

YouTube Transcript Summarization using Flask and T5 NLP Model

Saurabh Saini*, Sudhanshu**, Shikha Jain***

*Computer Science Department, KIET Group of Institutions

Abstract- YouTube Transcript Summarization is web application that is developed using Flask web framework and T5 model for summarizing the video transcript or subtitles. This application is useful for the student who wants to make notes of YouTube lectures as well as for all the people who uses YouTube for learning purpose

Keyword – YouTube, Transcript, Summarization, NLP, Flask, python, transformer, torch, REST, API..

I. INTRODUCTION

There are number of video lectures and recording being created and uploaded on internet throughout the day. Sometimes it is difficult to cover the whole video because we don't have much time to view the whole video. Although if we can watch the whole video we couldn't find relevant information out of it. So summarization of transcript or subtitles can solve this problem. The task of a text summarizer is to produce a synopsis of any document (or set of documents) submitted to it. The level of sophistication of a synopsis can vary from a simple list of isolated keywords that indicate the major content of the document(s), through a list of independent single sentences that together express the major content, to a coherent, fully planned and generated text that compresses the document(s). The more sophisticated a synopsis, the more effort it generally takes to produce.

Youtube transcript summarization is abstractive summarization which means the model produces a completely different text that is shorter than the original, it generates new sentences in a new form, just like humans do. In this project, we will use transformers for this approach. YouTube Transcript Summarization is web application made using flask and T5 model that summarize the YouTube video transcript or subtitle. This application uses HTML as fronten which is used to render the web page where user can type the youtube video URL and also to see the summary the of the transcript. For backend we have used the flask. .This application uses three dependencies flask , youtube_transcript_api and transformers[torch].Flask is micro web framework written in python which is used for making RESTful API apps. YTS is also made using flask web framework. YTS uses Youtube Transcript API. It allows you to get the transcript or subtitles for a given YouTube video. It also works for automatically generated subtitles, supports translating subtitles.Youtube transcript api also used for formatting the the transcript from JSON to text.

II. LITERATURE SURVEY

literature review revealed research and studies based on the implementation of a summarization of text using Natural Language Processing and creating web application using Flask and also the different python dependencies required for making this web application i.e. YoutubeTranscriptApi ,Transformer , Flask, T5 pretrained model:

- [1] Automated Text Summarization in SUMMARIST by Eduard Hovy and Chin-Yew Lin: They describe the about the Text Summarization equation i.e. summarization = topic identification + interpretation + generation and also about Text Pre-processing.
- [2] Text Summarization ISSN: 2278-0181: This discuss about Text Summarization using Restricted Boltzmann Machine(RBM),Deep Neural Networks(DNN).
- [3] Text Summarization e-ISSN: 2395-0056 : They describe about summarization using Recurrent Neural Network(RNN).
- [4] Python Flask Framework by Vasanth Nagarajan - A Step by Step Guide For the Beginners for learning flask web framework.
- [5] T5 by Hugging Face Community: Pretrained T5 model documentation.
- [6] Youtube-transcript-api: pypi.org.

The approach builds up after literature review

- Get transcripts/subtitles for a given YouTube video Id using a Python API.
- Perform text summarization on obtained transcripts using HuggingFace transformers.
- Build a Flask backend REST API to expose the summarization service to the client.
- Develop a chrome extension which will utilize the backend API to display summarized text to the user

III. SYSTEM OBJECTIVE

This application is used to summarize the transcript of youtube video by NLP techniques. The objective of this application is to help students to revise the lecture without actually seeing the whole video. Summarization of text is essential to get the important information while dealing with large collection of documents. With the advent of World Wide Web information has become intrinsic part of our life. To remember the details of every information is not possible for human mind. Therefore summarization of text documents plays a very important role in information gathering. Also the objective to make application.

- Responsive
- User Friendly

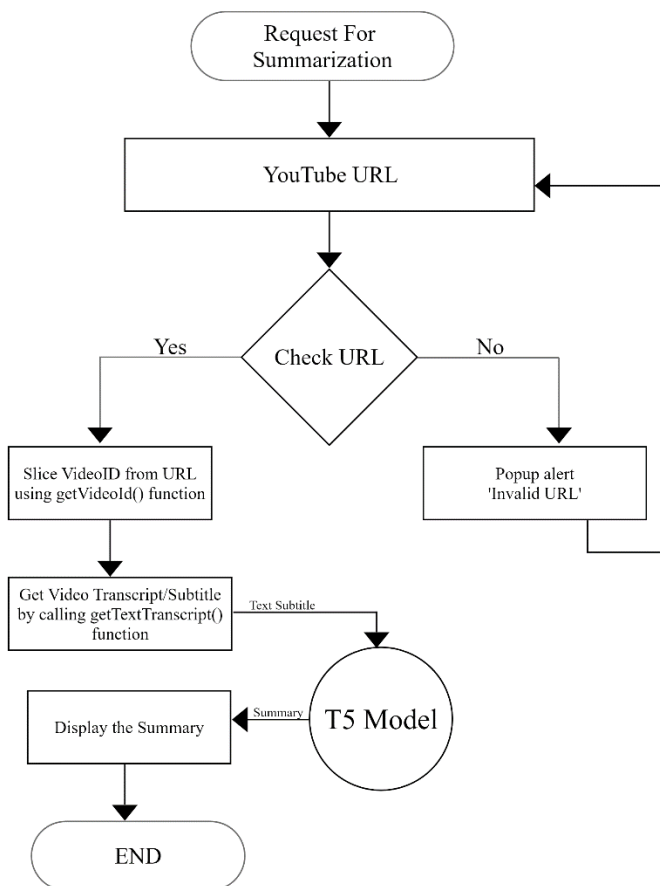
- Robust

This application can also be used for

- Meetings and video-conferencing - A system that could turn voice to text and generate summaries from your team meetings.
- Patent research - A summarizer to extract the most salient claims across patents.

IV. APPLICATION ARCHITECTURE

Summarization Flow



YouTube Transcript Summarization application involves following step to perform summarization.

1. Render the home page

The application first render the home page which have input field where user can type the youtube video url

```
@app.route('/home')
```

```
def home_page():
```

```
    return render_template('home.html')
```

2. Validating the URL

In this step we will validate the URL whether the URL is valid or not. For this we will use Javascript which do this operation on client side.

```
function isValidURL(string) {
    var res = string.match(/(http(s)?:\/\/.)?(www\.)?[-a-zA-Z0-9@:_%._\+~#={2,256}\.]{2,6}\b([-a-zA-Z0-9@:_%._\+~#?&//=]*)/g);
    return (res !== null)
};
```

3. Requesting Argument

The application request for youtube video which is passed as argument in the URL.

```
args = request.args
```

```
youtube_url = args.get("youtube_url")
```

4. Slice the videoID

In this step the application slice or trim the url so that we can get videoID by calling the getVideoID() function. VideoID is a unique key used for identifying a particular video.

```
def getVideoId(youtube_url):
    id = youtube_url.split("v=")
    videoID = id[1][1:]
```

5. Fetch the transcript

In this phase we will fetch the transcript by calling Youtube Transcript

6.. Transforming JSON transcript to Text

Fetches transcript is in JSON format but we have convert this into text so that we can summarization on that.

```
transcript = getYoutubeTranscript(youtubeID)
```

```
formatter = TextFormatter()
```

```
text_transcript = formatter.format_transcript(transcript)
```

7. Summarize the transcript

Parse the text transcript to the T5 model for summarizing the transcript.

```
def summarizeTranscriptT5(youtubeID):
    model = T5ForConditionalGeneration.from_pretrained("t5-base")
    tokenizer = T5Tokenizer.from_pretrained("t5-base")
    text_transcript = getTextTranscript(youtubeID)
    inputs = tokenizer.encode("summarize: " + text_transcript,
    return_tensors="pt", max_length=1024, truncation=True)
    outputs = model.generate(
        inputs,
        max_length=150,
        min_length=40,
        length_penalty=2.0,
        num_beams=4,
        early_stopping=True)
    return tokenizer.decode(outputs[0])
```

You can control the summary length by changing the value of max_length and min_length.

8. Render the summary

Finally we render the summary to the web page where user can see the summary.

V. FEATURES

1. It saves time as it allows multiple students to give their exams at the same time and displays the result immediately as the test gets over.
2. It will reduce paper wastage.
3. System will check responses given by the students automatically and immediately.
4. Question papers are automatically generated by the server.
5. Administrator has the administrative power to create, modify and delete the exam.
6. Users can log in with a specific id and give the test and can see the result as well.
7. It will reduce the work of accessing the answers given by the candidates.
8. The android app will help share the images of the questions and the practice papers created by the administrator.
9. Questions can include diagrams, graphical representations, etc.
10. The online examination system has a password-based authentication.
11. Students and faculty details are stored in the database.

VI. CONCLUSION

Online examination tests are unquestionably turning into a medium of assessing candidates' knowledge and aptitude. They provide flexibility to educational institutes to create, manage and evaluate examinees effortlessly. With quick assessment and real-time report generation, results can be declared immediately after the students have completed the test. Online examination tests have revolutionized the education industry by automating manual, tedious, and cumbersome processes. Online Examination System is significantly superior among the other exams. We have come to result that the problems can be solved by introducing new security systems using biometrics, we can identify the student's identity by analyzing digital signature or by fingerprint mechanism and also by providing web cameras in the examination hall. Although web cameras Sometimes gets failed, if supposed a candidate is giving exam and facing downwards in such case Iris recognition and face recognition must be used. We conclude that no mechanism is ideal. Each mechanism has some restriction on its own. Key concepts are to develop paperless environment and to convert all the documentation in digital form.

VII. FUTURE WORK

As exam portal has to be secure and should have some anti-cheating mechanism so there should be proctoring. So in upcoming version of exam engine will have following functionalities like proctoring, live candidate viewing options.

There will be more role in applications. The variety of question type will be increased in upcoming version.

REFERENCES

- [1] F. Chen, K. Han, and G. Chen, "An approach to sentence-selection based text summarization," in TENCON'02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, vol. 1. IEEE, 2002, pp. 489-493.
- [2] Y. Sankarasubramaniam, K. Ramanathan, and S. Ghosh, "Text summarization Using Wikipedia," Information Processing & Management, vol. 50, no. 3, pp. 443-461, 2014.
- [3] The automatic creation of literature abstract H.P.Luhn
- [4] 1 Text Summarization: An Essential Study [Prabhudas Janjanam and CH Pradeep Reddy
- [5] Extractive Text Summarization Using Sentence Ranking[J.N.Madhuri and Ganesh Kumar.R]
- [6] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023,2016.
- [7] K. M. Svore, L. Vanderwende, and C. J. Burges, "Enhancing singledocument summarization by combining ranknet and third-party sources." in EMNLP CoNLL, 2007, pp. 448-457.
- [8] CS224N Project Report Faster Transformers for Text Summarization(Amaury Sabran asabran@stanford.edu Alexandre Matton alexmt@stanford.edu
- [9] Automated Text Summarization in SUMMARIST by Eduard Hovy and Chin-Yew Lin: They describe the about the Text Summarization equation i.e. summarization = topic identification + interpretation + generation and also about Text Pre-processing.
- [10] Text Summarization ISSN: 2278-0181: This discuss about Text Summarization using Restricted Boltzmann Machine(RBM),Deep Neural Networks(DNN).
- [11] Text Summarization e-ISSN: 2395-0056 : They describe about summarization using Recurrent Neural Network(RNN).
- [12] Python Flask Framework by Vasanth Nagarajan - A Step by Step Guide For the Beginners for learning flask web framework.
- [13] T5 by Hugging Face Community: Pretrained T5 model documentation.
- [14] Youtube-transcript-api: pypi.org.
- [15] G.Padmapiya,Dr.K.Duraiswamy,"An Approach for text summarization using Deep Learning Algorithm ",JCSP,2014.
- [16] Heena A. Chopade , Dr.Meena Narvekar, "Hybrid Auto Summarization ,"Using Deep Neural Network And Fuzzy Logic System", IEEE, 2017.
- [17] Ashwini Anbekar, Kajol Shah , Minakshi Agarwal, Simica Pawar , Asma Shaikh,"Text Summariazation Using Restricted Boltz-mann Machine:Unsupervised Deep Learning Approach",IEEE,2018.

- [18] Abdullah Goktug Mert, "Text Summarizer with Deep Learning", METU, 2016.
- [19] Trun Kumar, "Automatic text summarization", NITR, 2014.
- [20] S.Santhana Megala, Dr. A.Kavitha, Dr. A.Marimuthu, "Enriching Text Summarization using Fuzzy Logic", IJCSIT, 2014.
- [21] G.Padmapriya, Dr.K.Duraiswamy, "Association Of Deep Learning Algorithm With Fuzzy Logic For Multi Document Text Summariazation", JATIT, 2014.
- [22] Mehdi Jafari, Amir Shahab Shahabi, Jing Wang, Yongruri Qin, Xiaohui Tao, Mehdi Gheisari, " Automatic Text Summarization using Fuzzy Inference", IEEE, 2016.

AUTHORS

First Author – Saurabh Saini, Research Scholar, KIET Group of Institution, saurabh.1822cs1124@kiet.edu

Second Author – Sudhanshu, Research Scholar, KIET Group of Institution, sudhanshu.1822cs1146@kiet.edu

Third Author - Shikha Jain, Assistant Professor, KIET Group Of Institutions, shikha.jain.cse@kiet.edu