IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, April 2023
Impact Factor: 7.301

Video-to-Text Summarization using Natural Language Processing

Prerna Mishra[1], Kartik Garg[2], Naveen Rathi[3]
UG Student, Department of Computer Science & Information Technology[1]
UG Student, Department of Information Technology[2]
Assistant Professor, Department of Computer Science & Information Technology[3]
Dronacharya College of Engineering, Gurugram, Haryana, India

Abstract: *Video summarization aims to produce a high-quality text-based summary of videos so that it can convey all the important information or the zest of the videos to users. The process of video summarization involves the conversion of video files to audio files, which are then converted to text files. This entire process is accompanied by the use of transformer architecture of Natural Language Processing. Although a lot of studies have been carried out for text summarization, we present our model, an extractive-video-summarizer, that utilizes state-of-the-art pre-trained ML models and open-source libraries at its core. The extractive-video-summarizer uses the following regime(I) Preparation of a multidisciplinary dataset of videos, (II) Extraction of audios from video files, (III)Text generation from audio files, (IV) Text summarization using extractive summarizers, (V)Entity extraction. We conducted our research primarily on two widely used languages in India - Hindi and English. To conclude, our model performs significantly well and generates tags for videos appropriately.*

Keywords: Natural Language Processing, Video Summarization, Machine Learning, Transformers, Abstractive Summarization, Extractive Summarization

I. INTRODUCTION

Speech Recognition is a popular topic under machine learning concepts. Speech Recognition is getting used more in many fields. For example, the subtitles that we see on Netflix shows or YouTube videos are created mainly through machines using Artificial Intelligence. Other great examples of speech recognizers are personal voice assistants such as Google's Home Mini, Amazon Alexa, and Apple's Siri. Entity extraction, also known as named entity extraction (NER), enables machines to automatically identify or extract entities, like product name, event, and location. It's used by search engines to understand queries, chatbots to interact with humans, and teams to automate tedious tasks like data entry. Entity extraction is a text analysis technique that uses Natural Language Processing (NLP) to automatically pull-out specific data from text and may classify it according to predefined categories. These are named entities, words, or phrases. This includes proper names and numerical expressions of time or quantity, such as phone numbers, monetary values, or dates. The goal of the model is to create an audio file from video and take the input audio and extract the entities. Using NLP techniques, applications must process these videos and generate audio files & text transcripts from the audio file and the related entities Process this text to extract entities and enrich videos with relevant tags. This information will be used to improve the content recommendations.

II. TEXT SUMMARIZATION

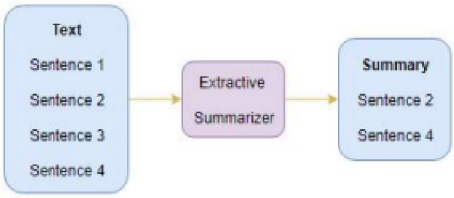
The text thus obtained from the video is put through a summarizer (pre-trained language models from Hugging Face library) and a summary is obtained. Basically, there are 2 types of summarizers: - Extractive(traditional) and Abstractive(advanced). The Facebook BART Transformer model(Abstractive) is an approach to machine learning which combines two existing algorithms, BART and Transformer. Moreover, both algorithms are combined in order to use their complementary abilities. BART is a model composed of a mixture of classic techniques and it is a model focused on regression. Conversely, Transformer is a recurrent model which does not use parameter optimization. It

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, April 2023
Impact Factor: 7.301

follows a transformer-based approach, with encoders and decoders, returning an abstractive summary of the text which is extracted.

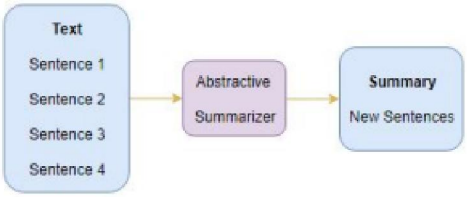
2.2 Extractive Summarization

The extractive text summarizing approach entails extracting essential words from a source material and combining them to create a summary. Without making any modifications to the texts, the extraction is done according to the given measure.

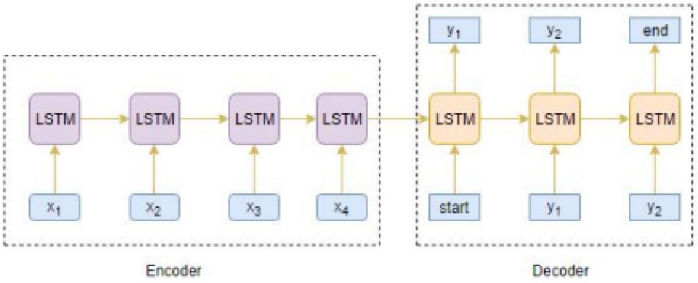


2.2 Abstractive Summarization

This is a very interesting approach. Here, we generate new sentences from the original text. This is in contrast to the extractive approach we saw earlier where we used only the sentences that were present. The sentences generated through abstractive summarization might not be present in the original text.



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.

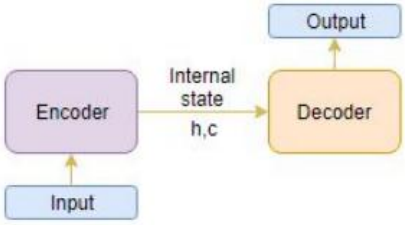


There are two major components of a Seq2Seq model:

- Encoder
- Decoder

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

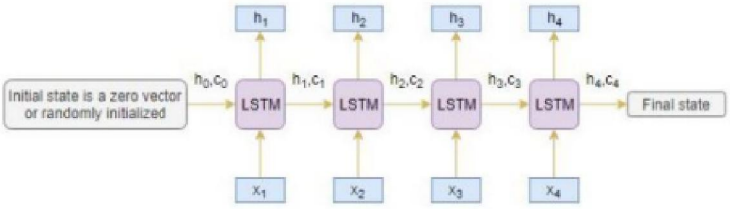
III. ENCODER-DECODER ARCHITECTURE



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, April 2023
Impact Factor: 7.301

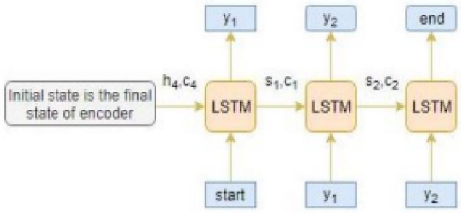
3.1 Encoder

An Encoder Long Short-Term Memory model (LSTM) reads the entire input sequence wherein, at each timestep, one word is fed into the encoder. It then processes the information at every timestep and captures the contextual information present in the input sequence.



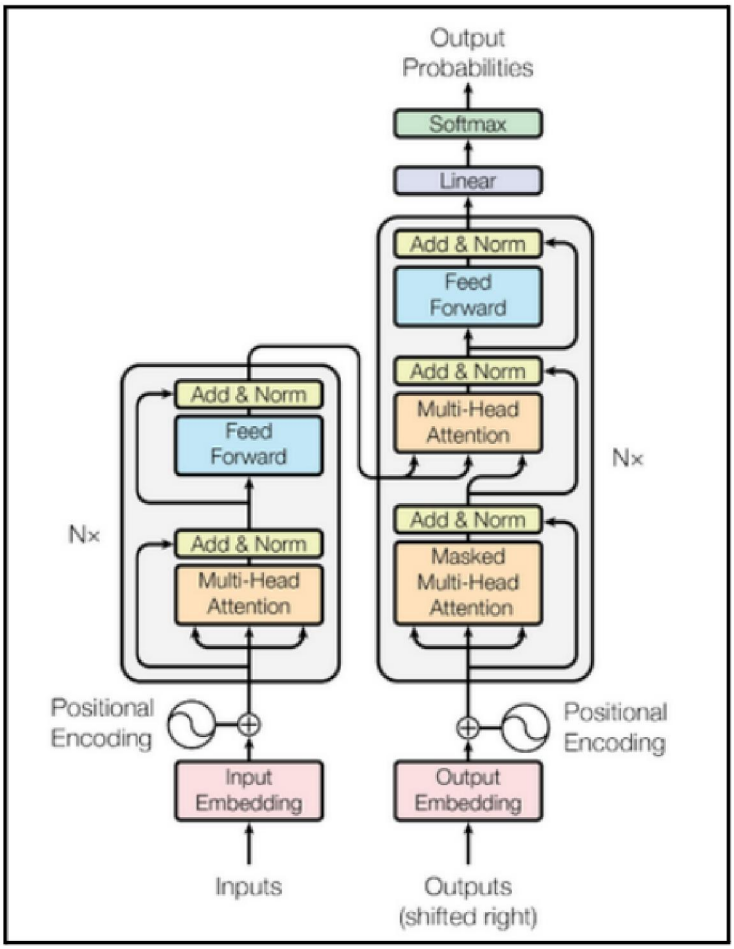
3.2 Decoder

The decoder is also an LSTM network which reads the entire target sequence word-by-word and predicts the same sequence offset by one timestep. The decoder is trained to predict the next word in the sequence given the previous word.



<start>and<end> are the special tokens which are added to the target sequence before feeding it into the decoder. The target sequence is unknown while decoding the test sequence. So, we start predicting the target sequence by passing the first word into the decoder which would be always the token. And the token signals the end of the sentence. More about the model (Source Facebook Research) Our model is composed of a multi-layer convolutional feature encoder f : X 7! Z which takes as input raw audio X and outputs latent speech representations z1,..., zT for T time-steps. They are then fed to a Transformer g : Z 7! C to build representations c1,..., cT capturing information from the entire sequence [9, 5, 4]. The output of the feature encoder is discretized to qt with a quantization module Z 7! Q to represent the targets (Figure 1) in the self-supervised objective (§ 3.2). Compared to vq-wav2vec [5], our model builds context representations over continuous speech representations and self-attention captures dependencies over the entire sequence of latent representations end-to-end. Feature encoder: The encoder consists of several blocks containing a temporal convolution followed by layer normalization [1] and a GELU activation function [21]. The raw waveform input to the encoder is normalized to zero mean and unit variance. The total stride of the encoder determines the number of time-steps T which are input to the Transformer (§ 4.2). Contextualized representations with Transformers. The output of the feature encoder is fed to a context network which follows the Transformer architecture [55, 9, 33]. Instead of fixed positional embeddings which encode absolute positional information, we use a convolutional layer similar to [37, 4, 57] which acts as relative positional embedding. We add the output of the convolution followed by a GELU to the inputs and then apply layer normalization.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, April 2023
Impact Factor: 7.301



IV. NAMED ENTITY RECOGNITION

Named entity recognition (NER)is probably the first step toward information extraction that seeks to locate and classify named entities in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, and monetary values, percentages, etc. NER is used in many fields of Natural Language Processing (NLP), and it can help answer many real-world questions, such as:

Which companies were mentioned in the news article?

- Were specified products mentioned in complaints or reviews?
- Does the tweet contain the name of a person?
- Does the tweet contain this person's location?

Here we have used the Spacy library to extract entities from the summary and to give tags to the text. Named entity recognition (NER) — sometimes referred to as entity chunking, extraction, or identification — is the task of identifying and categorizing key information (entities) in text. An entity can be any word or series of words that consistently refers to the same thing. Every detected entity is classified into a predetermined category. For example, a NER machine learning (ML) model might detect the word "super.AI" in a text and classify it as a "Company". NER is a form of natural language processing (NLP), a subfield of artificial intelligence. NLP is concerned with computers processing and analysing natural language, i.e., any language that has developed naturally, rather than artificially, such as with computer coding languages.

At the heart of any NER model is a two-step process:

- Detect a named entity
- Categorize the entity

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 3, Issue 2, April 2023
Impact Factor: 7.301

Beneath this lie a couple of things. Step one involves detecting a word or string of words that form an entity. Each word represents a token: "The Great Lakes" is a string of three tokens that represents one entity. Inside-outside-beginning tagging is a common way of indicating where entities begin and end.
The second step requires the creation of entity categories. Here are some common entity categories:

Person
E.g., Elvis Presley, Audrey Hepburn, David Beckham

Organization
E.g., Google, Mastercard, University of Oxford

Time
E.g., 2006, 16:34, 2am

Location
E.g., Trafalgar Square, MoMA, Machu Picchu

Work of art
E.g., Hamlet, Guernica, Exile on Main St

NER is suited to any situation in which a high-level overview of a large quantity of text is helpful. With NER, you can, at a glance, understand the subject or theme of a body of text and quickly group texts based on their relevancy or similarity

V. IMPLEMENTATION

5.1 Preparation of Dataset

The dataset contains all types of videos (including news debates, sports commentary, funny videos, TV shows etc.) The videos were downloaded from open resources like YouTube. Because we are dealing with videos, so firstly we need a set of videos on which we can train our model, and then we need videos to test our model. For this, we take 25 videos of 3-5 mins of duration in our train dataset and 20 videos of 5-10 mins of duration in our test dataset. This dataset contains all types of videos from a news anchor reporting from the studio to a politician giving a speech at a political rally. We also have vines/comedy and one-to-one interviews in our dataset.

5.2 Extraction of audio from video

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 GIFs.

5.3 Generating text from audio files

For this task, we have made use of an open-source API from Google - sr.Recognizer(). We basically split the large audio files into chunks, as this API can only read small-sized audio, and apply speech recognition on each of these chunks in order to extract text

5.4 Summarizing the text

The text is passed through a pre-trained model for summarization.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 7.301
Volume 3, Issue 2, April 2023

REFERENCES

[1]. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations
[2]. https://towardsdatascience.com/named-entity-recognition-with-nlt k-and-spacy-8c4a7d88e7da
[3]. J. Li, A. Sun, J. Han and C. Li, "A Survey on Deep Learning for Named Entity Recognition, " in IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 1, pp. 50-70, 1 Jan. 2022, DOI: 10.1109/TKDE.2020.2981314.
[4]. https://medium.com/sciforce/towards-automatic-text-summarizatio n-extractive-methods-e8439cd54715

BIOGRAPHY

Hello readers, we are undergraduate students of computer science and information technology. We have deep interest in natural Language Processing and are keen towards researching in this field.