the TextRank Algorithm gives a better result than the other two algorithms.

N. Moratanch [11] In this paper the author presents an exhaustive survey on abstraction based text summarization techniques. The paper presents a survey on two broad abstractive summary approaches:Structured based abstractive summarization and Semantic-based abstractive summarization. The author presents the review of various researches on both approaches of abstractive summarization. The author also covered the various methodologies and challenges, in abstractive summarization.

Dharmendra Hinhu [12] In this paper the author uses the extractive text summarization. The author gives the Wikipedia articles as input to the system and identifies text scoring.Firstly, the sentences are Tokenized through pattern matching using regular expressions. Then we get data in the form of a set of words, then stop words are removed from the set of words. The words are then stemmed. Then traditional methods are used for scoring of the sentences. Scoring helps in classifying the sentences if they are included in summary or not. It is found that scoring sentences based on citation gives better results.

N. Moratanch [13]In this paper the author presents a comprehensive review of extraction based text summarization techniques. In this paper the author provides a survey on extractive summarization approaches by categorizing them in: Supervised learning approach and Unsupervised learning approach. Then different methodologies, the advantages are presented in the paper. The author also includes various evaluation methods ,challenges and future research direction in the paper.

Methodology:

We can build a Seq2Seq model on any problem which involves sequential information. This includes Sentiment classification, Neural Machine Translation, and Named Entity Recognition – some very common applications of sequential information.In the case of Neural Machine Translation, the input is a text in one language and the output is also a text in another language.Our objective is to build a text summarizer where the input is a long sequence of words (in a text body), and the output is a short summary (which is a sequence as well). So, we can model this as a Many-to-Many Seq2Seq problem. Below is a typical Seq2Seq model architecture:

There are two major components of a Seq2Seq model:

- Encoder
- Decoder

Let's understand these two in detail. These are essential to understand how text summarization works underneath the code.

The Encoder-Decoder architecture is mainly used to solve the sequence-to-sequence (Seq2Seq) problems where the input and output sequences are of different lengths.

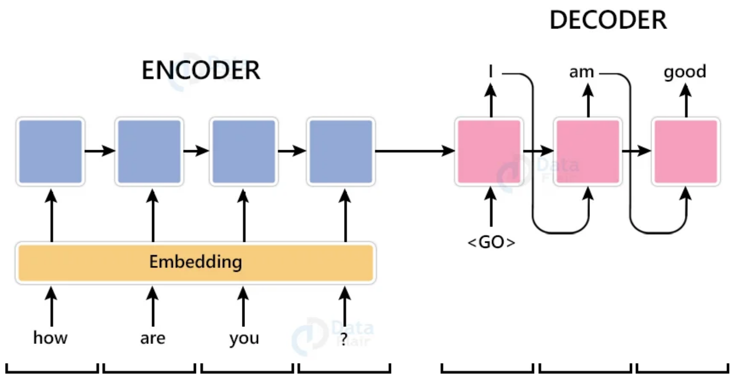


Fig-4

Let’s understand this from the perspective of text summarization. The input is a long sequence of words and the output will be a short version of the input sequence.

Generally, variants of Recurrent Neural Networks (RNNs), i.e. Gated Recurrent Neural Network (GRU) or Long Short Term Memory (LSTM), are preferred as the encoder and decoder components. This is because they are capable of capturing long term dependencies by overcoming the problem of vanishing gradient.

We can set up the Encoder-Decoder in 2 phases:

- Training phase
- Inference phase

Let’s understand these concepts through the lens of an LSTM model within the phases of encoder and decoder.

Encoder:

Encoder Model is used to encode or transform the input sentences and generate feedback after every step. This feedback can be an internal state i.e hidden state or cell state if we are using the LSTM layer. Encoder models capture the vital information from the input sentences while maintaining the context throughout.

In Neural Machine translation, our input language will be passed into the encoder model where it will capture the contextual information without modifying the meaning of the input sequence. Outputs from the encoder model are then passed into the decoder model to get the output sequences.

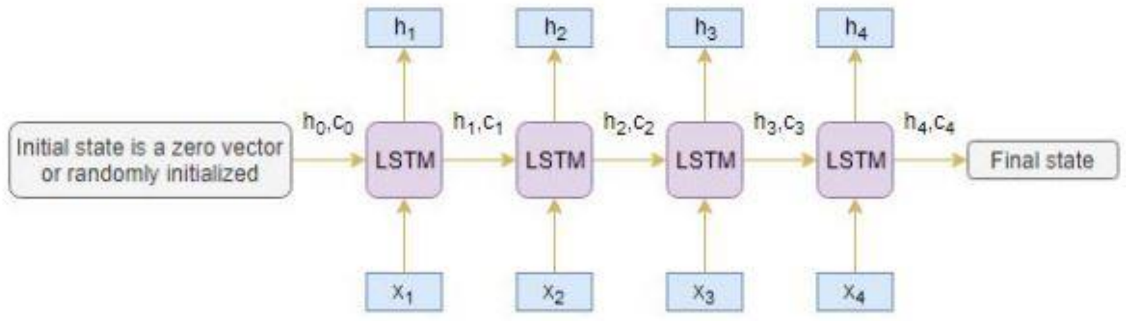
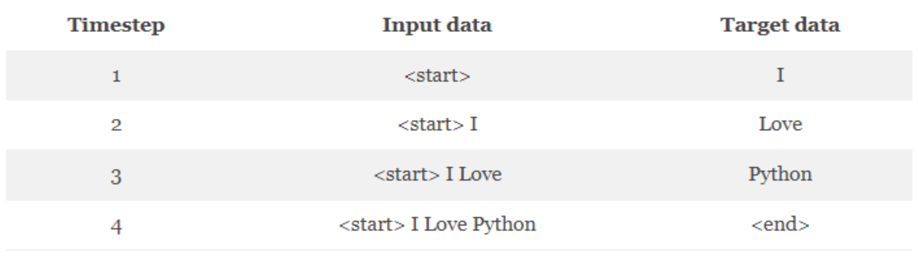


Fig-5

Decoder:

The decoder model is used to decode or predict the target sentences word by word. Decoder input data takes the input of target sentences and predicts the next word which is then fed into the next layer for the prediction. ‘<start>’ (start of target sentence) and ‘<end>’ (end of target sentence) are the two words that help the model to know what will be the initial variable to predict the next word and the ending variable to know the ending of the sentence. While training the model, we first provide the word ‘<start>’, the model then predicts the next word that is the decoder target data. This word is then fed as input data for the next timestep to get the next word prediction.

For example, if our sentence is **‘ I Love Python’** so we will add ‘<start>’ at starting and ‘<end>’ at the end of the sentence therefore our sentence will be **‘ <start> I Love Python <end> ’** now let’s see how it works.



Timestep	Input data	Target data
1	<start>	I
2	<start> I	Love
3	<start> I Love	Python
4	<start> I Love Python	<end>

Table-1

As you can see our input data will start from ‘<start>’ and the target will predict the next word with the help of input data at every timestep. Our input data doesn’t contain the last word as our target data at the last timestep is ‘<end>’ which tells us that we have reached the end of our sentence and stop the iteration. The same way our target data will be one-time step ahead as the first word ‘<start>’ is provided by the Input data.

Video to Audio:

MoviePy is a python module for video editing , which can be used for basic operations (like cuts, concatenations, title insertions) , video compositing (non linear editing) , video

processing or to create advanced effects. It can read and write the most common video formats, including GIF.
You need the function write_videofile, If you do not it won't reflect on the hard disk file as the file was loaded in RAM.We can merge 2 video files to a single file in order of our requirement.

MoviePy is a python module for video editing , which can be used for basic operations on videos and GIFs. Visual multimedia source that combines a sequence of images to form a moving picture. The video transmits a signal to a screen and processes the order in which the screen captures should be shown. Videos usually have audio components that correspond with the pictures being shown on the screen.

Video is formed by the frames, a combination of frames creates a video each frame is an individual image. We can store the specific frames at any time with the help of save frame method with the Video file Clip object.

Audio to Text:

When an object vibrates, the air molecules oscillate to and fro from their rest position and transmits its energy to neighboring molecules. This results in the transmission of energy from one molecule to another which in turn produces a sound wave.
Parameters of an audio signal:

- Amplitude: Amplitude refers to the maximum displacement of the air molecules from the rest position
- Crest and Trough: The crest is the highest point in the wave whereas trough is the lowest point
- Wavelength: The distance between 2 successive crests or troughs is known as a wavelength
- Cycle: Every audio signal traverses in the form of cycles. One complete upward movement and downward movement of the signal form a cycle
- Frequency: Frequency refers to how fast a signal is changing over a period of time

Sampling the signal is a process of converting an analog signal to a digital signal by selecting a certain number of samples per second from the analog signal. Can you see what we are doing here? We are converting an audio signal to a discrete signal through sampling so that it can be stored and processed efficiently in memory.

The input audio for processing is obtained by recording using the device’s microphone. The recording process is facilitated by GUI based buttons. The recording starts when the button Start Recording is pressed and stops on the Stop recording button click event. The python module used for GUI creation is tkinter. The audio recording and processing are done with the help of pyaudio module. Audio is recorded by continuously appending frames of audio. The final output of this phase is in the form of an audio file of the “wav” format. The format wav is chosen since it is suitable for further processing.An audio signal is a continuous representation of amplitude as it varies with time. Here, time can even be in picoseconds. That is why an audio signal is an analog signal.Analog signals are memory hogging since they have an infinite number of samples and processing them is highly computationally demanding. Therefore, we need a technique to convert analog signals to digital signals so that we can work with them easily.

Automatic Speech Recognition (ASR) is the technology that allows us to convert human speech into digital text. This tutorial will dive into the current state-of-the-art model called Wav2vec2 using the Huggingface transformers library in Python.Wav2Vec2 is a pre-trained model that was trained on speech audio alone and then followed by fine-tuning on transcribed speech data . It has outperformed previous semi-supervised models.As in Masked Language Modeling, Wav2Vec2 encodes speech audio via a multi-layer convolutional neural network and then masks spans of the resulting latent speech representations. These representations are then fed to a Transformer network to build contextualized representations.

Pooling

Pooling layer plays an important role in pre-processing an image. Pooling layer reduces the number of parameters when the images are too large. Pooling is "downscaling" of the image obtained from the previous layers. It can be compared to shrinking an image to reduce its pixel density. Spatial pooling is also called downsampling or subsampling, which reduces the dimensionality of each map but retains the important information. There are the following types of spatial pooling:

Max Pooling:

Max pooling is a sample-based discretization process. Its main objective is to downscale an input representation, reducing its dimensionality and allowing for the assumption to be made about features contained in the sub-region binned. Max pooling is done by applying a max filter to non-overlapping sub-regions of the initial representation.

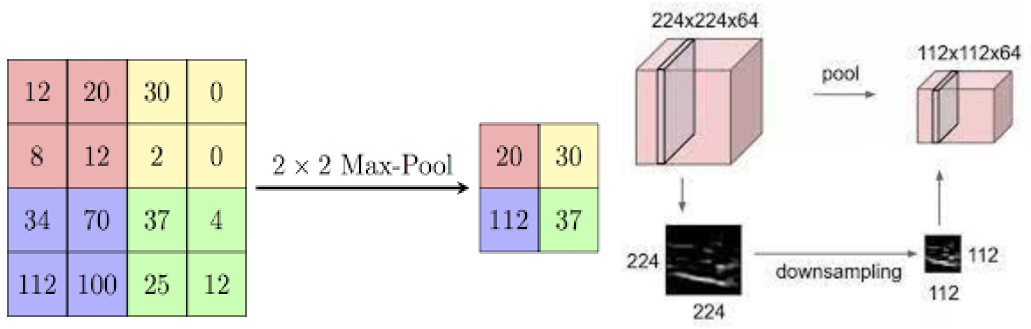


Fig-6

Average Pooling:

Down-scaling will perform through average pooling by dividing the input into rectangular pooling regions and computing the average values of each region.

Syntax:

layer = averagePooling2dLayer(poolSize)

layer = averagePooling2dLayer(poolSize,Name,Value)

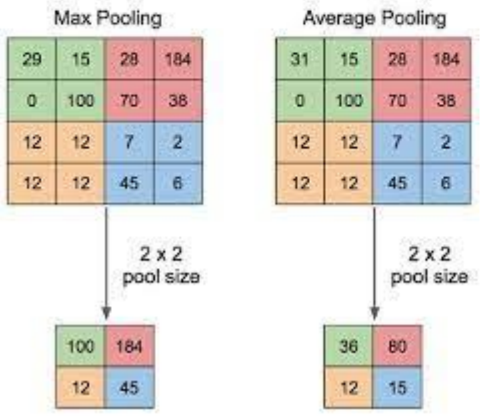


Fig-7

Sum Pooling:

The sub-region for sum pooling or mean pooling are set exactly the same as for max-pooling but instead of using the max function we use sum or mean. Sum pooling (which is proportional to Mean pooling) measures the mean value of existence of a pattern in a given region.

Global average pooling 2D:

The 2D Global average pooling block takes a tensor of size (input width) x (input height) x (input channels) and computes the average value of all values across the entire (input width) x (input height) matrix for each of the (input channels). In our model we have implemented the MaxPooling2D function and it is preferred over average pooling and other techniques as it picks max value from that particular size instead of the average. This layer helps in picking all the important regions in feature maps of the input, reduces the dimensions and extracts the prominent features. This takes a single argument for size of kernel i.e. 2X2 which is preferred and the length of stride will always be similar to length of window.

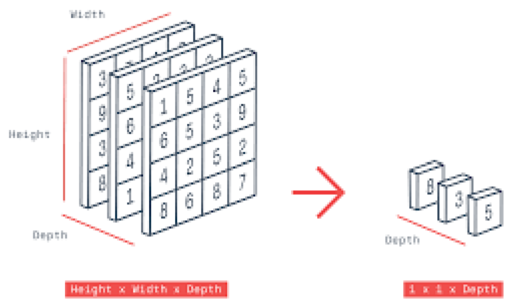


Fig-8

Data Preprocessing:

Data scaling or normalization is a process of making model data in a standard format so that the training is improved, accurate, and faster. The method of scaling data in neural networks is similar to data normalization in any machine learning problem. There are few preprocessing steps involved in our model which are mainly :

1.Normalization

2. Data Augmentation

Normalization:

The data set has normalized the images i.e adjusting the dimensions of the image, which was considered as a preprocessing strategy for image classification, performed before being sent straight to the model for training. After normalizing the picture it will be having pixel values in such a way that they have a comparable distribution of data with values ranging from 0 to 1, This is because if a default picture is given, it will have values from 0 to 255, and resulting in greater numeric values thereby complicating the process. By dividing these by 255 reduces computation complexity by ranging values between 0 and 1. This data set has been resized to 48 x 48 pixels and will be used as input to our model. This step is necessary since our photos are often of varying sizes, and our model accepts input images of constant size, but FER-2013 images may vary greatly in size. It makes learning quicker as we pass input in similar dimensions.

Data Augmentation:

Data augmentation is the process of increasing the amount and diversity of data. We do not collect new data, rather we transform the already present data. I will be talking specifically about image data augmentation in this article. Data augmentation is an integral process in deep learning, as in deep learning we need large amounts of data and in some cases it is not feasible to collect thousands or millions of images, so data augmentation comes to the rescue.It helps us to increase the size of the dataset and introduce variability in the dataset. Data Augmentation is a technique that can be used to artificially expand the size of a training set by creating modified data from the existing one. It is a good practice to use this technique if you want to prevent overfitting, or the initial dataset is too small to train on, or even if you want to squeeze better performance from your model.

Model Fine-Tuning and Classification:

The generated tokens were converted to vector representa- tions and fed to the pre-trained models during the fine-tuning process. Thus, the models were trained on the input vector transformations and their outputs generated. The output was then evaluated using the designated test data, and results were obtained. Emotions were then classified into joy, sadness, fear, anger, guilt, disgust, and shame for each of the pre-trained models in the emotion classification process.

Experimental Analysis and Results:

Dataset:

TVSum50

The TVSum dataset (Title-based Video Summarizing) is used to validate video summarization algorithms.Title-based video summarization is a relatively unex- plored domain; there is no publicly available dataset suit- able for our purpose. We therefore collected a new dataset, *TVSum50*, that contains 50 videos and their shot-level im- portance scores obtained via crowdsourcing.
.It includes 1000 annotations of shot-level relevance rankings determined through crowdsourcing .Based on this dataset, FrameRank, an unsupervised video summarization method is proposed which employs a frame-to-frame level affinity graph to identify coherent and informative frames to summarize a video. We use the Kullback-Leibler(KL)-divergence-based graph to rank temporal segments according to the amount of semantic information contained in the frames of the video.
The video and annotation data allow for the automated evaluation of several video summarizing approaches without the need for user research.The YouTube videos reveal both the video files and the URLs for the videos. Amazon Mechanical Turk was used to annotate the shot-level significance scores; each film was annotated by 20 crowd-workers. The dataset has been verified to ensure that it complies with Yahoo's data protection requirements, which include stringent privacy safeguards.

Our framework consists of four modules: shot segmentation, canonical visual concept learning, shot importance scoring, and summary generation. First, we group se- quences of visually coherent frames into shots so that the resulting summary contains shots rather than keyframes. Next, we learn canonical visual concepts using our novel co-archetypal analysis, which learns a joint-factorization of a video and images.We then measure the importance of each frame using the learned factorial representation of the video, and combine the importance measures into shot-level scores. Finally, a summary is generated by maximizing the total importance score with a summary length budget.

The primary task of the dataset is video summarization, where the goal is to create a short, meaningful summary of a given video. The summary may contain a few shots that capture the highlights of a video and are non redundant.Although that task is inherently subjective, we carefully curated the dataset and annotated it so that the evaluation is done in an objective way.

Fig-9

How do we know the quality of the generated summary? We can use the shot-level importance scores included in the dataset.

Each video has the following annotations: We cut a video into 2 second-long shots and asked 20 users to rank the importance of each shot in comparison to other shots from the same video on a scale of 1 to 5 . The average of the comments becomes the video's shot-level significance scores.We may define video summarizing as the 0/1 Knapsack Problem based on the shot-level significance scores: given a time budget , maximize the overall importance score of the shots included in a summary. The answer will be a vector of size T with values of 0 or 1 – we'll name this the gold standard.

If our summary is comparable to the gold standard, it indicates that our summary is similar to how the video would have been summarized by the 20 annotators. We may describe our summary as a T-dimensional vector whose entries are 1 if it is included in the summary.The image clearly portrays a graph of the ranks that are taken from different users.

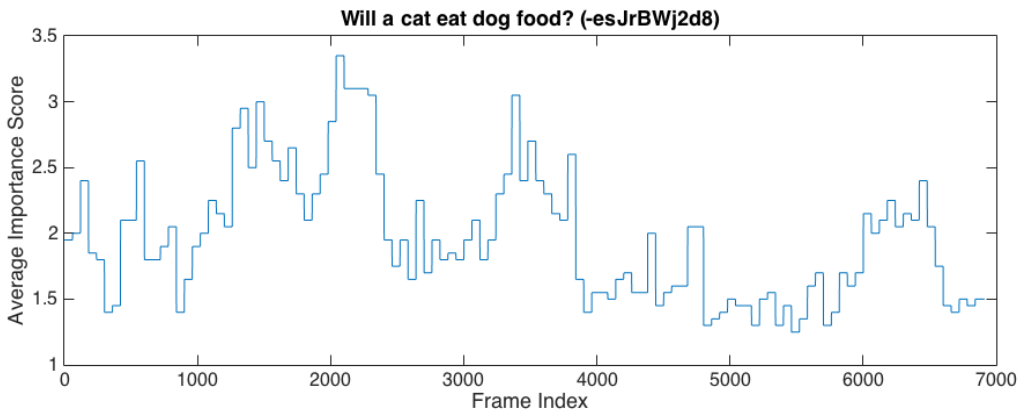


Fig-10

TV Sum Framework:

Our framework consists of four modules: shot segmen- tation, canonical visual concept learning, shot importance scoring, and summary generation. First, we group se- quences of visually coherent frames into shots so that the resulting summary contains shots rather than keyframes. Next, we learn canonical visual concepts using our novel co-archetypal analysis, which learns a joint-factorization of a video and images.We then measure the importance of each frame using the learned factorial representation of the video, and combine the importance measures into shot-level scores. Finally, a summary is generated by maximizing the total importance score with a summary length budget.

Shot Segmentation:

Shot segmentation is a crucial step in video sum- marization for maintaining visual coherence within each shot, which in turn affects the overall quality of a summary. Many existing approaches use heuristics, e.g., uniform seg- mentation . Similarly , we cast the problem as change-point detection, a more principled statistical method to find "changing moments" in a time-series sequence. While using a kernel-based approach, we instead frame the task as a group LASSO problem, based on .

We solve Equation using the group LARS algorithm. Given the maximum number of change points k (set to half the video duration in seconds), we compute an approximate (piecewise-linear) regulariza- tion path by iterating over k steps and adding a change-point to an active set A at each step, with a choice of λ that pro- duces over-segmentation. We then perform model selection tofindtheoptimalk′ ≤kbyfindingasubsetA′ ⊂A Such that it no longer improves the sum-squared errors (SSE) be- tween X and H, up to a threshold θ (set to 0.1). Our method does not require knowing the optimal k a priori thanks to the model selection. Also, the group LARS is highly efficient, i.e., O(dnk); although the model selection takes O(k3) for computing SSE for all k, in practice k ≪ n. This makes our method faster and more practical than the DP solution.

Canonical Visual Concept Learning:

We define canonical visual concepts as the patterns shared between video X and its title-based image search results Y, and represent them as a set of p latent variables Z = [z1,··· ,zp] ∈ Rd×p, where p ≪ d (in our experi- ments, p=200 and d=1,640).

Motivated by archetypal analysis , we find Z by learning the factorizations of X and Y with respect to Z un- der two geometrical constraints: (i) each video frame xi and image yi should be well approximated by a convex com- bination of latent variables Z; (ii) each latent variable zj should be well approximated *jointly* by a convex combina- tion of video frames X and by a convex combination of im- ages AND. Therefore, given Z, each video frame xi and image yi is approximated by ZαXi and ZαYi , respectively, where αXi and αYi are coefficient vectors in the unit simplex Δp:

p Δp =α∈Rp | □α[j]=1andα[j]≥0forallj

j=1

While xi and yi are approximated in terms of Z indepen- dently of each other, each zj is *jointly* approximated by XβXj and YβYj , i.e., zj ≈ XβXj ≈ YβYj , where βXj and βYj are coefficient vectors in Δn and Δm, respectively. This joint approximation encourages Z to capture canonical vi- sual concepts that appear both in a video and an image set, but not in either alone. Inspired by , we refer to the latent variables Z as *co-archetypes* of X and Y, and the process of finding Z as *co-archetypal analysis*.The optimization problem in Equation is non-convex, but is convex when all but one of the vari- ables among Ω are fixed. Block-coordinate descent (BCD) is a popular approach to solving such problems , for its simplicity and efficiency, and for the fact that it is asymp- totically guaranteed to converge to a stationary point . Algorithm shows our optimization procedure.

We cycle through block variables in a deterministic or- der (AX, AY, BX, then BY). For AX, we solve a quadratic program (QP) on each column vector αXi using the Fast It- erative Shrinkage-Thresholding Algorithm (FISTA) :

$$\min_{\alpha \in \Delta p} \| xi - Z\alpha \| 2 \quad (3)$$

We use the same QP solver to obtain the solution AY; these updates are expressed in Lines 6 and 9 of Algorithm . Solving for BX and BY is a bit more involved because of the third term in Equation . When all but one column vec- tor βXj is fixed, the update rule for BX can be expressed as:

where AXj is the j-th row vector. Letting RX = X − ZAX, we can rewrite the above formula as

$$\min_{\beta \in \Delta n} \| RX + (zj - X\beta)AXj \| 2F + \lambda \| zj - X\beta \| 2 \quad (4)$$

We use a generic QP solver to solve Equation and simi- larly for BY; these updates are expressed in Lines 13 and 14 of Algorithm . We acknowledge that we can solve this more efficiently by implementing a customized QP solver using an SMO-style algorithm, expressing the unit sim- plex condition as constraints and solving the problem block- wise (two variables at a time); we leave this as future work

Shot Importance Scoring:

We first measure frame-level importance using the learned factorization of X into XBA. Specifically, we measure the importance of the i-th video frame xi by com- puting the total contribution of the corresponding elements of BA in reconstructing the original signal X, that is, where Bi is the i-th row of the matrix B. Recall that each x is a convex combination of co-archetypes Z, and each z is a convex combination of data X (and of Y). Intuitively, this "chain reaction"-like formulation makes our scoring func- tion measure how representative a particular video frame xi is in the reconstruction of the original video X using Z. Also, the joint-formulation of Z using X and Y allows us to measure the relevance of xi to the canonical visual con- cepts Z shared between the given video and images. We then compute shot-level importance scores by taking an av- erage of the frame importance scores within each shot.

Consequently, our definition of importance captures the relevance and the representativeness of each shot to the canonical visual concepts. Note that, by incorporating the notion of representativeness, it assigns low scores to similar looking, yet less representative frames; hence, it implicitly removes redundant information in the summary.

Summary Generation:

To generate a summary of length l, we solve the follow- ing optimization problem:

where s is the number of shots, vi is the importance score of the i-th shot, and wi is the length of the i-th shot. Note that this is exactly the 0/1 knapsack problem; we solve it using dynamic programming. The summary is then created by concatenating shots with ui $\neq$ 0 in chronological order. Following , we set l to be 15% of the video length.

BERT

Summarization has long been a challenge in Natural Language Processing. To generate a short version of a document while retaining its most important information, we need a model capable of accurately extracting the key points while avoiding repetitive information. Fortunately, recent works in NLP such as Transformer models and language model pretraining have advanced the state-of-the-art in summarization.
In this article, we will explore BERTSUM, a simple variant of BERT, for extractive summarization from Text Summarization with Pretrained Encoders . Then, in an effort to make extractive summarization even faster and smaller for low-resource devices, we will fine-tune DistilBERT and MobileBERT , two recent lite versions of BERT, and discuss our findings.

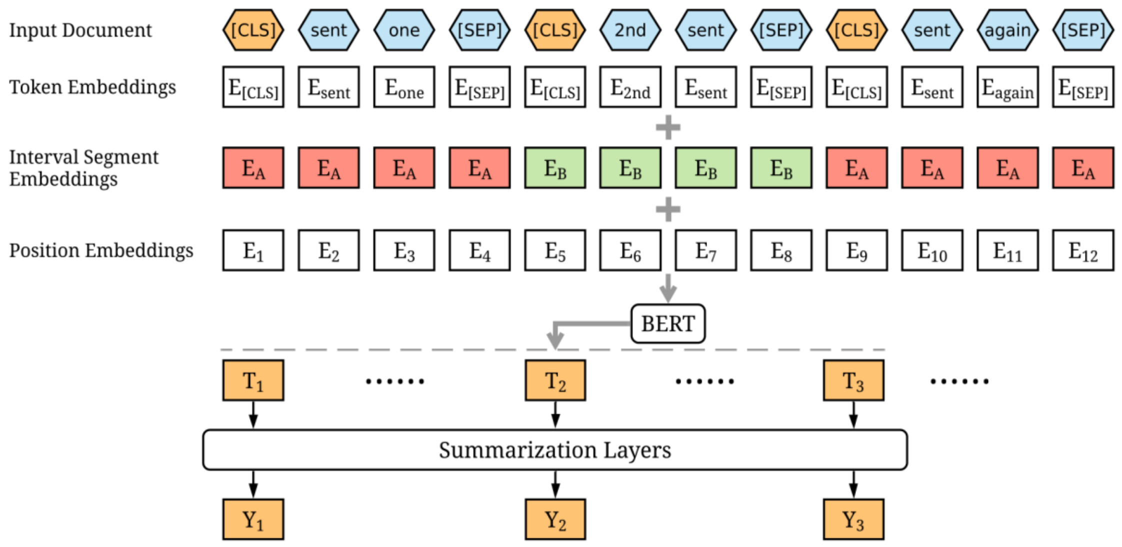


Fig-11

Our BERT encoder is the pretrained BERT-base encoder from the masked language modeling task . The task of extractive summarization is a binary classification problem at the sentence level. We want to assign each sentence a label $y_i \in {0, 1}$ indicating whether the sentence should be included in the final summary. Therefore, we need to add a token before each sentence. After we run a forward pass through the encoder, the last hidden layer of these tokens will be used as the representations for our sentences.

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 perform the NLP tasks as specified, Summarization in our case.

Summarization Classifier

After getting the vector representation of each sentence, we can use a simple feed forward layer as our classifier to return a score for each sentence. In the paper, the author experimented with a simple linear classifier, a Recurrent Neural Network and a small Transformer model with 3 layers. The Transformer classifier yields the best results, showing that inter-sentence interactions through self-attention mechanisms are important in selecting the most important sentences.

So in the encoder, we learn the interactions among tokens in our document while in the summarization classifier, we learn the interactions among sentences.

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.

The complete process can be divided into several phases, as follows:

Encoding Multiple Sentences

In this step sentences from the input document are encoded so as to be preprocessed. Each sentence is preceded by a CLS tag and succeeded by a SEP tag. The CLS tag is used to aggregate the features of one or more sentences.

Interval Segment Embeddings

This step is dedicated to distinguishing sentences in a document. Sentences are assigned either of the labels discussed above. For example,

{senti}= EA or EB depending upon *i*. The criterion is basically as EA for even *i* and EB for odd *i*.

Embeddings

It basically refers to the representation of words in their vector forms. It helps to make their usage flexible. Even Google utilizes 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.

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

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

b. Segment Embeddings - It is used to distinguish or we can say classify the different inputs using binary coding. For example, input1- "I love books" and input2- "I love sports". Then after the processing through token embedding we would have

```[CLS],I,love,books,[SEP],I,love,sports ```

segment embedding would result into

```[0,0,0,0,0,1,1,1] ```

``` Input1=0 , Input2=1```

c. 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 the same representation as vectors. For example, "We did not play,however we were spectating."

In the sentence above "we" must not have the same vector representations.

NOTE - Every word is stored as a 768 dimensional representation. Overall sum of these embeddings is the input to the BERT.

BERT uses a very different approach to handle the different contextual meanings of a word, for instant "playing" and "played" are represented as play+##ing and play+ ##ed . ## here refers to the subwords.

BERT Architecture:

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 the 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 a 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.

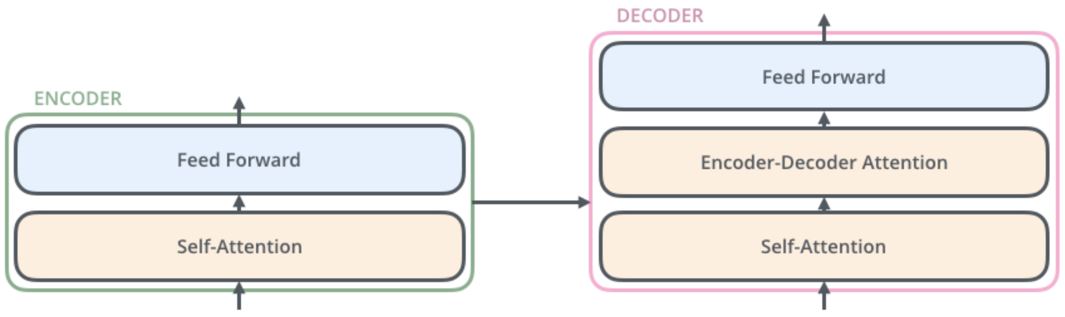


Fig-12

Model BERT:

Install Transformers and Bert for summarization :

```
!python -m spacy download en_core_web_lg
!pip install -q transformers  rouge-score sentence-transformers
!pip install -U sentence-transformers
```

Bert summarization from scratch:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import spacy
from tqdm.notebook import tqdm
import os
import torch
import torch.nn as nn
import torch.nn.functional as F
import numpy as np
from transformers import AutoTokenizer, AutoModel
nlp = spacy.load('en_core_web_lg')
# model not loading in Colab? Restart runtime after installing en_core_web_lg
# get mean pooling for sentence bert models
# ref https://www.sbert.net/examples/applications/computing-
embeddings/README.html#sentence-embeddings-with-transformers
def mean_pooling(model_output, attention_mask):
    token_embeddings = model_output[0] #First element of model_output contains all
token embeddings
    input_mask_expanded = attention_mask.unsqueeze(-
1).expand(token_embeddings.size()).float()
    sum_embeddings = torch.sum(token_embeddings * input_mask_expanded, 1)
    sum_mask = torch.clamp(input_mask_expanded.sum(1), min=1e-9)
    return sum_embeddings / sum_mask

# Creating the customized model, by adding a drop out and a dense layer on top of distil
bert to get the final output for the model.
class SentenceBertClass(torch.nn.Module):
    def __init__(self, model_name="sentence-transformers/msmarco-distilbert-dot-
v5"):#to use other model (like large bert etc.->124 models are available) change the
```

```
model_name parameter to other models available in "https://huggingface.co/sentence-
transformers"
        super(SentenceBertClass, self).__init__()
        self.l1 = AutoModel.from_pretrained(model_name)
        self.pre_classifier = torch.nn.Linear(768*3, 768)#change the dimensions according
to huggingface documentation #hint:dimensions of pooling in the sentence transformer
function, can be seen when scrolled down.
        self.dropout = torch.nn.Dropout(0.3)
        self.classifier = torch.nn.Linear(768, 1)
        self.classifierSigmoid = torch.nn.Sigmoid()

    def forward(self, sent_ids, doc_ids, sent_mask, doc_mask):
        sent_output = self.l1(input_ids=sent_ids, attention_mask=sent_mask)
        sentence_embeddings = mean_pooling(sent_output, sent_mask)
        doc_output = self.l1(input_ids=doc_ids, attention_mask=doc_mask)
        doc_embeddings = mean_pooling(doc_output, doc_mask)
        # elementwise product of sentence embs and doc embs
        combined_features = sentence_embeddings * doc_embeddings
        # get concat of both features and element wise product
        feat_cat = torch.cat((sentence_embeddings, doc_embeddings, combined_features),
dim=1)
        pooler = self.pre_classifier(feat_cat)
        pooler = torch.nn.ReLU()(pooler)
        pooler = self.dropout(pooler)
        output = self.classifier(pooler)
        output = self.classifierSigmoid(output)
        return output
LEARNING_RATE=0.01
from torch import cuda
device = 'cuda' if cuda.is_available() else 'cpu'
extractive_model = SentenceBertClass()
#use this below line if the end-user has any pre-trained extractive summarization model
(.pth file)
#extractive_model.load_state_dict(torch.load(model_path,
map_location=torch.device(device) )) #model_path is path of .pth file
#extractive_model.eval()
extractive_model.to(device)
loss_function = torch.nn.BCELoss()
optimizer = torch.optim.Adam(params =  extractive_model.parameters(),
lr=LEARNING_RATE)
```

```
tokenizer = AutoTokenizer.from_pretrained('sentence-transformers/msmarco-distilbert-
dot-v5')
# tokenize text as required by BERT based models
def get_tokens(text, tokenizer):
  inputs = tokenizer.batch_encode_plus(
            text,
            add_special_tokens=True,
            max_length=512,
            padding="max_length",
            return_token_type_ids=True,
            truncation=True
        )
  ids = inputs['input_ids']
  mask = inputs['attention_mask']
  return ids, mask
# get predictions given some an array of sentences and their corresponding documents
def predict(model,sents, doc):
  sent_id, sent_mask = get_tokens(sents,tokenizer)
  sent_id, sent_mask = torch.tensor(sent_id, dtype=torch.long),torch.tensor(sent_mask,
dtype=torch.long)
  doc_id, doc_mask = get_tokens([doc],tokenizer)
  doc_id, doc_mask = doc_id * len(sents), doc_mask* len(sents)
  doc_id, doc_mask = torch.tensor(doc_id, dtype=torch.long),torch.tensor(doc_mask,
dtype=torch.long)
  preds = model(sent_id, doc_id, sent_mask, doc_mask)
  return preds
def bert_model(doc, model, min_sentence_length=14, top_k=3, batch_size=3):
  doc = doc.replace("\n","")
  doc_sentences = []
  for sent in nlp(doc).sents:
    if len(sent) > min_sentence_length:
      doc_sentences.append(str(sent))
  doc_id, doc_mask = get_tokens([doc],tokenizer)
  doc_id, doc_mask = doc_id * batch_size, doc_mask* batch_size
  doc_id, doc_mask = torch.tensor(doc_id, dtype=torch.long),torch.tensor(doc_mask,
dtype=torch.long)
  scores = []
  # run predictions using some batch size
  for i in tqdm(range(int(len(doc_sentences) / batch_size) + 1)):
    batch_start = i*batch_size
    batch_end = (i+1) * batch_size if (i+1) * batch_size < len(doc) else len(doc)-1
    batch = doc_sentences[batch_start: batch_end]
    if batch:
      preds = predict(model, batch, doc)
      scores = scores + preds.tolist()
  sent_pred_list = [{"sentence": doc_sentences[i], "score": scores[i][0], "index":i} for i in
range(len(doc_sentences))]
  sorted_sentences = sorted(sent_pred_list, key=lambda k: k['score'], reverse=True)
  sorted_result = sorted_sentences[:top_k]
  sorted_result = sorted(sorted_result, key=lambda k: k['index'])
  summary = [ x["sentence"] for x in sorted_result]
  summary = " ".join(summary)
  return summary, scores, doc_sentences
bert_summary=[0]*(end_video)
plot=int(input("0: for no plot, 1: sentence score plot"))
for i in range(start_video-1,end_video):
  if len(data[i]):
    summary, scores, sentences = bert_model(data[i], extractive_model,
min_sentence_length=14, top_k=3, batch_size=4)
  else:
    continue
  if len(summary)==0:
    bert_summary[i]=""
  else:
    bert_summary[i]=summary
  if plot==1 and len(summary)>0:
    sent_lenth = 70
    score_vals = ([x[0] for x in scores] )
    sub_sents = [ str(round(score_vals[i],2)) + " : " + sentences[i][:sent_lenth] + " ..  " for i
in range(len(sentences))]
    with plt.xkcd():
        plt.figure(figsize=(16,10));
        plt.barh(sub_sents[::-1], score_vals[::-1])

print(bert_summary[end_video-1])
```

Summarization layers:

The one major noticable difference between RNN and BERT is the Self attention layer. The model tries to identify the strongest links between the words and thus helps in representation.

We can have different types of layers within the BERT model each having its own specifications:

1. Simple Classifier - In a simple classifier method , a linear layer is added to the BERT along with a sigmoid function to predict the score $\hat{Y}_i$. $\hat{Y}_i = \sigma(W_o T_i + b_o)$
2. Inter Sentence Transformer - In the inter sentence transformer ,a simple classifier is not used. Rather various transformer layers are added into the model only on the sentence representation thus making it more efficient. This helps in recognizing the important points of the document.
    a. $\tilde{h}^l = LN(h^{l-1} + MHAtt(h^{l-1})$
    b. $h^l = LN(\tilde{h}^l + FFN(\tilde{h}^l))$\

where $h^0 = PosEmb(T)$ and T are the sentence vectors output by BERT, PosEmb is the function of adding positional embeddings (indicating the position of each sentence) to T, LN is the layer normalization operation, MHAtt is the multi-head attention operation and the superscript l indicates the depth of the stacked layer.

These are followed by the sigmoid output layer

$$\hat{Y}_i = \sigma(W_o h^L_i + b_o)$$

h^L is the vector for $sent_i$ from the top layer (the L-th layer) of the Transformer.

3. Recurrent Neural network - An LSTM layer is added with the BERT model output in order to learn the summarization specific features. Where each LSTM cell is normalized. . At time step *i*, the input to the LSTM layer is the BERT output Ti.

$$C_i = \sigma(F_i).C_{i-1} + \sigma(I_i).\tanh(G_{i-1}) \quad h_i = \sigma(O_t).\tanh(LN_c(C_t))$$

where Fi, Ii, Oi are forget gates, input gates, output gates; Gi is the hidden vector and Ci is the memory vector, hi is the output vector and LNh, LNx, LNc are there different layer normalization operations. The output layer is again the sigmoid layer.

Training Fine-Tuned DistilBERT Summarizer

This experiment shows an effort to explore the DistilBERT model for extractive summarization tasks. The model was trained using the Adam optimizer with a learning rate 2 × 10−3, with a batch size of 3000, and a 10% dropout. shows the ROUGE scores of the fine-tuned summarizer trained on 30,000 epochs on the CNN/DM dataset. Every 10,000 epochs, we save the model weights and generate the ROUGE-1, ROUGE-2, and ROUGE-L validation scores. The performance of the model before 10,000 epochs was relatively low, so we limited saving the model weights by starting with the 10,000 as our first checkpoint. The results record the output of the evaluation algorithm, where the summary is evaluated by computing the value of precision, recall, and F-score. Average R represents the average weight of recall, Average P represents the average weight of precision, and Average F represents the average weight of the F-measure. The training time taken for a DistilBERT summarizer was around 25 min per 1000 checkpoints on a Google GPU session.

The DistilBERT-uncased model was used in this experiment. It contained six transformer layers, 768-hidden layers, and twelve attention heads. After tokenizing the input texts and converting the tokens into input ids, they were padded and fed into the DistilBERT model for the multi classification task.

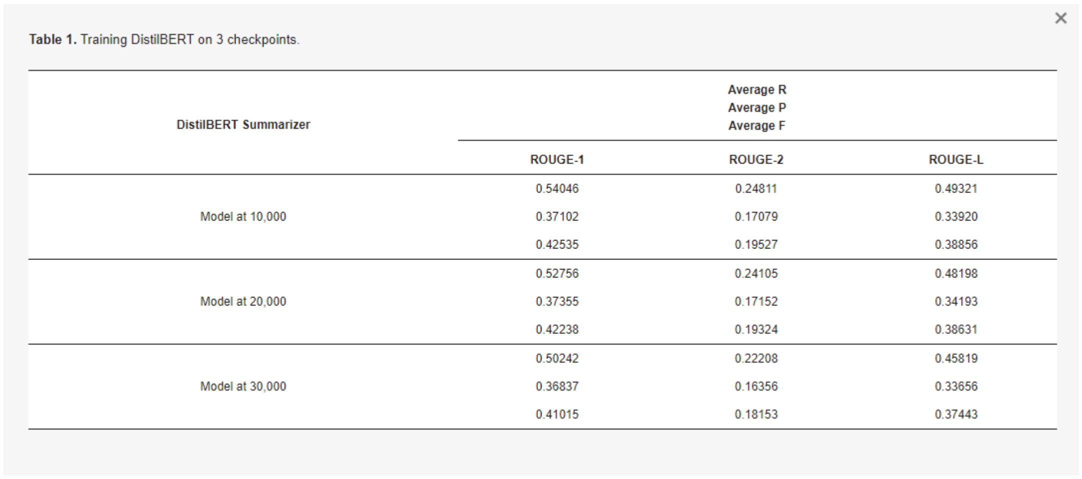
Evaluation Metric:

Rouge Score:

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially a set of metrics for evaluating automatic summarization of texts as well as machine translations.

It works by comparing an automatically produced summary or translation against a set of reference summaries (typically human-produced). Let’s say that we have the following system and reference summaries:

If we consider just the individual words, the number of overlapping words between the system summary and reference summary is 6. This, however, does not tell you much as a metric. To get a good quantitative value, we can actually compute the **precision** and **recall** using the overlap.



×

Table 1. Training DistilBERT on 3 checkpoints.

DistilBERT Summarizer	Average R Average P Average F		
	ROUGE-1	ROUGE-2	ROUGE-L
Model at 10,000	0.54046	0.24811	0.49321
	0.37102	0.17079	0.33920
	0.42535	0.19527	0.38856
Model at 20,000	0.52756	0.24105	0.48198
	0.37355	0.17152	0.34193
	0.42238	0.19324	0.38631
Model at 30,000	0.50242	0.22208	0.45819
	0.36837	0.16356	0.33656
	0.41015	0.18153	0.37443

Table 2

Simply put, recall (in the context of ROUGE) refers to how much of the **reference summary** the **system summary** is recovering or capturing. If we are just considering the individual words, it can be computed as:

$$\frac{number\_of\_overlapping\_words}{total\_words\_in\_reference\_summary}$$

In this example, the recall would thus be:

$$Recall = \frac{6}{6} = 1.0$$

This means that all the words in the **reference summary** have been captured by the **system summary**, which indeed is the case for this example. Voila!

This looks really good for a text summarization system. But it does not tell you the other side of the story. A machine generated summary (system summary) can be extremely long, capturing all words in the reference summary. But, many of the words in the system summary may be useless, making the summary unnecessarily verbose.

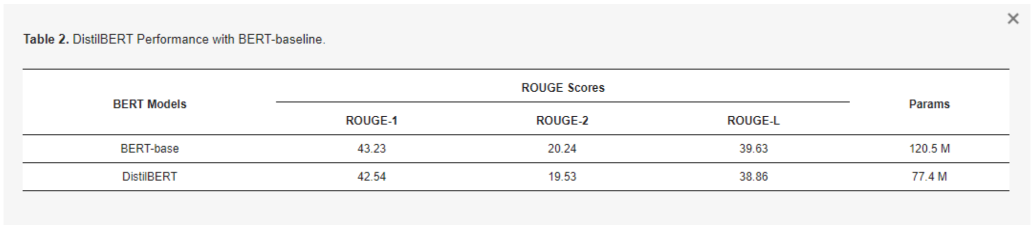


Table 2. DistilBERT Performance with BERT-baseline.

BERT Models	ROUGE Scores			Params
	ROUGE-1	ROUGE-2	ROUGE-L	
BERT-base	43.23	20.24	39.63	120.5 M
DistilBERT	42.54	19.53	38.86	77.4 M

Table 3

This is where precision comes into play. In terms of precision, what you are essentially measuring is, **how much of the system summary was in fact relevant or needed**? Precision is measured as:

$$\frac{number\_of\_overlapping\_words}{total\_words\_in\_system\_summary}$$

In this example, the Precision would thus be:

$$Precision = \frac{6}{7} = 0.86$$

This simply means that 6 out of the 7 words in the system summary were in fact relevant or needed. If we had the following system summary, as opposed to the example above —

System Summary 2:

the tiny little cat was found under the big funny bed

The Precision now becomes:

$$Precision = \frac{6}{11} = 0.55$$

Now, this doesn't look so good, does it? That is because we have quite a few unnecessary words in the summary. The **precision** aspect becomes really crucial when you are trying to generate summaries that are concise in nature. Therefore, it is always best to compute both the **precision** and **recall** and then report the **F-Measure**.

- If your summaries are in some way forced to be concise through some constraints, then you could consider using just the **recall,** since precision is of less concern in this scenario.

- ROUGE-N, ROUGE-S, and ROUGE-L can be thought of as the granularity of texts being compared between the system summaries and reference summaries.
- ROUGE-N — measures **unigram**, **bigram**, **trigram** and higher order n-gram overlap
- ROUGE-L — measures **the longest matching sequence** of words using LCS. An advantage of using LCS is that it does not require consecutive matches but in-sequence matches that reflect sentence level word order. Since it automatically includes longest in-sequence common n-grams, you don't need a predefined n-gram length.
- ROUGE-S — Is any pair of words in a sentence in order, allowing for arbitrary gaps. This can also be called skip-gram concurrence. For example, **skip-bigram** measures the overlap of word pairs that can have a maximum of two gaps in between words. As an example, for the phrase *"cat in the hat"* the skip-bigrams would be *"cat in, cat the, cat hat, in the, in hat, the hat".*

For example, **ROUGE-1** refers to the overlap of ***unigrams*** between the system summary and reference summary. **ROUGE-2** refers to the overlap of ***bigrams*** between the system and reference summaries.

Let's take the example from above. Let us say we want to compute the **ROUGE-2 precision and recall** scores.

System Summary:

the cat was found under the bed

Reference Summary:

the cat was under the bed

System Summary Bigrams:

the cat, cat was, was found, found under, under the, the bed

Reference Summary Bigrams:

the cat, cat was, was under, under the, the bed

Based on the bigrams above, the ROUGE-2 recall is as follows:

$$ROUGE2_{Recall} = \frac{4}{5} = 0.8$$

Essentially, the system summary has recovered 4 bigrams out of 5 bigrams from the reference summary, which is pretty good! Now the ROUGE-2 precision is as follows:

$$ROUGE2_{Precision} = \frac{4}{6} = 0.67$$

The precision here tells us that out of all the system summary bigrams, there is a 67% overlap with the reference summary. This is not too bad either. Note that as the summaries (both system and reference summaries) get longer and longer, there will be fewer overlapping bigrams. This is especially true in the case of abstractive summarization, where you are not directly re-using sentences for summarization.

The reason one would use ROUGE-1 over or in conjunction with ROUGE-2 (or other finer granularity ROUGE measures), is to also show the fluency of the summaries or translation. The intuition is that if you more closely follow the word orderings of the reference summary, then your summary is actually more fluent.

ROUGE Method

```
!pip install rouge-score

from rouge_score import rouge_scorer
def ROUGE_score(input_text,model_name,video_no):
  scorer = rouge_scorer.RougeScorer(['rouge1','rouge2','rougeL'], use_stemmer=True)
  scores = scorer.score(reference[video_no],input_text)
  print("Rouge scores for the model:"+model_name+" for the
video:"+str(video_no+1))
  print(scores)

model=input("Enter bert only ")
for i in range(start_video-1,end_video):
  if model=="bert":
    if bert_summary[i]!=0:
      ROUGE_score(bert_summary[i],model,i)
    else:
      print("No text to generate summary")
  else:
    print("Error: Model not found")
```

Code:

Module Download:

```
!pip install moviepy
!pip install SpeechRecognition
import moviepy. editor as mp
!pip install pydub
```

Dataset Loader (Video to Audio):

```
start_video=int(input("Enter starting video number"))
end_video=int(input("Enter ending video number"))
for i in range(start_video-1,end_video):
my_clip = mp.VideoFileClip(str(i+1)+".mp4")
my_clip.audio.write_audiofile(str(i+1)+".wav")
```

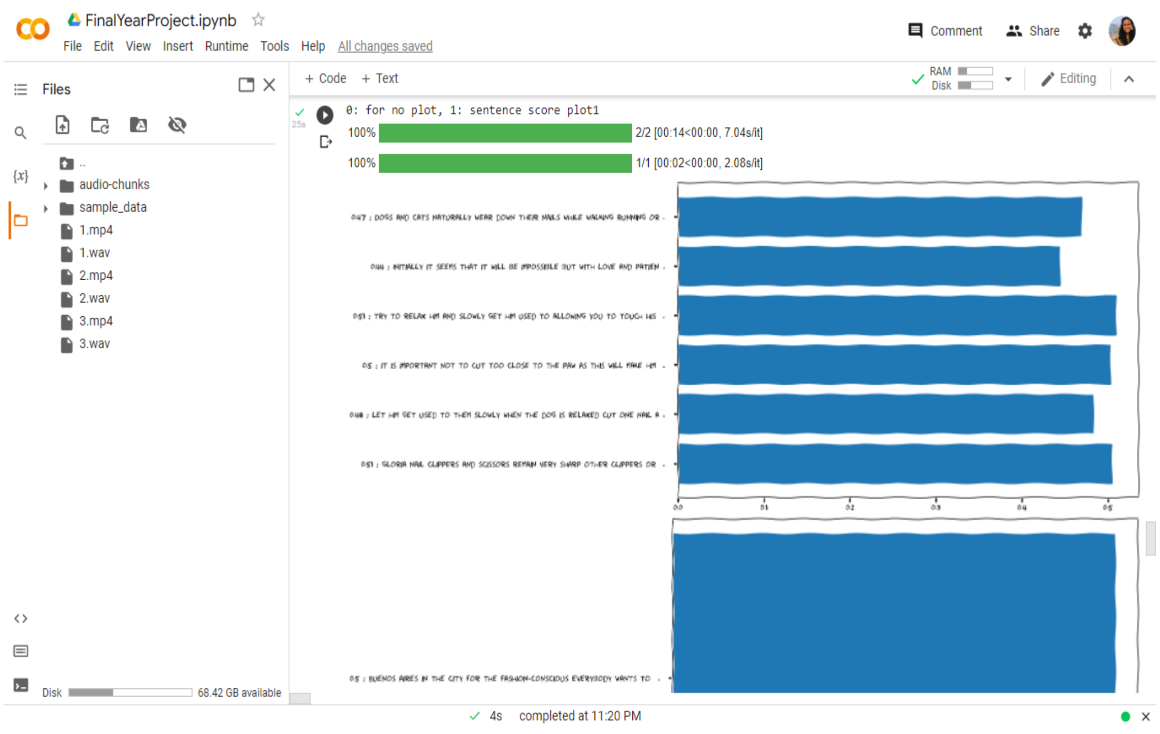
Audio to Text:

```
import os
from pydub import AudioSegment
from pydub.silence import split_on_silence
import speech_recognition as sr
# create a speech recognition object
r = sr.Recognizer()

# a function that splits the audio file into chunks
# and applies speech recognition
def get_large_audio_transcription(path):
    """
    Splitting the large audio file into chunks
    and apply speech recognition on each of these chunks
    """
    # open the audio file using pydub
    sound = AudioSegment.from_wav(path)
    # split audio sound where silence is 500 milliseconds or more and get chunks
    chunks = split_on_silence(sound,
        # experiment with this value for your target audio file
        min_silence_len = 500,
        # adjust this per requirement
        silence_thresh = sound.dBFS-14,
        # keep the silence for 1 second, adjustable as well
        keep_silence=500,
    )
    folder_name = "audio-chunks"
    # create a directory to store the audio chunks
    if not os.path.isdir(folder_name):
        os.mkdir(folder_name)
    whole_text = ""
    # process each chunk
    for i, audio_chunk in enumerate(chunks, start=1):
        # export audio chunk and save it in
        # the `folder_name` directory.
        chunk_filename = os.path.join(folder_name, f"chunk{i}.wav")
        audio_chunk.export(chunk_filename, format="wav")
        # recognize the chunk
        with sr.AudioFile(chunk_filename) as source:
            audio_listened = r.record(source)
            # try converting it to text
            try:
                text = r.recognize_google(audio_listened)
            except sr.UnknownValueError as e:
                pass
                #print("Error:", str(e))
            else:
                text = f"{text.capitalize()}. "
                #print(chunk_filename, ":", text)
                whole_text += text
    # return the text for all chunks detected
    return whole_text
```

Dataset Loader (Audio to Text):

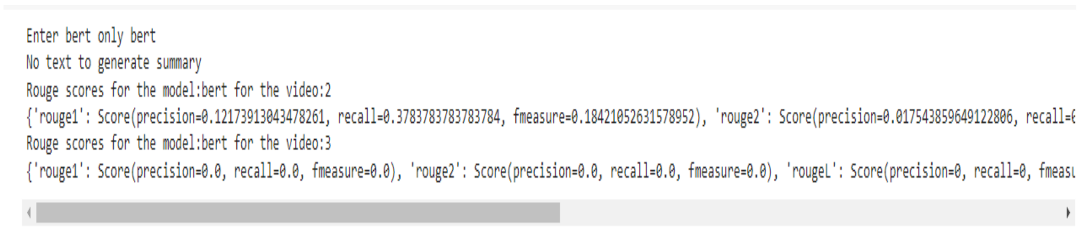
```
data1=[0]*(end_video)
for i in range(start_video-1,end_video):
  path=str(i+1)+".wav"
  data1[i]=get_large_audio_transcription(path)
  print("\nFull text "+str(i+1)+": ",data1[i])

for i in range(start_video-1,end_video):
  if data1[i]==0:
    data1[i]=""
```

Result:

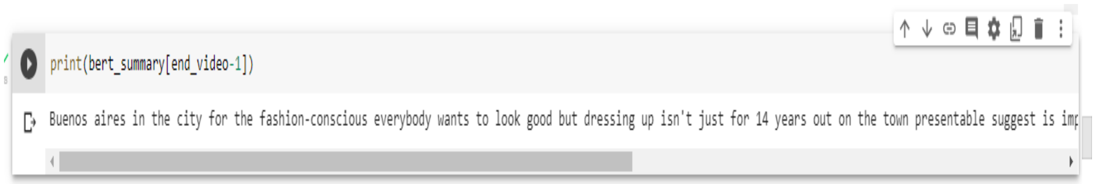


```
0: for no plot, 1: sentence score plot1
100% 2/2 [00:14<00:00, 7.04s/it]
100% 1/1 [00:02<00:00, 2.08s/it]
```

FinalYearProject.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Files
..
audio-chunks
sample_data
1.mp4
1.wav
2.mp4
2.wav
3.mp4
3.wav
Disk 68.42 GB available
0.47 : DOGS AND CATS NATURALLY WEAR DOWN THEIR NAILS WHILE WALKING RUNNING OR
0.44 : INITIALLY IT SEEMS THAT IT WILL BE IMPOSSIBLE BUT WITH LOVE AND PATIEN
0.51 : TRY TO RELAX HIM AND SLOWLY GET HIM USED TO ALLOWING YOU TO TOUCH HIS
0.5 : IT IS IMPORTANT NOT TO CUT TOO CLOSE TO THE PAW AS THIS WILL MAKE HIM
0.48 : LET HIM GET USED TO THEM SLOWLY WHEN THE DOG IS RELAXED CUT ONE NAIL A
0.51 : GLORIA NAIL CLIPPERS AND SCISSORS REMAIN VERY SHARP OTHER CLIPPERS OR
0.0
0.1
0.2
0.3
0.4
0.5
0.5 : BUENOS AIRES IN THE CITY FOR THE FASHION-CONSCIOUS EVERYBODY WANTS TO
4s completed at 11:20 PM



```
Enter bert only bert
No text to generate summary
Rouge scores for the model:bert for the video:2
{'rouge1': Score(precision=0.12173913043478261, recall=0.3783783783783784, fmeasure=0.18421052631578952), 'rouge2': Score(precision=0.017543859649122806, recall=
Rouge scores for the model:bert for the video:3
{'rouge1': Score(precision=0.0, recall=0.0, fmeasure=0.0), 'rouge2': Score(precision=0.0, recall=0.0, fmeasure=0.0), 'rougeL': Score(precision=0, recall=0, fmeas
```



```
print(bert_summary[end_video-1])
```

```
Buenos aires in the city for the fashion-conscious everybody wants to look good but dressing up isn't just for 14 years out on the town presentable suggest is imp
```

Model Evaluations:

reference = [
"",
"Dogs and cats naturally wear down their nails while walking running or scratching trees however some animals don't and this can be a problem cutting the nails can be a difficult problem for both of you.",
"",
"",
"",
"",
"You coming tonight to call the breakdown service. Honest toyota the air compressor for inflating the tires. So take the lead from the compressor and plug it into the cigarette lighter or 12 volt socket inside your car. If the tire isn't inflating pro play troy rolling the car forward a few meters to help the seeding spread around if that doesn't work you may still need to cool the breakdown service.Take the links to find out best boy toys readout going to saving money on tires and find out which breakdown companies we recommend.",
"There's a slogan is charging an electric car is charging the energy of the earth. This means that you get to play a role in reducing greenhouse gas emissions by driving eco-friendly vehicles. The minister of environment seoul metropolitan government in korea and government corporation is established infrastructure for electric cars. Electric cars and infant explain quality improvement. The 2012 electric vehicle and smart curry salvation was held in october 17th to display eco-friendly electric vehicles and related equipment. Can you tell a funny pictures of electric vehicles related industries including two wheeled vehicle of material charging system dissipated into this is a business fair that is a deficit exchanges between perfect.",
"'You know in small vehicle tires just with a couple screwdrivers or something like that. First of all there's three kinds of bead styles the first feed style is the tube style tire. Lock the rubber beaded attire. Put the rubber part of the tire. On your face of your jack and put it underneath the high vehicle or something heavy. I like to set the nipple for the air phil least to that side or that side and start. I start to unpeel it was still keeping the pressure on the other side to keep the bead into the deep groove. Pinche rim a little bit with a vice grip

right there. If you have a tube type tire. My tire had a tube in it that's the way i wanted to be but when you're installing youtube didn't put the old one back in tubes very often head of have a right or left side. So that means a nipple ism moroch center. We can't get ahold of nipple anymore.
Cut the valve stem over there a vice grip opposite to it now i'm going to stand as best as i can hear sometimes it takes two people get that lip into the deep groove. Especially on quad tires to reseal the bead takes a lot of pressure because it's at high pressure locking be because it's a low pressure tire. Quad tire to a shop and had them user machine to break the deed. Atv tires or quad tires are very low pressure so they don't use air 2.
They have this kind of high-pressure locking bead because the tires are low pressure and if you're doing hard cornering.'",
"I like really hit the gym really thorough that night. About about 7 years ago and since then we've other people will start to join us with your grass so i was with larenz made out of people. Parkour is a discipline of training your mind and body to overcome obstacles and i'll be trained s5 running jumping climbing vaulting balancing things like that you want to try a new job for you not just you know just physically job at the same time. Space. Rolls or something that can save your life or your knee i tore two muscles in my neeks i fell off a building and i didn't roll my leg buckled to the side but had i actually rolled i would've been okay. Poor decisions so. And if you ever seen people do the dive rolls that's going to be teaching you too so we'll start with rolls on the ground fall rolls and then dive world. So coming through. I'll just go here. How many people. My compaq.",
"I totally jacked up his bum. I think i feel a little bit better or i'm just on so many. Jugs. It doesn't hurt. Bentley a bath and a trim cuz i don't like the way that he came out. Going to the groomer. Last week i feel like she doesn't deserve him. And i liked it so imma do this. Myself. Give him a bath bomb out i ordered the wahl bravura clipper set and not should be delivered any minute now by fedex. But figured i would just wash. I'm plum out you know he's already kind of clean. But everything i read about using clippers and clipping dogs. Mohanty. And i. Think i kind of desensitized him to the hair dryer. In his face. Cuz i just rapid fired kimball at him. When i was using the blow dryer in the parts that he didn't like cuz i read in a job training book. How to desensitize a man i think it worked a little bit i was able to actually blow out his head and it's not all wavy. I mean i could have blown out better but you know what i think i actually did a pretty good job even though your face is a little dirty. So now i'm just going to let him play. And we're waiting for our clippers to be delivered you not going to go to the groomer while your mama's going to do it. It's really not that bad probably need a little more. Tear. Say i look good mama you did good job. Oh my god i like totally jacked up his butthole. And i'm just going to i don't even know what i did i just kind of went along with maltese obsession. With what she didn't her video has a couple spots that aren't perfect. But it's my first time it's okay and also he just got blown out so when his hair grows back. And his natural oils come back it'll blend a lot better. You my little puppy for everything a do you want to do i want his legs a little smaller. And i

kind of don't make fun of me like his but i did way too much. Show me your butt. I did too much on your but i'm still having my bubbies.",
"",
"A hydrocodone. Broccoli. Ncg. Is a fascist. Am i. Gorgeous. You don't have to pay in d.c. peanuts. Una cancion. Music opera. La liga. Plow on at the cinema. Aaa. Mclaren. If so, don't even with you. L'appel du vide downfall squally mobile. Gwtw steal a security officer during a table. Indonesian police hawaii.",
"Hangout.com. Optimus. Dr. jacarilla according to the admiralty personality moves and all the stuff. I needed that this one sees the lady she's a lady. What's this all about.",
"As extension educators we get frequent phone calls concerning paper wasps and their nests. And as a homegardner you'll probably run into these things at one point in time now. This is a little bit of a problem because paper wasps are actually a beneficial insects are out collecting. And. Usually they're found in the eaves of homes and structures. And they can be knocked down with a shovel or other implements. And then taken away and thrown away. It's a good idea to go ahead and remove it whenever you possibly can. And that can be done chemically. Their products available that are specific to wasp and hornet control. And wait until evening. Spray that product so that all the wasps are on the nest.",
"What's the number one question people ask about six gear bikes. How do you stop. How to stop on a fixed gear bike is the most frequently asked question from mom's pedestrians or other people who can't believe you're riding a bike with no brakes. Are lots of techniques to stop a fixed gear bike. The one we recommend is by using your friend break if you're going to be a real g and you got rid of the front brake here's some tips that will help you slow your roll. First. The skip stop. This is a technique where you skip on your rear wheel gradually slowing you down. When you land the bike skips and reduces speed. Refugee together and you just save yourself from crashing and that parked car. Hey what's up cuz i'm new york and if you're wondering how to do a skip stop i got your answer. First of all you want to get up to speed. You don't want to rise out of your saddle way to take weight off the back wheel. Then using your feet you're literally going to pick up the back wheel which are basically stopping in midair. Also keep in mind whatever foot is forward if you actually what you put up with and whatever for his back is the one you put you down with. Basically those are the steps to do a skip stop. Skip. is where you lock up your rear wheel causing your bike to skid and slow down. This is a fast way to stop but it also eats up your rear tire. In order to skid you need to be going somewhat fast. Rose on your side of take weight off the back wheel. Pull up with your left pedal by pushing down the right pedal keeping in mind leah pedal should be horizontal. If you need extra leverage you can always push your leg against the top two with a frame and that's pretty much how you doing skin. The easiest way to slow down is to apply back pressure to your pedal. It really looks like nothing but pull back on the pedal that they rotate and you'll slowly cruise to a stop. When ready fixed gear it is very important to have some kind of a retention on your pedals. You can either use straps or clips either way you need some kind of flirtation to perform any of

the stops we feeling today. Do you have any questions or comments we want to hear from you. So hit us up on our twitter at pure fix tv. Or drop at the comment right down below. Do the next week for a new episode on pureflix tv.com. Tell next time i'll see you in the street.",
"Do you want some cat food. Play some other kind of food we got some dog food. Can cats eat dog food didn't actually eat dog food. You say what do you say yes or no yes or you think silly dog food yes.Y'all trying some dog food. Why does dog food. Can you eat that dog food. No they're watching this at our cat is really picky there's a lot of cat food that she won't eat.This is kind of strange to see her eating dog food but she really likes this dog food so i might have to buy this again. We just fed our cat and dog food. The dog food. Dog food.Will mickey survive the night she ate dog food and didn't get sick. You can see that you really liked your dog food. So yes cat can eat dog food yay.",
"",
"",
"",
"Initial film i'm going to explain how to repair a tire using the compressor and sealing kit provided instead of vehicles. Having a kid in your vehicle increases space and reduces weight. Play the parking brake to prevent the vehicle moving accidentally. Make some old passengers leave the vehicle and move to a safe area.The kick cannot be use of the wheel room is damaged or the vehicle has been tripping with an under-inflated or flat tire for an extended. If you don't use the punch repair kit. You will need to arrange vehicle recovery by contacting skoda roadside assistance. The kit comprises of a compressor unit and the sealing kit.Valve to spare tire valve and warning sticker in the back. Now we remove the valve cap from toy valve itself. Puppet into the valve. Now we get the fluid and attach the philips hue to the time about.I just met a where the time valve is located when the vehicle is stationary. As a suitable work its way around the tire for the vehicle is driven. Disconnect the philips hue from the tiebout and refit the valve insert. Now we take the compressor. So please see your local skoda retailer or tire retailer immediately who can repair or replace the damaged tire. Please do not hesitate to contact your local skoda retailer.",
"I'm jeff 92 malin welcome to my kitchen today we are making something super delicious is always we're making a reuben sandwich with what. Sauerkraut you can see the recipe on one of my bootleg tip videos we have some homemade pickles again another wonderful bootleg tip video you can catch swiss cheese rye bread we're making a very quick easy russian dressing and you wonder why i have a side of. Rye bread that you can use any type of bread you want you don't have to use where i read even have to use dark rye bread really quick cuz i can do it that way get on your play next going to start building i like that so the fat stays on player one.",
"A truck at 12 asleep deliver something is watches that. And the truck. The truck ran off the road nearly tipped over went down the galley. Should the truck.Cuz there's a truck coming that's going to try to pull us out. We got another big truck. To pull the truck that

markets. In the rat.Moxtek truck the back truck. Glozell raincloud said that gabe's there's a listing for soul as you can see the gun to employ the palfinger indominus sounding names piece of machinery responsible without whom the situation with niva and happens not stuck truck. Rescue truck. Finger pal and mock stuck truck peel-off bowling branch.",
"Master grooming little bedding for them to make sure they're comfortable and so they tend to lay down and rather than sit up waiting. Their tables a lot of places have tables that don't lower. So here if you want to demonstrate rosie. We just take a moment to. In a comfort calm him down a little time spent cards and grooming. Easier. Camino them well we know their habits. You know simon things like that grain-free gluten-free you know. Serious issues for dogs i mean a lot of commercial foods have. Fillers that food they can't process their body can't process i need to come hill.",
"The granddaddy of bikes. Either you can do it or you. Today we're going to teach you. How to wheelie. Hey guys. Doing a wheelie something with all designer do since our very first bike ride. Go ahead and do it man. First you comfortable pulling your front wheel up. Then push with your lead foot and lean back. Practice is running around empty parking lot. But don't use a strap just in case you have to bail. Wright ford and pull the front wheel up. Once you push off your leave foot lean back and find your balance point. Once you're locked in remember the combination of shipping your body weight. And the speed that you are pedaling.",
"Cadillac cts. Your car's not worth a lot to look at it. Got any battery. He complained. How about your flat tire you could definitely say that. Maybe you could. Wonder what happened. Are waiting for becca to get here with her jack from her car. My muscle man. We've been having fun taking video. What's happiness. Show us happy face. Have you ever done this before they feel manly. I'm going to eat another piece of chocolate. Well they're figuring out what to do. Bad boy allergies. I can't help song. Ladies take note. Change a tire.. I feel about steinbach your head. I was in a bad jumper back tonight. Now the key to being successful in life it's always have a stent. The best that's a spider. Now it did. It's asking your dad to see if i'm if i did it right.",
"One dog getting a new leash on life at his heaviest ob a five-year-old dachshund weight a whopping 77 lb that or foster parent he's going on a strict diet. When you heard about a 77-pound dachshund. We are just putting them on a high-protein high-fiber low-energy diet and he's able to eat we have to meet we're feeding about 2 cups a day. Yeah i find it to the treats and it's benji it just like people to binge-eating relation between overweight pet owners and overweight pets.I don't know if there's going to study done to look at that butt in my clinical practice yes i see a lot of dogs were overweight a new owners tend to be overweight as well.",
"Okay so you finally invested in a bike and now you need to know how to keep it safe and we're going to show you how. Is getting rid of any quick release lever. Quick release levers on your wheels and seatpost make removing them easy. It only takes a few seconds to remove a quick-release front wheel or saddle you should replace any quick release levers

with allen bolt or 15 mm nuts.A bike that looks fresh and clean the tracks would be c. Your bike won't ride me different but i'll be much less attractive. Don't ever leave your bike unlocked. It's not a lock your bike using you lock.Alright thank you so much that you guys got to remember to get the right lock & lock up your bike if you have any comments please comment below or hit us up on our twitter.",
"",
"",
"I get very irritated with food that when you bite into it, it Burns your mouth. I think the toppings make it unique. I think just the way the chicken itself plays with the honey, the sweetness and that, the spiciness and the Tabasco, the crunch of the cucumber and the radish in it, I think it's a unique sandwich. My philosophy on food, I think ultimately, no matter what, it boils down to one common principle, which is the intention is to create a product that we're proud of that makes the person eating it smile.This is a damn good sandwich. We wanted to do what we call an American Bistro here because we're in America. I'm American. Part of my family is American.And then within that offer a few inflictions of French food that we love we are trying really hard to make connections with our guests.",
"",
"",
"",
"Basically whipping the bike is. Basically with the bike out sideways so that the goal would be to get the bike either you know parallel to the ground. Just kind of chicken the bike out. Some people get confused as to what actually goes on in the web and how you get the bike back but when you do the whip correctly the bike and the momentum of your bike will actually bring you back around and straighten you out before you hit the ground.Above your ability level. Which will bring the bike. Your bike back up straight. Straight.Bike all the way back. The weapons basically just star. I'll bring the bike around. You got the bike out.The key is just to stay within your ability level. Use your your body and your bike. And let the momentum carry the cross. The weapons by far the most impressive looking tracking.Let the bike fall out there in the momentum. Like i said the weapons a lot of fun.",
"Melissa laurel sheen parade it's a great part of mount holyoke tradition where we carry the screen laurel and feels like. Side-by-side we look forward to doing this this every year and we we plan. Months in advance to make sure that we have all the roping. Signs the signs are always creative. They're always appropriate to the era in which we went to college and it's great fun to see the seniors 13 is our granddaughter class. The seniors and. Adopting. Granddaughters. It's part of the tradition it really doesn't matter. Whether they are the class of.",
"We are inside an outbuilding on a farm here in bisbee where this massive killer bee hive was located. Big enough we're told to hold more than 250,000 killer bee. It is the attack of the killer bees sound like a horror flick this is very real and animal farm located off highway

80 in bisbee freelance videographer alice ignore a shot this unbelievable footage 1000-lb bore under attack. Don't like i've been boxing without gloves but everyone else runs this guy comes buzzing in reboot called himself the killer bee guy.",
"New details tonight about a train accident we first told you during our four newscast take a look for my cvs atlanta sky i. You can see the accident. Steve is this accident an isolated event. You know stephanie it's really not and the driver of that vehicle is thankfully able to get out of her car with plenty of time to spare before that train came along another problem maybe the way that the asphalt was late is about a 6 inch drop between the surface.And the railroad ties and neighbors hearsay cars. Locomotive push the vehicle 1000 feet before finally stopping. All too often we've been having truckers get stuck out here vehicle getting stuck on the tracks i'll help maybe 45 people get off the track. Apart from several collisions in the past year there have been 22 documented incidents of cars.",
"",
"Okay let's tell him to tell us what happened. Okay well. Second i'm driving and. Driving down the pass through following the dog. Yeah. Emma's car kinda got stuck in the snow were you two young lovers doing in the back forty acres anyways all alone. Driving to run out of gas. I think you're stuck for good. Liquid ammonia merryweather. Well that's pretty bad. There's no use taking this one out. We'll have to leave and help the indians don't destroy it. Everybody know we can't get it out. The snow is on the frame. We need 50 more kids like you and we might have hoped but. Great great wolf. Waffle show us the way. Garage rock. Gunners mouse and i think it's better. The front window.",
"",
"Fresh avocado we had some chipotle pepper some jalapeno loopsy jalapeno peppers queso blanco you can use oaxaca cheese or any type of penalties that you wanted it can't find that use mozzarella we have finley sliced chicken breast. And a couple of eggs all right let's get cooking going to do a little prep in the morning to get the food over the stove first thing first going to get my tomatoes cut i like to use roma tomato just because i like roma tomatoes tomatoes you can use any type of tomatoes or tomato is a tomato. Just enough for the sandwich plate. Cheese.Some cheese from mexico that don't melt. You find one that you like and use it and avocado i don't like my avocado too mushy i like it just right. Next yes you got it we are going to bread or chicken okay this is an easy process first what i do is i get my eggs in a bowl with it a little bit and then we're going to dip or chicken into the eggs and then into the breading and that's it. As always get extra fat into my meals going to do that i'm going to butter my bread and make it nice and super crispy.Oil claim is nice. Indian goes my bread meanwhile. Get some beans into the pan then just do a nice little heat smash so i have some how you say lately fried beans i think about one more minute for my chicken and it's ready to come out you can do disdain recipe with steak like a new aneja you could do with fish if you wanted to or you can do it and make the same book a sandwich that's right you can't give it a little more smashy smash. Okay we are now going to do something that takes a lot of concentration extremely serious we're going to build a torta

they have are black beans already at the bottom layer that's right we had some cheese.I'm going to do some tomatoes. Are you go tomatoes. You could cut it up into little slick little time time time time time time time flavors but why i was going to make this sandwich the chicken charger we are going to add pickled jalapenos. It's going to lay it across okay i got my lettuce on i got my jalapeno peppers.Avocado going on there.",
"Play the undertaker bees guard bees nursery bees. Are we go over here and open up a hi-c with the bees are doing now you going to get in trouble those fees by open up that high. Is it working on a honey super now. Well i produce 3/4 lb of honey low and how long does it take for them to do that that depends on the honey flow and how many bees you got in the hive how much honey you're going to get so if you got a full two boxes of honey bees in the bottom why you can in a good honey flow fill one of these is there any way of knowing that approximately 60,000 and august they build up during the summer.We sometimes have fireweed honey or boo boo berry honey or the blackberry. Honey from a local beekeeper if you have allergies.",
"",
"",
"We're taking smartphone a spaceship technology on a motorcycle platform with the safety of a car. They see it there is a robot motorcycle. Itself oz and i really creases like you know your safety and your range. And excitement of riding a motorcycle. Jordan miles for charge to two-wheel drive. Wheel hub motor electric vehicle. Stare like a car. You said it like a car pedals like a car but. You know the robot side it takes your steering input. And eileen he returns a lot like a jet fighter. So it's you can see full like tronics in here. And this is actually a real jar all right here so take a look at this the gyroscopes inside. Basically output around 1300 ft lb of torque. So you basically the baby elephants knock it over it's a lot like a cmg or control momotaro. The international space station and most satellites use the same fry orientation technology we've just kind of taking that and you don't need a commercially available their servos and there a gyro motors there traction motors on but sensors like an imu ir sensors. There is actually a temperature and heat sensors. Myriad of santa was actually all feed data that has to be processed. And through that process commando to the gyros to tilton lean. On the vehicle to keep it balanced or either to lean into a turn. All i know it's the very dependent lee base on our processing system. The project that we have now uses to i7s at our end product we should be using something that could power a small ramon today. What hurdles in our development is having the vehicle actually be more aware of its surroundings do it but also having good sensory perception. I'm of the road of any kind of external influences or disturbances. And having that all get computer to processed till out for stability. All scenarios. On the heads-up display you see battery life range. How much left do you have in your possession budget. You kind of get your artificial horizon. And also you know being expensive rolling smartphone so all the information on your smartphone will be on a side panel upgrading the hardware can be done pretty easily. Vdara taxi, in the battery pack coming until tomorrow storage unit. So you can upgrade the

gyro output you can upgrade the battery size you know or or both. So if you want a faster a more robust or stabilized you know c1 it's pretty easy to upgrade that the chassis made out of steel we could be using us some kind of composite body. For the interior swell neil parts count where 1/10 of a part of a car. Hi car seat typically have run 24000 parts if you drive an audi at 40,000. William 2239. So it's freezing main factor that's why we can make it affordable. In our audience urban professionals tech junkies you know they have to have the first thing. They can mute their professional life they still want to have fun and young. They want to have first best thing but at affordable price.",
"The dogs develop dirt and debris and discharging their ears it has to be taken out and so will you solutions like the ear cleansing solution to help break up debris and dirt to help change the ph of the ears to help make it where it's not a hospitable environment for the bugs. One thing to remember is that if you're treating and cleaning ears the same time need to clean the ears 30 minutes prior to treatment. Couple tips on cleaning ears first of all you want to grip the ear at the base of the year but not so much that were causing pain.",
"I'm sure you've learned your gmc really lives up to its never-say-never reputation. One that will continue for a long time. Get there comes a time when the tires are gmcs came with need to be replaced. Gmc recommends rotating your tires every time you change your oil or 7,500 miles. Your certified service experts will use a tread depth measurement tool like this to see if your tires need replacing. Cuz of course riding on bald tires is definitely a safety hazard. You also need proper tire balanced and unbalanced or improperly balance tire and wheel and make your car vibrate as well as kraton even tread wear. Bent wheels were misaligned suspension can also produce uneven tread wear and cause tires to lose air so watch out for those sneaky potholes and curves.",
"Everybody what's going on it's josh beats here in this is the first episode of poor man's meals so let's go ahead and get right into it. Not turd ferguson. Hey turd ferguson. Yeah we won.What should we do i've got some hot links i've got some ketchup on got some onions what do you think. Obviously myself is being stubborn or turd ferguson. So bottom line is this we're going to be making some spicy sausage sandwiches. Onion.Peel that onion. Cutting onions. Play kid off onions are layers. Layers of onion.Spicy sausage. Spicy hot links. We're going to cook this for about 45 minutes what's that going to do to catch up. Oh yeah kitchen.Extra 45 minutes right here on the stove. I hope you enjoyed join me next time for another episode of poor man's meals until then.",
""
]

ROUGE Method:

```
!pip install rouge-score
from rouge_score import rouge_scorer
```

```
def ROUGE_score(input_text,model_name,video_no):
  scorer = rouge_scorer.RougeScorer(['rouge1','rouge2','rougeL'], use_stemmer=True)
  scores = scorer.score(reference[video_no],input_text)
  print("Rouge scores for the model:"+model_name+" for the video:"+str(video_no+1))
  print(scores)
model=input("Enter bert only ")
for i in range(start_video-1,end_video):
  if model=="bert":
    if bert_summary[i]!=0:
      ROUGE_score(bert_summary[i],model,i)
    else:
      print("No text to generate summary")
  else:
    print("Error: Model not found")
```

Activation Function:

Activation function decides whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.
There are 7 Activation functions:

- Linear or Identity Activation Function.
- Non-linear Activation Function.
- Sigmoid or Logistic Activation Function.
- Tanh or hyperbolic tangent Activation Function.
- ReLU (Rectified Linear Unit) Activation Function.
- Leaky ReLU.
- Softmax.

Softmax Activation Function:

The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always remain between 0 and 1. The softmax function is sometimes called the softmax function, or multi-class logistic regression. This is because the softmax is a generalization of logistic regression that can be used for multi-class classification, and its formula is very similar to the sigmoid function which is used for logistic regression. The softmax function can be used in a classifier only when the classes are mutually exclusive. This soft max was implemented in the CNN classifier.

$$softmax(z\ i\ ) = exp\ (z\ i\ )$$

Conclusion:

We compared many available approaches for Multimedia Summarization in this survey. According to the study, many suggested models provide extensive descriptions of the proposed architectures, however there are constraints such as the model being biased, i.e., the approach works better on select types of data. In this work, we presented deep learning approaches such as BERT, Spacy, and GPT-2, as well as assessment metrics such as BLEU and ROUGE scores, to improve performance, and in the future, we will use those methods on real-time data, such as TVSum-50, to generate efficient summaries. For future improvements of this model, one strategy could be to develop a website where an end-user can upload a video and get the summary as output. The other improvement would be to fill in the gaps for missing context from the summary, and automatically determine the best number of sentences to represent the given multimedia.

References:

[1] Derek Miller, “Leveraging BERT for Extractive Text Summarization on Lectures” arXiv:1906.04165, Jun 2019.

[2] Bowen Tan, Virapat Kieuvongngam, Yiming Niu, “Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2” arXiv:2006.01997v1 3 Jun 2020.

[3] Dmitrii Aksenov, Julian Moreno-Schneider, Peter Bourgonje, Robert Schwarzenberg, Leonhard Hennig, Georg Rehm, “Abstractive Text Summarization based on Language Model Conditioning and Locality Modeling”, arXiv:2003.13027v1, 29 Mar 2020

[4] Evlampios Apostolidis, Eleni Adamantidou, Alexandros I. Metsai, “Video Summarization Using Deep Neural Networks: A Survey” arXiv:2101.06072v2 September 2021.

[5] G. Vijay Kumar, Arvind Yadav, B. Vishnupriya, M. Naga Lahari, J. Smriti, D. Samved Reddy, “Text Summarizing Using NLP” Recent Trends in Intensive Computing, doi:10.3233/APC210179, 2021.

[6] Sanjana R, Sai Gagana V, Vedhavathi K R, Kiran K N, “Video Summarization using NLP” International Research Journal of Engineering and Technology (IRJET) Volume: 08 Issue: 08, Aug 2021.

[7] Vishal Pawar, Manisha Mali, “Abstractive Text Summarization of Multimedia News Content using RNN” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-8 Issue-6, August 2019.

[8] Pankaj Gupta, RituTiwari and Nirmal Robert, “Sentiment Analysis and Text Summarization of Online Reviews: A Survey.” International Conference on Communication and Signal Processing, 2016.

[9] Chin-Yew Lin, “Rouge: A Package for Automatic Evaluation of Summaries.” Barcelona Spain, Workshop on Text Summarization Branches Out, Post- Conference Workshop of ACL 2016..

[10] Akshi Kumar, Aditi Sharma, Sidhant Sharma, Shashwat Kashyap, “Performance Analysis of Keyword Extraction Algorithms Assessing Extractive Text Summarization.” International Conference on Computer, Communication, and Electronics (Comptelix), 2017.

[11] N. Moratanch, Dr. S. Chitrakala, “A Surveyon Abstractive Text Summarization.” International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2016.

[12] DharmendraHingu, Deep Shah, Sandeep S.Udmale, “Automatic Text Summarization of Wikipedia Articles.” International Conference on Communication, Information & Computing Technology (ICCICT), 2015.

[13] N.Moratanch, S. Chitrakala, “A Surveyon Extractive Text Summarization.” IEEE International Conference on Computer, Communication and Signal Processing (ICCCSP), 2017.

[14] https://developer.nvidia.com/cuda-toolkit

[15] https://skimai.com/tutorial-how-to-fine-tune-bert-for-summarization/

[16] https://iq.opengenus.org/bert-for-text-summarization/

[17] https://www.mdpi.com/2078-2489/13/2/67/htm

[18]https://www.freecodecamp.org/news/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840/

[19]https://arxiv.org/pdf/2011.04843.pdf

[20]https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/

[21] https://pypi.org/project

[22]https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/

[23]https://data-flair.training/blogs/machine-learning-text-summarization/

[24]https://www.researchgate.net/publication/4220450_An_audio-video_summarization_scheme_based_on_audio_and_video_analysis

[25]https://stackoverflow.com/questions/60381170/which-deep-learning-algorithm-does-spacy-uses-when-we-train-custom-model

[26]https://github.com/designingwithml/mlconcepts/blob/main/extractivesummarization/notebooks/03_Extractive_Summarization_Inference.ipynb

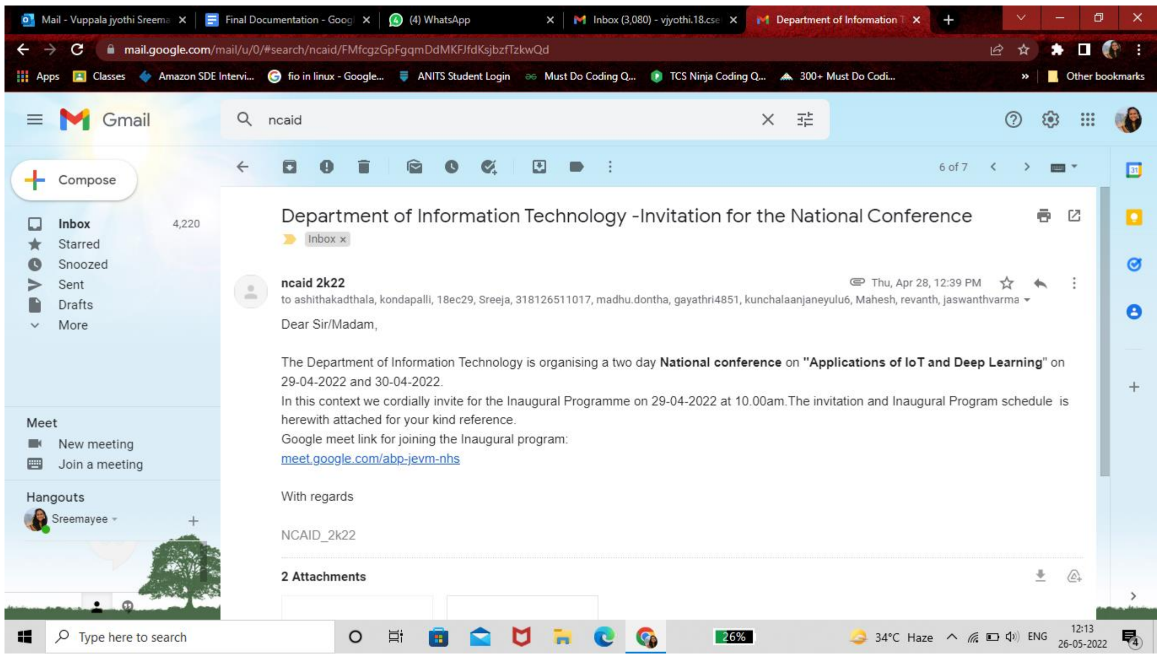
[27] https://huggingface.co/sentence-transformers

[28] https://theaidigest.in/summarize-text-document-using-transformers-and-bert/

8. BASE PAPER

9. PUBLISHED PAPER

10. PUBLISHED CERTIFICATE



Department of Information Technology -Invitation for the National Conference

ncaid 2k22 Thu, Apr 28, 12:39 PM

to ashithakadthala, kondapalli, 18ec29, Sreeja, 318126511017, madhu.dontha, gayathri4851, kunchalaanjaneyulu6, Mahesh, revanth, jaswanthvarma

Dear Sir/Madam,

The Department of Information Technology is organising a two day **National conference** on **"Applications of IoT and Deep Learning"** on 29-04-2022 and 30-04-2022.
In this context we cordially invite for the Inaugural Programme on 29-04-2022 at 10.00am.The invitation and Inaugural Program schedule is herewith attached for your kind reference.
Google meet link for joining the Inaugural program:
meet.google.com/abp-jevm-nhs

With regards

NCAID_2k22

2 Attachments