**Mini Project Report on**



**YOUTUBE TRANSCRIPT SUMMARIZER**

**DJANGO-REACT - CHROME EXTENSION**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name University Roll no**

**Jai Anand** **2016792**

***Under the Mentorship of***

**Ms. Meenakshi Maindola**

**Assistant Professor**

**Graphic Era Deemed to be University.**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

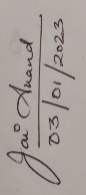
**Dehradun, Uttarakhand**

**July 2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“YOUTUBE TRANSRIPT SUMMARIZER DJANGO-REACT – CHROME EXTENSION”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Ms. Meenakshi Maindola, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.



Jai Anand 2016792

Name University Roll no

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-4** |
| Chapter 2 | Literature Survey | **5-7** |
| Chapter 3 | Methodology | **8-12** |
| Chapter 4 | Result and Discussion | **13-14** |
| Chapter 5 | Future Work | **15** |
|  | References | **15** |

**Chapter 1**

**Introduction**

In the following sections, a brief introduction and the problem statement for the work has been included.

* 1. **Introduction**

Summarization is a vital process in the field of natural language processing that aims to condense a piece of text or speech into a shorter version while preserving its main ideas and key points. With the exponential growth of digital content, summarization techniques have become increasingly important in helping users efficiently retrieve and comprehend information.

Summarization has numerous applications across various domains. In news articles, for example, summarization algorithms can automatically generate concise summaries of lengthy news stories, enabling readers to quickly grasp the main points. In academic research, summarization techniques can assist researchers in efficiently reviewing large volumes of papers and identifying relevant information. Additionally, summarization plays a crucial role in video transcription. With the rise of platforms like YouTube, there is a growing need for tools that can automatically summarize video transcripts, making it easier for users to navigate and comprehend the content.

Example, if we are summarizing a research paper, we may want to focus on the main findings and conclusions of the paper. If we are summarizing a news article, we may want to focus on the most important facts and events. No matter how we choose to summarize a text, it is important to be clear and concise. Our summary should be easy to read and understand, and it should accurately reflect the original text.

**1.2 Need & Importance of Summarization**

1. Save time by quickly reading and understanding long texts.
2. Improve understanding by better understanding the main ideas of a text.
3. Highlight key points by identifying the most important information in a text.

There are many ways to summarize a text, but some common methods include:

1. Identifying the main points of the text.
2. Paraphrasing the text in a shorter and more concise way.
3. Writing a summary that includes the key points of the text and their supporting evidence.
   1. **Types of Summarizations**

There are two main types of summarization techniques: extractive summarization and abstractive summarization.

1. **Extractive Summarization:** Extractive summarization involves selecting and combining important sentences or phrases from the original text to create a summary. This approach relies on statistical methods, such as ranking sentences based on their relevance and importance. Extractive summarization algorithms identify key sentences by considering factors like word frequency, sentence position, and the presence of important keywords. These selected sentences are then combined to form a summary that retains the main ideas of the original text. Extractive summarization is often favored for its simplicity and ability to preserve the original wording.
2. **Abstractive Summarization:** Abstractive summarization, on the other hand, involves generating new sentences that capture the essence of the original content. This technique requires a deeper understanding of the text and often employs natural language generation models. Abstractive summarization algorithms analyze the input text, comprehend its meaning, and generate concise and coherent summaries by paraphrasing and rephrasing the original content. This approach allows for more flexibility and creativity in summarization, as it can generate summaries that may not be present verbatim in the original text. However, abstractive summarization is generally more challenging due to the need for advanced language understanding and generation capabilities.

Both extractive and abstractive summarization techniques have their advantages and limitations. When it comes to ensuring factual correctness and preserving the original phrase, extractive summarization is frequently favored. Contrarily, abstractive summarization produces summaries that are shorter and more coherent but may also include some interpretation or paraphrasing. The decision between these methods depends on the application's particular needs and the desired balance of flexibility and simplicity.

A diagram of an excretory procedure

Description automatically generated

Fig 1.1 Extractive and Abstractive Summarization

**1.4 Project Motivation**

1. Automation: With a REST API, the process of summarizing YouTube transcripts can be automated. Users can submit video transcripts to the API, and the summarization process can be executed programmatically without manual intervention.
2. Scalability: A REST API allows for scalability, enabling multiple users to access the summarization service simultaneously.
3. Integration: The REST API can be easily integrated with other services or platforms. For example, it can be integrated into content management systems, video platforms, or educational platforms, enhancing their functionality by providing automatic transcript summarization capabilities.
4. Customization: By exposing a REST API, developers can customize and extend the functionality of the YouTube transcript summarization service. They can build additional features, such as sentiment analysis, keyword extraction, or topic modeling, on top of the existing summarization capabilities, tailoring the service to specific needs.
5. Collaboration: A REST API allows for collaboration among developers and researchers. They can share and exchange ideas, techniques, and improvements related to YouTube transcript summarization. This fosters a collaborative environment and encourages the development of more advanced and accurate summarization algorithms.
6. Accessibility: By creating a REST API, the YouTube transcript summarization service becomes easily accessible to developers and users. They can integrate the API into their own applications, websites, or services, allowing for seamless access to the summarization functionality.

**Chapter 2**

**Literature Survey**

* 1. **TF-IDF Method for Summarization**

The TF-IDF (Term Frequency-Inverse Document Frequency) method is a popular technique used in text summarization. It calculates the importance of words or terms in a document by considering their frequency within the document and across a collection of documents. In TF-IDF, the term frequency (TF) measures how often a term appears in a document. It is calculated by dividing the number of occurrences of a term in a document by the total number of terms in that document. A higher TF value indicates that a term is more important within the document. The inverse document frequency (IDF) measures the rarity of a term across a collection of documents. It is calculated by taking the logarithm of the total number of documents divided by the number of documents that contain the term. A higher IDF value indicates that a term is more unique or significant across the collection.

The TF-IDF score is obtained by multiplying the TF and IDF values for each term. This score represents the importance of a term within a specific document and the entire collection. Terms with higher TF-IDF scores are considered more important and are more likely to be included in the summary.

In text summarization, the TF-IDF method can be used in extractive summarization approaches. It helps identify the most important sentences or phrases by considering the TF-IDF scores of the terms they contain. Sentences with higher TF-IDF scores are more likely to be selected for inclusion in the summary. However, it's important to note that TF-IDF is a simple method that does not capture the semantic meaning of the text. It relies solely on term frequencies and document frequencies. Therefore, it may not always capture the full context or produce summaries that are coherent and comprehensive. Advanced techniques, such as neural networks and natural language processing, are often used in conjunction with TF-IDF to improve the quality of text summarization.

* 1. **Working of TF-IDF Extractive Summarization**

1. The TF component measures the frequency of a term within a document. It is calculated by dividing the number of occurrences of a term by the total number of terms in the document. A higher TF value indicates that a term is more important within the document.
2. The IDF component measures the rarity of a term across the document collection. It is calculated by taking the logarithm of the total number of documents divided by the number of documents that contain the term. A higher IDF value indicates that a term is more unique or significant across the collection.
3. To obtain the TF-IDF score for a term in a document, the TF value is multiplied by the IDF value. This score represents the importance of the term within the document and the entire collection. Terms with higher TF-IDF scores are considered more important and are more likely to be included in the summary.

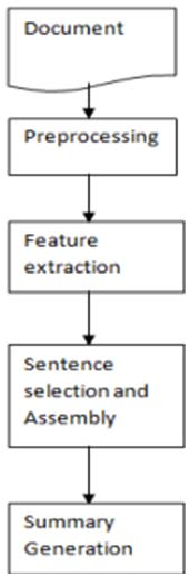


Fig 2.1 Working of the TF-IDF Algorithm for Extractive Text Summarization

* 1. **REST-API overview**

A REST API (Representational State Transfer Application Programming Interface) is a web-based interface that allows communication between different software systems. It follows the principles of REST, using HTTP methods like GET, POST, PUT, and DELETE to perform operations on resources. REST APIs enable developers to access and manipulate data over the internet using standard HTTP protocols. They provide a standardized way for systems to interact, making it easier to integrate and exchange data between different applications. REST APIs are widely used in web development, mobile app development, and cloud computing, providing a flexible and scalable approach to building and integrating software systems.

**A diagram of a company

Description automatically generated**

Fig 2.2 overview of a REST-API built webapp.

**Chapter 3**

**Methodology**

* 1. **Tools Requirements**

Chrome extension for a Django-React YouTube transcript summarizer using Transformer pipelines and TF-IDF requires :

* 1. Django: A Python-based web framework that provides the backend infrastructure for handling data processing and serving API endpoints.
  2. React: A JavaScript library for building user interfaces. It will be used to create the frontend of the Chrome extension, allowing users to interact with the summarization functionality.
  3. Transformer Pipelines and TF-IDF : A Python library that provides a high-level API for using pre-trained Transformer models, such as BERT, for various natural language processing tasks, including text summarization.
  4. Chrome Extension Development Tools: These include the Chrome browser, a text editor or integrated development environment (IDE) for writing code, and the necessary tools for packaging and publishing the extension.
  5. YouTube API: You will need to set up a YouTube API key to access the YouTube transcript data and integrate it with your Django backend.
  6. HTML, CSS, and JavaScript: These web technologies will be used to design and style the user interface of the Chrome extension.
  7. API-Contracts – Used Postman for testing the API built.
  8. **Working Methodology**

1.) Setting up the Django Backend for REST- architecture

Firstly, we can start by creating a test environment by virtualenv, creating a virtual environment, and installing Django in it. Now we are required to some of the necessary packages for the project listed below :

1. Django-rest-framework
2. Cors-headers
3. Tensofrflow/pytorch
4. Youtube transcript api
5. Nltk and punkt package in nltk

2.)Devise the logic for the TF-IDF extractive summarization method :

def calc\_TF(text\_data, freq\_list): # calculating the TF score of each word in each sentence

# {'id': 1, 'tf\_score': 0.08333333333333333, 'key': 'lorem'}, {'id': 1, 'tf\_score': 0.08333333333333333, 'key': 'ipsum'}

tf\_scores = []

for item in freq\_list:

ID = item['id']

for k in item['freq\_dict']:

temp = {

'id': item['id'],

'tf\_score': item['freq\_dict'][k]/text\_data[ID-1]['word count'],

'key': k

}

tf\_scores.append(temp)

return tf\_scores

# calculating the IDF score of each word in each sentence

# {'id': 1, 'idf\_score': 0.0, 'key': 'lorem'}

def calc\_IDF(text\_data, freq\_list) -> list:

idf\_scores = []

cnt = 0

for item in freq\_list:

cnt += 1

for k in item['freq\_dict']:

val = sum([k in tempDict['freq\_dict'] for tempDict in freq\_list])

temp = {

'id': cnt,

'idf\_score': math.log(len(text\_data)/(val+1)),

'key': k

}

idf\_scores.append(temp)

return idf\_scores

# calculating the TF-IDF score of each word in each sentence

# {'id': 1, 'tfidf\_score': 0.0, 'key': 'lorem'}

def calc\_TF\_IDF(tf\_scores, idf\_scores):

tfidf\_scores = []

for j in idf\_scores:

for i in tf\_scores:

if j['key'] == i['key'] and j['id'] == i['id']:

temp = {

'id': j['id'],

'tfidf\_score': j['idf\_score']\*i['tf\_score'],

'key': j['key']

}

tfidf\_scores.append(temp)

return tfidf\_scores

3.)Devise a logic to extract youtube link and then send it to the model for generating the summary from of the transcripts :

from youtube\_transcript\_api import YouTubeTranscriptApi

from transformers import pipeline

def get\_transcript(video\_id):

transcript\_list = YouTubeTranscriptApi.get\_transcript(video\_id)

transcript = ' '.join([d['text'] for d in transcript\_list])

return transcript

def get\_summary(transcript):

summariser = pipeline(

"summarization",

model="sshleifer/distilbart-cnn-12-6",

tokenizer="sshleifer/distilbart-cnn-12-6",

revision="a4f8f3e",

max\_length=100,

)

summary = ''

for i in range(0, (len(transcript)//1000)+1):

summary\_text = summariser(

transcript[i\*1000:(i+1)\*1000])[0]['summary\_text']

summary = summary + summary\_text + ' '

return summary

4.)Build an endpoint for both logic and build a corresponding view for handling the post request made by the client from the frontend.

class SummarizeView(APIView):

def post(self, request):

youtube\_link = request.data.get('youtube\_link')

youtube\_link = youtube\_link.split('=')[1]

transcripts = get\_transcript(youtube\_link)

summary = get\_summary(transcripts)

return Response(summary)

5.)Check if the api is working from POSTMAN. Fire up the localhost url with the given endpoint pass on the json data to it to test the api.

A black background with many small text

Description automatically generated

Fig 3.1 Request to the API and waiting for a response

A black background with blue and white lights

Description automatically generated

Fig 3.2 Sample Response for the Project

6.)Build a frontend (React and make an API call to the url for the response)

function MyForm() {

const [inputValue, setInputValue] = useState('');

var data = '';

const handleSubmit = (event) => {

event.preventDefault();

console.log('sent');

axios

.post('http://127.0.0.1:8000/api/summarize/', {

youtube\_link: inputValue,

})

.then((response) => {

console.log(response.data);

data = response.data;

})

.catch((error) => {

console.error(error);

});

};

7.)Add CSS/SCSS and styling to frontend as per the requirements

**A diagram of a cloud with arrows

Description automatically generated**

Fig 3.3 Working Methodology for the project

**Chapter 4**

**Result and Discussion**

Link : <https://www.youtube.com/watch?v=Np6Z0mOH3gg>

Transcripts : Number of Words : 721

Summary : Tourism ministers handling a sector worth over 2 trillion dollars globally . Tourism is not just about sightseeing it is an immersive experience whether it is Muji or food art or culture . India has something for everyone during our G20 presidency we are organizing nearly 200 meetings in 100 different locations all over India . Focus area is on developing spiritual tourism after all India attracts pilgrims of every major religion of the world . The Eternal City of Varanasi just one of the major spiritual centers now attracts 70 million pilgrims a 10-fold increase from earlier . Hospitality sector has great potential for employment generation . Tourism sector employs more women and Youth compared to many other sectors . Tourism sector is also recognizing the relevance of Tourism sector for Speedy achievements of sustainable development goals excellencies you are working on file interconnected priority areas of green tourism digitalization and skill development tourism is and destination management . A G20 tourism dashboard is being developed in partnership with unwto it will bring together best practices case studies and inspiring stories it will be a first of its kind platform and will be your enduring Legacy . The motto of India G20 presidency is come one Earth one family one future can itself be a motto for Global tourism excellencies . Oters will be celebrating this Festival reaffirming their Abiding Faith in democratic values with more than a million voting booths there will be a no shortage a place is for you to witness this Festival . I invite you all to visit India for this most important of global festivals and with that invitation I wish you all success in your deliberations .

Summary length : 237 words

* 1. **Challenges**

1. Summarization Quality: Generating high-quality summaries that capture the essence of the original content can be challenging. Extractive summarization may result in disjointed sentences, while abstractive summarization may produce summaries that deviate from the original text.
2. Scalability: Handling large volumes of YouTube transcripts and processing them in real-time can be resource-intensive. Ensuring that the summarization process is scalable and efficient is crucial.
3. Language Understanding: YouTube transcripts can contain informal language, slang, or domain-specific terminology. Ensuring that the summarization algorithm understands and accurately captures the meaning of the text can be a challenge.
4. Text Preprocessing: YouTube transcripts often contain noise, such as timestamps, speaker labels, and non-textual elements. Preprocessing the transcripts to remove these noise elements and clean the text can be a complex task.

**Chapter 5**

**Future Work**

1. Fine-tuning Models: Fine-tune the pre-trained Transformer models on domain-specific data to improve the quality of the summaries. This would involve collecting and annotating a dataset of YouTube transcripts specific to a particular domain and fine-tuning the models on this data.
2. User Feedback and Evaluation: Gather user feedback on the generated summaries to evaluate their quality and usefulness. Use this feedback to iteratively improve the summarization system and make it more aligned with user expectations.
3. Real-time Summarization: Develop a real-time summarization feature that can generate summaries as the video is being played. This would require efficient processing and streaming of the transcript data to provide instantaneous summaries.
4. Integration with Recommendation Systems: Integrate the summarization system with recommendation systems to provide personalized video recommendations based on the user's summarized preferences. This would enhance the user experience and help users discover relevant content more efficiently.

**References**

1.) www.analyticsvidhya.com/blog/2022/01/youtube-summariser-mini-nlp-project/#:~:text=The%20basic%20structure%20of%20the,for%20summarizing%20the%20given%20text.

2.) https://pypi.org/project/youtube-transcript-api/

3.)

[https://www.semanticscholar.org/paper/Single-Document-Automatic-Text-Summarization-using-Christian-Agus/b61e1d017eb3c1b2a00e7d0f1230a07397376aa5](https://www.semanticscholar.org/paper/Single-Document-Automatic-Text-Summarization-using-Christian-Agus/b61e1d017eb3c1b2a00e7d0f1230a07397376aa5  4)

[4](https://www.semanticscholar.org/paper/Single-Document-Automatic-Text-Summarization-using-Christian-Agus/b61e1d017eb3c1b2a00e7d0f1230a07397376aa5  4).) <https://www.crio.do/projects/python-youtube-transcript/>

5.)https://www.researchgate.net/publication/228619779\_A\_Survey\_of\_Text\_Summarization\_Extractive\_Techniques