



Dissertation on
“Smart Video Retrieval and Question Answering System”

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE19CS390B – Capstone Project Phase - 2

Submitted by:

Kusuma Shree V	PES1UG19CS241
Likhith	PES1UG19CS242
Mahim Dashora	PES1UG19CS251
Mahima Dubey	PES1UG19CS252

Under the guidance of

Dr. Jayashree R
Professor, CSE Dept.
PES University

August - December 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, Bengaluru – 560 085, Karnataka, India



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100 Feet Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

‘Smart Video Retrieval and Question Answering System’

is a bonafide work carried out by

**Kusuma Shree V
Likhith
Mahim Dashora
Mahima Dubey**

**PES1UG19CS241
PES1UG19CS242
PES1UG19CS251
PES1UG19CS252**

in partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE19CS390B) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period August - December 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature
Dr. Jayashree R
Professor, CSE Dept.

Signature
Dr. Shylaja S.S.
Chairperson

Signature
Dr. B.K. Keshavan
Dean of Faculty

External Viva

Name of the Examiners

Signature with Date

1. _____

2. _____

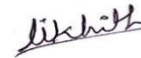
DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled “**Smart Video Retrieval and Question Answering System**” has been carried out by us under the guidance of Dr Jayashree, Professor in CSE Dept and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology in Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester August - December 2022. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1UG19CS241 Kusuma Shree V



PES1UG19CS242 Likhith



PES1UG19CS251 Mahim Dashora



PES1UG19CS252 Mahima Dubey



ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Jayashree, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE19CS390B - Capstone Project Phase – 2.

We are grateful to the project coordinator, Prof. Mahesh H.B., for organizing, managing, and helping with the entire process.

We take this opportunity to thank Dr. Shylaja S.S. Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support we have received from the department. We would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

We are deeply grateful to Dr. M.R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J., Vice-Chancellor, PES University for providing us various opportunities and enlightenment every step of the way.

Finally, this project could not have been completed without the continual support and encouragement we have received from our family and friends.

ABSTRACT

With more and more video content being produced on a daily basis, utilizing it to the fullest is becoming more challenging. The recent research in the video retrieval area has focused on automated, high level feature indexing on shots or frames. One important application of such indexing is to support precise video retrieval. In this work, we aim to propose a new video retrieval and question-answering system on lecture videos, using of their verbose and narrative nature. Given a question and a video corpus, we attempt to search relevant video, provide not just short text answers but also long answers and provide timestamps of the video segment which contains the answer to that question, which saves a great deal of time for the users from watching the entire video to find the answers themselves.

Our solution for this problem as a smart video retrieval system, leverages the progress made in extractive question-answering models like RoBERTa and DistilBERT. The system is designed mainly for educational videos in English, with major use cases in schools and colleges. The problem is divided mainly into two parts: 1. Video retrieval and 2. Timestamps retrieval within the video. We implemented different variations of the proposed architecture involving slide-text extraction, TFIDF and semantics-based. We also introduced two new datasets based on educational YouTube videos and Cloud-computing course videos, which are used to generate results and evaluate performance across all the variations implemented. We were able to achieve the best of 89.35% and 84.83% accuracy in video retrieval, and just 1% and 1.7% error in timestamp retrieval considering outliers across the two datasets respectively.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
1.	INTRODUCTION	1
2.	PROBLEM STATEMENT	3
	2.1 Video Retrieval System	3
	2.2 Question-Answering Model	3
	2.3 Final User Interface	3
3.	LITERATURE REVIEW	4
	3.1 Introduction	4
	3.2 An Optimized E-Lecture Video Retrieval based on Machine Learning Classification	4
	3.2.1 Video Processing	4
	3.2.2 Search System	5
	3.3 Content Based Lecture Video Retrieval Using Speech & Video Text Information	5
	3.3.1 Slide Video Segmentation	5
	3.3.1.1 Models Used	6
	3.3.1.2 Video OCR for Lecture Videos	6
	3.3.2 Keyword Extraction and Video Search	6
	3.4 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	6
	3.4.1 Input Text Processing	7
	3.4.2 Datasets Used and Tested	7
	3.4.2.1 Pre-training Data	7
	3.4.2.2 Evaluation Data	7
	3.4.2.3 Drawbacks	7
	3.5 Tutorial VQA: Question Answering Dataset for Tutorial Videos	7
	3.5.1 Dataset	7

3.5.1.1 Data Collection	8
3.5.2 Methodology Baselines	8
3.5.2.1 First Baseline: Sentence-level Prediction	8
3.5.2.2 Second Baseline: Segment Retrieval	8
3.5.2.3 Third Baseline: Pipeline Segment Retrieval	8
3.5.3 Results and Conclusions	9
3.6 Automatic Lecture Video Indexing Using Video OCR Technology	9
3.6.1 Introduction	9
3.6.2 Approach	9
3.6.3 Conclusions	9
3.7 Summarization of Video Lectures	10
3.7.1 Introduction	10
3.7.2 Approach	10
3.7.2.1 Transcriptor	10
3.7.2.2 Summarizer	10
3.7.2.3 Segmenter, Ranking Algorithm and Condenser	10
3.7.2.4 Chalkboard Extractor and PDF Generator	11
3.7.3 Conclusions	11
3.8 Extensive Literature Survey	11
3.8.1 Transcription	11
3.8.2 Retrieving Text from Videos	12
3.9 Conclusions from the Literature Survey	12
4. DATASET	13
4.1 YouTube Dataset	13
4.2 Cloud Computing Course Dataset	14
5. PROJECT REQUIREMENTS SPECIFICATION	15
5.1 General Constraints, Assumptions and Dependencies	15
5.2 Risks	15
5.3 External Interface Requirements	15

5.4 Hardware Requirements	16
5.5 Software Requirements	16
6. SYSTEM DESIGN	17
6.1 Proposed Design Methodology	17
6.2 High Level Design	18
6.2.1 Video Uploading Handling	19
6.2.2 Query Handling	19
6.3 Use-Case Diagram	20
7. PROPOSED METHODOLOGY	21
7.1 Video Retrieval System	21
7.1.1 Constructing an Index of the Videos Based on the Video Content	21
7.1.2 Automated Segmentation of the Videos	22
7.2 Question Answering and Timestamp Retrieving Model	23
8. IMPLEMENTATION AND PSEUDOCODE	24
8.1 Video Transcript Generation	24
8.2 Keyword Index	24
8.3 Semantic Index	26
8.4 Title Based Segmentation	26
8.5 Similarity Based Segmentation	27
8.6 Question Answering and Timestamp Retrieving Model	27
8.7 Implementation of Different Variations of Proposed Architecture	28
8.7.1 Base	28
8.7.2 V1	28
8.7.3 V2	28
8.7.4 V3	28
8.8 Metrics	29
8.8.1 Confusion Matrix	29

8.8.2 Effective Error	29
9. RESULTS AND DISCUSSION	31
9.1 YouTube Dataset Results	31
9.1.1 Confusion Matrices, Effective-Error Graphs and Accuracy Table	31
9.1.2 Observations	34
9.2 Cloud Computing Dataset Results	34
9.2.1 Confusion Matrices, Effective-Error Graphs and Accuracy Table	34
9.2.2 Observations	37
9.3 Overall Timestamp Retrieval Results	38
10. CONCLUSION AND FUTURE WORK	40
REFERENCES/BIBLIOGRAPHY	41
APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS	

LIST OF FIGURES

Figure No.	Title	Page No.
4.1	Snapshot of YouTube Videos Dataset	13
4.2	Snapshot of Cloud Computing Videos Dataset	14
6.1a	Final methodology of video processing	17
6.1b	Final methodology of query processing	18
6.2.1	High Level Design of Video Handling	19
6.2.2	High Level Design of Query Handling	19
6.3	Use-Case Diagram	20
8.2a	Database schema used	25
8.2b	Overview of database storage	25
8.3	Overview of pinecone database	26
9.1.1a	Base- No segmentation, only transcript-based index on Yt Dataset	31
9.1.1b	Similarity based segmentation, only transcript-based index on Yt Dataset	31
9.1.1c	Similarity based segmentation, transcript and extracted-text-based index on Yt Dataset	32
9.1.1d	1-minute length segmentation, transcript and extracted-text based index on Yt Dataset	32
9.1.1e	DistilBERT Model: Effective-Error on Yt Dataset	32
9.1.1f	RoBERTa Model: Effective-Error on Yt Dataset	33
9.2.1a	Base- No segmentation, only transcript-based index on CC Dataset	34
9.2.1b	Similarity based segmentation, only transcript-based index on CC Dataset	35
9.2.1c	Similarity based segmentation, transcript and extracted-text-based index on CC Dataset	35
9.2.1d	1-minute length segmentation, transcript and extracted-text based index on CC Dataset	35
9.2.1e	DistilBERT Model: Effective-Error on CC Dataset	36
9.2.1f	RoBERTa Model: Effective-Error on CC Dataset	36

LIST OF TABLES

Table No.	Title	Page No.
9.1.1	Accuracy of different models with Yt Dataset	33
9.2.1	Accuracy of different models with CC Dataset	37
9.3a	DistilBERT Model: Effective Error Table	38
9.3b	RoBERTa Model: Effective Error Table	39

CHAPTER 1

INTRODUCTION

Videos are the fastest-growing medium to create and deliver information effectively in recent times. Consequently, the videos have been extensively used as main data sources in question-answering problems. The previous studies have only concentrated on clustering videos into categories and some generating short-text answers to simple questions based on the multimodal contents of the video. These proved helpful but not sufficient to meet the needs of the user. For example, students might have forgotten to note down some important points related to a sub-topic taught within an educational video. In this case, they will have to find the right video within the provided category and re-watch the entire video to find the required specific part as short-answer-generating models do not prove useful here.

The most common problem while watching lecture videos is that of watching the entire video, even though we are interested in just a particular sub-part of the video. Though several ways like manually segmenting and captioning the videos are practiced addressing this problem, there is still a gap left to obtain better automated results. Retrieving accurate timestamps of the video based on the users' queries or interests can help the users get their desired information faster and easier without wasting much time.

In this project, we have implemented and discussed the video retrieval system based on a variety of indexes built using different contents of the video like transcript and text extracted from the frames. This video retrieval system helps search videos containing topics or concepts related to the users' query from a large corpus of videos. The videos retrieved are further passed into a question-answering module for extracting answers to the user's questions. We also compared the performances of different Question-answering models, particularly RoBERTa [1] and DistilBERT [2], specifically for answer extraction

and timestamp retrieval. The proposed architecture takes care of finding the right video, extracting the right answer, and providing the right snippet or timestamps of the video. We have also created two different datasets comprised of educational videos from YouTube and from the Cloud-computing course at our university to test and evaluate the results of the proposed solution that was implemented. We believe this would be greatly beneficial to the students and can find major applications in educational institutions.

CHAPTER 2

PROBLEM STATEMENT

The problem statement is briefed in the three subsections below:

2.1 Video Retrieval System:

The video retrieval system will be able to retrieve the video that matches the user's question from the uploaded video corpus. The system maintains a global inverted index for all the videos using their transcripts in the corpus, using which relevant videos are fetched and ranked according to texts from the video frames and other criteria.

The system also provides the timestamps in the video where the answer for the user's question is present, using the previously generated timestamp index.

2.2 Question-Answering Model:

The question answering model generates answers to the questions posted by the user based on the contents of the video. Our approach not only works for short-answered questions, but also will be capable of extracting long (descriptive) answers to the questions if required by the user. From the video retrieved, answers will be extracted for a particular question of the user and timestamps corresponding to answers will be calculated.

2.3 Final User Interface:

The final user interface will be provided where the user will be able to search for the video from the uploaded video corpus based on their interested topic. The user can post a query/question related to the video contents and our system will extract a long answer to that question and will display it to the user. The user can also request for the timestamp of the segment in which their interested sub-topic appears. This request will be processed by our system based on the transcript/frame-text representation of the video contents generated by our model and will be provided to the user.

CHAPTER 3

LITERATURE SURVEY

3.1 Introduction

This section gives the details about the literature survey we conducted on our problem statement. We have provided the contents of four papers below.

3.2 An Optimized E-Lecture Video Retrieval based on Machine Learning Classification:[1]

Machine Learning based text classification algorithm is presented here, for efficient search and retrieval of lecture videos. Here, necessary information is extracted from lecture-videos and is used to classify videos into different categories of similar videos. This necessary information is extracted in this manner: Video -> Audio -> transcript -> Summary -> keyword extraction

OCR (Optical Character Recognition) and ASR (Automatic Speech Recognition) methods are used to generate the textual information, which represent the content of the video lectures.

3.2.1 Video Processing:

The audio track is extracted in WAV format. Audio is divided into segments of around 20 sec, then the Google Speech Recognition (GSR) library of Python is used for the text transcript generation. Summary and important keywords from the transcript documents are generated using Python's gensim library. It diminishes the complexity and length of the document, and therefore reduces the time required for training the text classification model while efficiency of the model is retained.

(But we may not consider the summary, as we aren't classifying the videos, and we prefer continuous portions of videos to be fetched)

3.2.2 Search System:

Firstly, the search category and the keyword are fetched from the user query. If the category is not mentioned by the user while giving the query, it is predicted by the trained ML text classification model. Using the text transcripts of the category, the relevant documents are chosen based on the TF-IDF and Cosine Similarity scores.

The system's performance is compared by training the system using different models like Naive Bayes, Support Vector Machine and Logistic Regression. It is found that Naive Bayes classification algorithm achieved better performance both in terms of time and also search relevance results.

3.3 Content Based Lecture Video Retrieval Using Speech and Video Text Information:[2]

Here the paper is about automated video indexing and search on huge lecture video corpus. Video is converted into textual data using OCR (Optical Character Recognition) on different frames of video and ASR (Automatic Speech Recognition) on audio tracks.

This extracted textual data is used as keywords.

3.3.1 Slide Video Segmentation:

Video segment is nothing but continuous partition of video on the same topic or subtopic. Based on temporal scope of lecture slides, different segments are determined. Connected Components (CCs) are used instead of pixel-level-differencing for the differencing analysis, while creating segments. The text lines, figures, tables, etc. are part of Connected-Components.

3.3.1.1 Models Used:

Image Intensity Histogram and Support Vector Machine (SVM) classifier. Radial Basis Function (RBF) is used as the kernel. The Histogram of Oriented Gradients (HOG) is also used in order to make the comparison. The achieved segmentation precision and recall were 95 and 98 percent for the test data respectively.

3.3.1.2 Video OCR for Lecture Videos:

An edge-based multi-scale text detector is in the detection stage to quickly localize candidate text regions with a low rejection rate. Text and non-text blocks are found using Image entropy-based adaptive refinement algorithm. Non-text blocks are removed using SVM and SWT (Stroke Width Transform). Tesseract OCR engine is used. Slide structure is also analyzed based on positions of texts.

3.3.2 Keyword Extraction and Video Search:

Segment level keywords are extracted by taking each individual lecture video as a document corpus with each video segment as a single document, whereas video-level keywords are extracted by processing all lecture videos in the database, with each video as a single document. OCR and ASR transcripts are used for the above processes. TF-IDF approach is used to retrieve and rank videos.

3.4 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding:[4]

In this paper, BERT is designed to pretrain deep bidirectional representations from **unlabeled text** by enabling representations from **both left and right context** in all layers. The Transformer is a remarkable improvement because **it can process data in any** order, whereas RNNs and CNNs do require data to be processed in order. Hence pre-trained models like **BERT** was trained on huge amounts of language data .

3.4.1 Input Text Preprocessing:

To show the position of words in a sentence BERT uses **positional embeddings** because a Transformer unlike an RNN, cannot capture “sequence” of information. BERT takes sentence pairs as inputs for tasks (Question-Answering) so that it can distinguish between first and second sentences(**segment embeddings**). **Token embeddings** are the specific tokens taken from the WordPiece vocabulary.

3.4.2 Datasets Used and Tested:

3.4.2.1 Pre-training data: BERT model was pretrained on English Wikipedia (2,500M words) and Books Corpus (800M words) .

3.4.2.2 Evaluation data :

BERT was evaluated on SQuAD v2.0 (F1 score 83.1) and SQuAD v1.1 (F1 Score 93)

3.4.2.3 Drawbacks:

BERT was significantly undertrained, and can match or exceed the performance of every model published after it. The XLNet architecture (Yang et al., 2019) is pre trained using nearly 10 times more data than the original BERT .

3.5 Tutorial VQA: Question Answering Dataset for Tutorial Videos:[5]

This paper is about a new question answering model for the non-factoid, that is open-ended or passage-expecting questions on tutorial videos.

3.5.1 Dataset:

Their dataset is about instructional videos about an image editing software. It had 76 videos in corpus. Verbal instructions in the video are transcribed and segmented. These segments are used to answer the question the user asks.

3.5.1.1 Data collection:

Each video was preprocessed to extract the transcripts and the time-stamp details for each sentence. Amazon Mechanical Turk was used by them in order to form the question-answer pairs.

3.5.2 Methodology Baselines:

3.5.2.1 First Baseline: Sentence-level prediction:

This baseline predicts the starting and the ending sentence index. It is based on RaSor, which joins the embedding vectors pertaining to the beginning and the ending words in order to represent a span. Two bi-LSTMs were used to encode the transcript as sentence encoding and passage level encoding (latent meaning). One-layer feed forward network is used to evaluate a score between each span and the question asked. Cross-entropy is used as an objective function. Consequently, the span with the highest score is chosen as an answer.

3.5.2.2 Second baseline: Segment retrieval:

The segmentation information is added to this baseline in addition to the transcript information. This baseline uses attentive LSTM. Firstly, the model encodes the two inputs given(segment and question) and then the encoded sentence is re-weighted using attention weights. The final score is obtained using a one-layer feedforward network. Again, the cross entropy is used as an objective function. Accuracy and MRR (Mean Reciprocal Ranking) were the chosen metrics to evaluate this model.

3.5.2.3 Third baseline: Pipeline Segment retrieval

In this approach, the cosine similarities between the segment embeddings and question embeddings are computed and compared. Firstly, the correct video from 76 videos that contains the answer is retrieved. Using the TF-IDF embeddings generated for the video transcripts and questions, the top ten videos are filtered and stored. The cosine similarity between the question and segments belonging to these filtered ten videos is computed. But segment representations are not learned here.

3.5.3 Results and Conclusions:

Third baseline produced better results out of all with an accuracy of 68%. Most of the errors from the answers fetched were because the predicted answer span for a question was either a subset or superset of the ground(true) answer span. Therefore, they have suggested exploring a pointer network which determines boundary sentences in order to predict the correct answer span. Also, for future works, they have proposed to work on multimodal data for better video representations rather than the transcription.

3.6 Automatic Lecture Video Indexing Using Video OCR Technology:[6]

3.6.1 Introduction:

This paper focuses on lecture video portals and multimedia indexes. Text displayed in the videos are considered a valuable source of data/keywords for indexing as displayed texts in case of lecture video are mostly related to the content. Lecture contents are retrieved based on this index. OCR technology is used for the above mentioned display-text extraction. A weighted DCT (discrete cosines transformation) based text detection and segmentation is used to analyze slide video structure.

3.6.2 Approach:

A video is decomposed into segmented key frames to index, these selected video frames get processed in later steps. To process the title and content of frame Slide content distribution algorithm is used. After detecting title and content components, text detection and text recognition determine adapted text.

3.6.3 Conclusions:

In this paper a novel slide video segmenter was developed along with text detection. The results of this paper's video text detection, segmentation and recognition algorithms are suitable for content-based lecture video indexing and retrieval.

3.7 Summarization of Video Lectures:[7]

3.7.1 Introduction:

This paper mainly focuses on the two main artifacts of any video lecture, the lecturer's explanation and the lecturer's handwritten information (chalkboard representation). With the information extracted from these, they aim to consolidate all the relevant important points into a PDF which acts as a useful reference material to the scholars and working professionals. Without transforming the general schema, the consolidated PDF shall have all the characteristics of the lecture.

3.7.2 Approach:

3.7.2.1 Transcripitor:

The process of creating a transcript is performed in three steps. The video file is converted to an audio file. The audio file is then uploaded into the Google cloud bucket. The audio file is converted into a transcript with the help of Google talk to transcript application programming interface. uses a neural network model to perform the conversion of audio to text format.

3.7.2.2 Summarizer:

The general approach is followed for extracting the summary using the Natural speech-based method of working. To calculate the weighted frequency of each term, its frequency is divided with the frequency of the most repeating term. The matching terms in the original segments are replaced with their corresponding weighted-frequencies. The segments are sorted in reverse order of their aggregate and the segments with highest frequencies are used to brief the text.

3.7.2.3 Segmenter, Ranking Algorithm and Condenser:

The segmenter mainly works on the audio to segment the video into silent and non silent durations. The silent ranges are removed from the video duration to generate non silent

ranges which are used to segment the video into individual non silent segments. The ranking algorithm picks up these non-silent segments and ranks them in order based on video duration, sentence weight and word frequencies. The important segments are passed to the condenser to generate final condensed video.

3.7.2.4 Chalkboard Extractor and PDF Generator:

The basic logic to extract chalkboard representation is to subtract the lecturer's image from the video frame. The redundant ones and the ones with zero handwritten material are removed to generate important ones. The PDF generator gathers all outputs from the respective modules and generates a summarized report.

3.7.3 Conclusions:

Through this paper, we gathered necessary information about converting the transcript file for a given lecture video. We also gathered insights about extracting important contents based on the ranking algorithm mentioned.

3.8 Extensive Literature Survey:

3.8.1 Transcription:

After extensive research about transcription APIs, services and models, we came to a conclusion that we use the already existing transcription API's. This is because, to implement a transcription model from scratch using neural networks will require additional training and processing time and resources. Also, to get a high accuracy model is an even more challenging task. Our main focus in this project is video retrieval and question answering model as transcription model implementation is a whole another area which is already well established. Existing API's and services like "Google Cloud Speech-to-Text", "AWS Transcribe", "SpeechText.AI", "IBM Watson Speech to Text", "Go Transcript", "Microsoft Azure Transcript", etc. have proved to work great with high accuracy.

3.8.2 Retrieving Text from Videos:

The text retrieval from our videos is relevant in our project as power point presentations are present in the background in our lecture videos, and the topic about which the lecturer is talking can be easily generated from the text in the ppt. Text generation along with the transcript can enhance the accuracy of video retrieval as relevant keywords can now be extracted easily now and relevance can be given accordingly using both. Tesseract tool provides an easy interface for use along with a Python client library. Therefore, this tool has become the first choice for any OCR-related projects. While conducting research on this, we came across cloud services like GoogleVision, AWS Textract, and Azure OCR. The problems found in non-traditional OCR can be addressed with the progressed works in computer vision, especially the object detection field. MaskRCNN is a model which is able to very successfully perform object detection and image segmentation.

3.9 Conclusions from the Literature Survey:

Segmenting videos into different parts based on content similarity (based on OCR/transcript/both) is better than summarizing the whole video transcript in case of lengthy videos. Transcription will be done using Amazon Transcribe or other similar services like Google speech-to-text, Microsoft Azure speech-to-text etc. Transformers are better than Bi-LSTMs for processing text and gaining context. BERT is under-trained and needs some modifications for the Q/A model to process longer sentence inputs.

CHAPTER 4

DATASET

With less available datasets for video-answering questions, there was a need for a dataset from which non-factoid questions or questions requiring detailed answers can be generated. Henceforth, we have prepared two new datasets to test our model and compare results of the variations of the proposed architecture.

4.1 YouTube Dataset:

This dataset consists of 20 videos and comprises different types of informational videos related to computer science topics obtained from YouTube. The duration of the videos are in the range of 7 minutes to 38 minutes and the total duration of all the videos in the dataset is 5.2 hours. The different columns in this dataset are video_id, question, answer, start_timestamp of the answer and end_timestamp of the answer. There are in total 338 question-answer detail rows.

	Video_ID	Question	Answer	Start_timestamp	End_timestamp
0	101	What is Kubernetes?	kubernetes is an open source container orchest...	64	98
1	101	Who developed Kubernetes?	Google	69	70
2	101	Who created Kubernetes?	Google	69	70
3	101	What is the job of controller manager in Kuber...	controller manager which basically keeps an ov...	321	332
4	101	What is the job of scheduler in Kubernetes?	scheduler which is basically responsible for s...	336	360

Figure 4.1: Snapshot of YouTube Videos Dataset

4.2 Cloud Computing Course Dataset:

This dataset consists of 24 videos and comprises PESU Academy AV summary lecture videos of Cloud Computing course. All the videos in the dataset follow the same slide structure. The duration of the videos are in the range of 7 minutes to 34 minutes and the total duration of all the videos in the dataset is 7.12 hours. The different columns in this dataset are video_id, question, answer and start_timestamp of the answer. There are in total 547 question-answer detail rows.

	Video_ID	Question	Start_timestamp
0	101	What is authentication?	28
1	101	What is Keystone?	54
2	101	What is a project in Keystone?	80
3	101	What is the fundamental purpose of the keystone?	92
4	101	What does assigning role to a user or user gro...	113

Figure 4.2: Snapshot of Cloud Computing Videos Dataset

CHAPTER 5

PROJECT REQUIREMENTS SPECIFICATION

5.1 General Constraints, Assumptions and Dependencies

Issues that will minimize the choices available to the developers.

These can include the following:

1. Videos should be descriptive, short (10-30 mins), and in English.
2. To store videos and index, and process queries in real-time.
3. The simulation program will usually be a python notebook file to demonstrate the core functionality, rather than to provide a complete end-product to the user.
4. 3rd party transcription APIs would be used.
5. Video uploading, processing, and query handling would run in parallel so that they don't affect each other's performance.
6. Videos uploaded must have permission to do so and are kept protected from unauthorized access.

5.2 Risks

1. Poor authentication practices.
2. Too many privileges.
3. Video Spamming.

5.3 External Interface Requirements

The logical characteristics of the interface connecting the system and the users, includes:

1. Web pages based on Material Design.

2. React.js based single-page application.
3. Python (Flask) based APIs to respond to user queries with low latency.
4. Appropriate error messages for different situations like “Answer not found”, “Error while uploading video”, etc. are displayed whenever necessary.

5.4 Hardware Requirements

System (preferably i5+ processors with min 8GB RAM and a GPU) to run a python-based server.

SSD/HDD for storage.

5.5 Software Requirements

The relationship between the product and the software components.

For each required product the following shall be provided,

1. Python3+ with Flask, Nltk, Transformers, etc., libraries.
2. MongoDB server and Pinecone for database.
3. React.js for frontend.
4. VS Code as code editor
5. Google Collab
6. Github

CHAPTER 6

SYSTEM DESIGN

6.1 Proposed Design Methodology:

Proposed design methodology involves transcript frame text to represent video data in the system, which is used as context while answering questions. On every video, the inverted index is updated. Following activity diagrams show the steps involved:

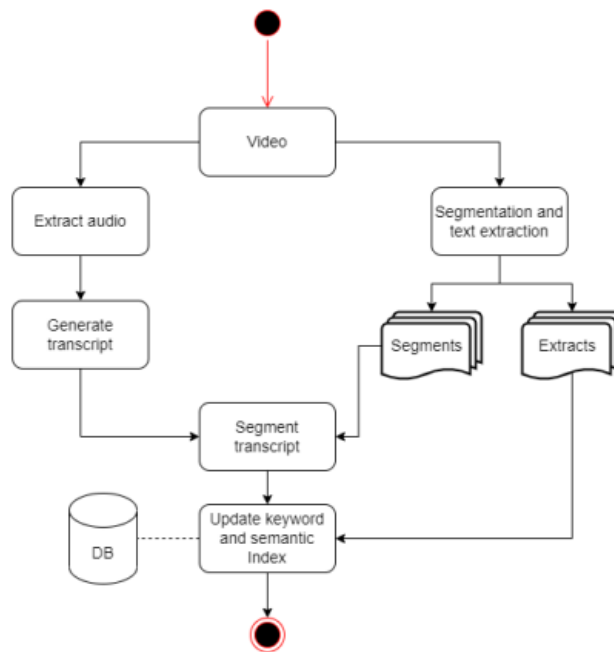


Figure 6.1a: Final methodology of video processing

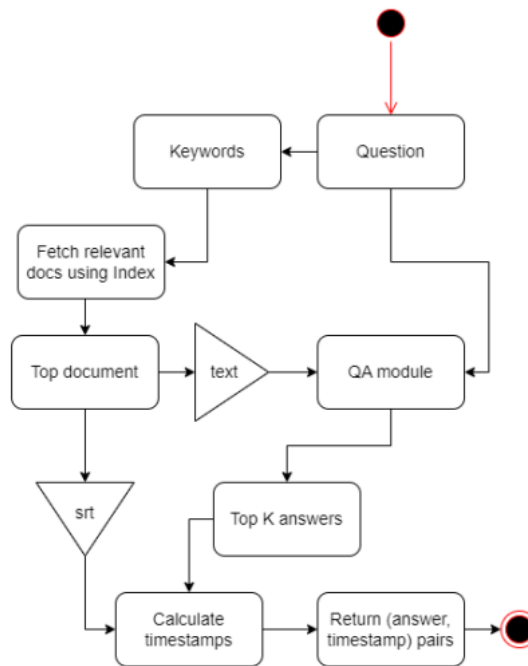


Figure 6.1b: Final methodology of query processing

6.2 High Level Design:

Video is being uploaded and stored in a storage bucket. Video Preprocessor section, then carry out necessary actions on the video mainly transcription and frame text extraction. The transcription generated is stored in another bucket, while both transcription and extracted text is used to create or update inverted indexes. Indexer section contains two modules, Inverted-indexer and Timestamp indexer to carry out the above-mentioned indexing process. Inverted index is stored in a central NoSQL database.

Once the process is complete, Transcoder transcodes the video into required formats like HLS (HTTP Live Streaming) which is a widely used format for video streaming and stores it on a transcode bucket which is associated with a CDN (Content Delivery Network).

User query is used to fetch relevant videos (searching and ranking) using inverted index from the database. Top videos' transcripts are retrieved and passed as context to the Question answering model to obtain answers.

6.2.1 Video Uploading Handling:

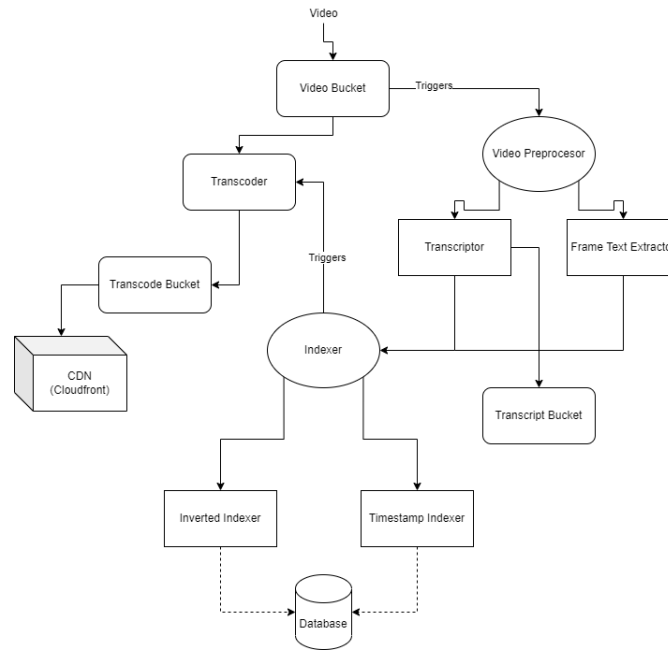


Figure 6.2.1: High Level Design of Video Handling

6.2.2 Query Handling:

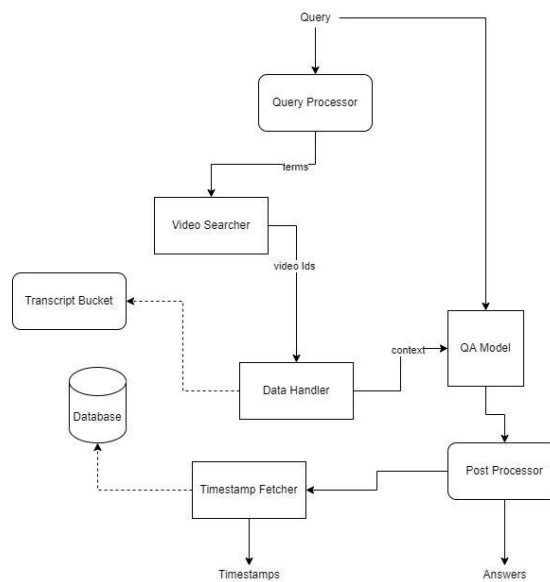


Figure 6.2.2: High Level Design of Query Handling

6.3 Use-Case Diagram:

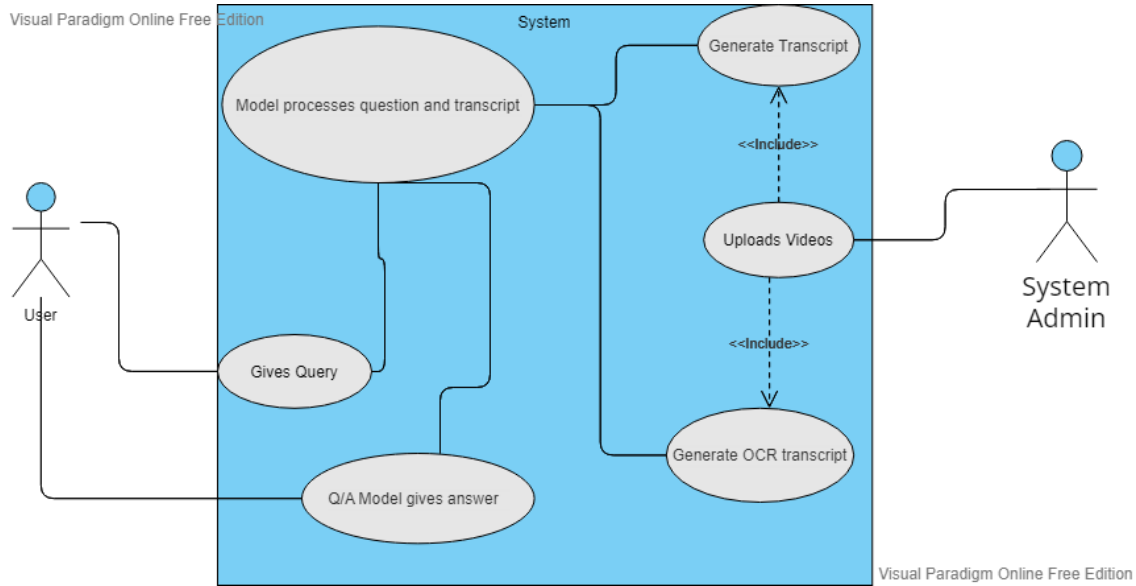


Figure 6.3: Use-Case Diagram

CHAPTER 7

PROPOSED METHODOLOGY

7.1 Video Retrieval System:

The first step of the proposed methodology is to retrieve videos from the corpus that contain the topics related to users' query and pass them to the question answering and timestamp retrieval model for extracting the answer to the questions posted by the user. In this part, the two major steps followed are automated indexing of the videos and automated segmentation of the videos. When segmentation is done on the videos, using index we retrieve only a segment of the video which is related to the users' query. Here, the video segments are treated as separate documents, whereas without segmentation, the whole video is treated as a separate document.

7.1.1 Constructing an Index of the Videos Based on the Video Content:

An index is constructed for all the videos in the transcript based on the contents like transcript and text extracted from various frames of the videos. This helps in the video retrieval part of the project where appropriate videos that contain the topic related to the users' query are retrieved and passed to the question-answering model. In our work, we have proposed three different indexes: i) Based on only transcript; ii) Based on only text extracted from video frames; and iii) Based on both transcript and text extracted from video frames combined.

i) Index based on Transcript of the video:

Firstly, the transcript of the videos is generated using Azure cloud speech to text API. The transcript of each video is stored in the database using which an index is created. The transcript-based index has further three more variations:

i) Keyword index using TF-IDF approach, ii) Semantic index using cosine similarity and vector embeddings, and iii) Combined index of keyword and semantic.

ii) Index based on text extracted from the video frames:

Another variant of index is generated using text extracted from video frames. Text extraction from the video frames is performed through the OpenCV model and Google Tesseract engine. Keywords from the extracted text are used to build the index.

iii) Index based on both transcript and text extracted from video frames:

The last variant of index is built by combining both index based on transcript and index built using text extracted from frames.

7.1.2 Automated Segmentation of the Videos:

Automated segmentation of the videos is performed to further increase the performance of the video retrieval system. Results are generated and compared when the index is used with and without the segmentation of the videos. In our work, we propose three different kinds of automated segmentation of the videos and are discussed below.

i) Segmentation based on title:

From the text extracted from the slides in the video frames, the title portion is extracted and non-similar titles are chosen to segment the whole video.

ii) Segmentation based on similarity:

A video is divided into non-overlapping continuous segments based on the cosine similarity of the content on video frames.

iii) Constant length segmentation:

A video is divided into segments of equal length of 1 minute.

7.2 Question-Answering and Timestamp Retrieving Model:

The top video document or video segment document related to the users' query from the video retrieval system is passed to the question answering and time-stamp retrieval model. Given the context and the question by the user, this model generates top k answers along with the text span, video span and is returned to the users.

We have used two different question-answering models:

- i) distilbert-base-cased-distilled-squad: DistilBERT is a small, fast, cheap and light Transformer model. It is ~260 MB in size.
- ii) deepset/xlm-roberta-large-squad2: XLM-RoBERTa is a multilingual version of RoBERTa. It is pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages, further trained on squad2. It is ~2.2GB in size.

CHAPTER 8

IMPLEMENTATION AND PSEUDOCODE

The whole project is implemented using **Python 3.10.6** on VS code editor. The project is divided into different components based on the tasks and are implemented as **class-based modules**. All these modules are used in a class of project.py file to provide the final interface for the proposed use-case.

8.1 Video Transcript Generation:

The libraries used are Azure Cloud speech to text and like-a-srt. A script was developed to convert .mp4 videos to .wav audio format. Using like-a-srt python package and Azure SDK, the .wav audio format was sent to Azure speech2text instance to get .srt files. This .srt (SubRip Subtitle File) contains the audio transcript from videos with text and timestamps range corresponding to it.

8.2 Keyword Index:

The libraries used are Nltk and Pymongo. The keyword index is generated from the transcript of the videos and is based on the TF-IDF approach. MongoDB is used to store the terms and its bulk operations are used for more efficiency.

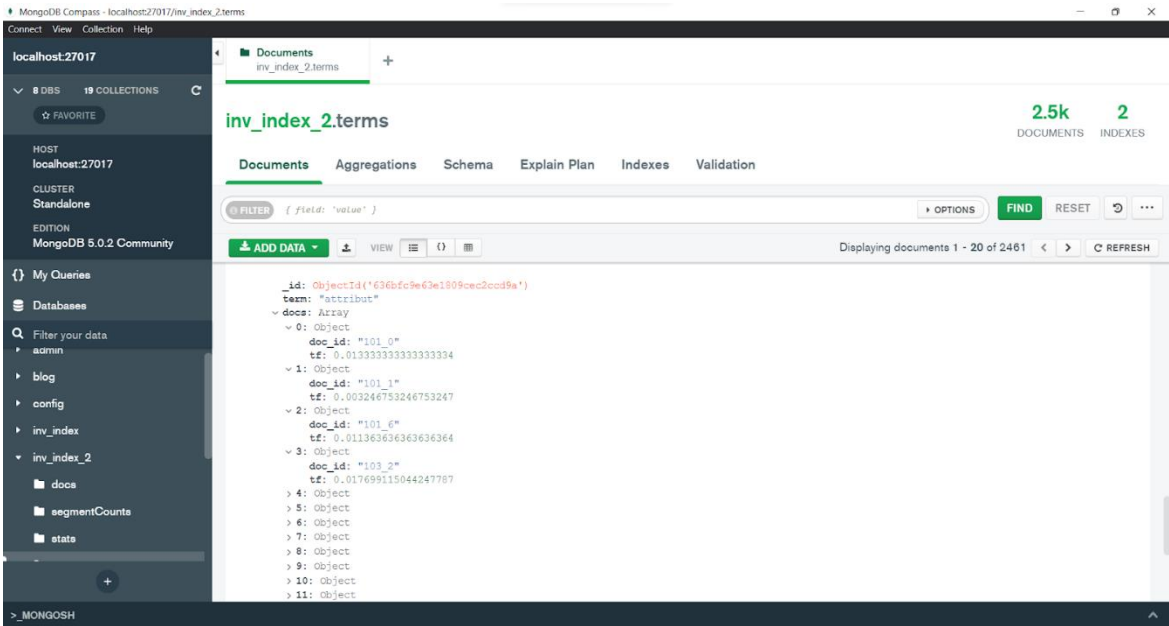


Figure 8.2a: Database schema used

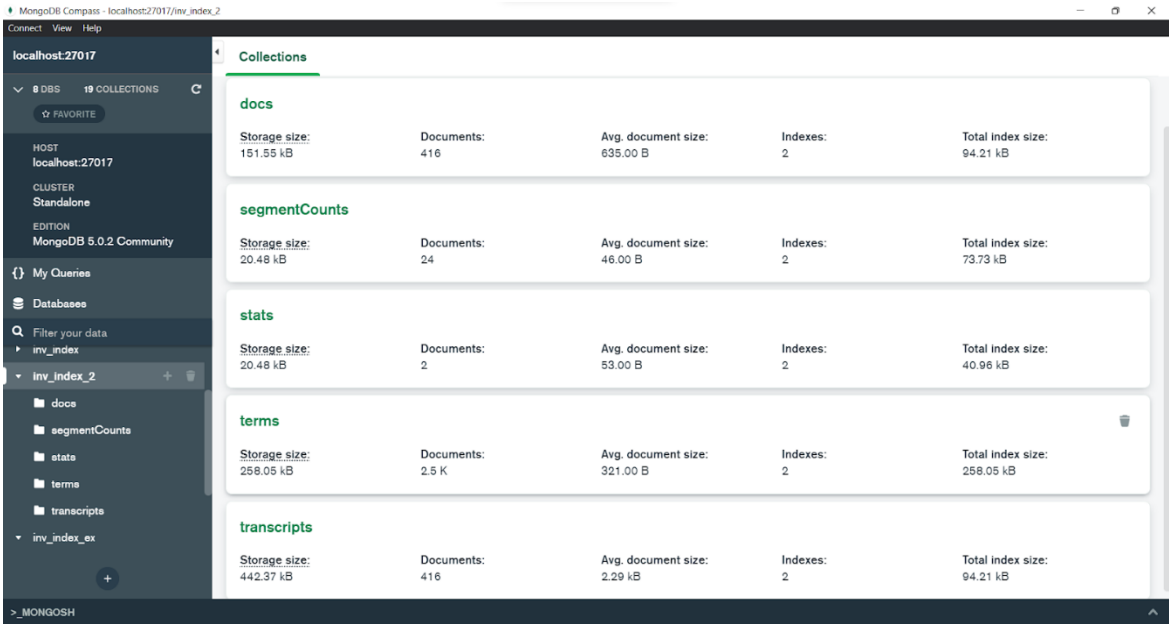


Figure 8.2b: Overview of database storage

8.3 Semantic Index:

The libraries used are sentence transformers and pinecone. The text is converted into a vector using multi-qa-mpnet-base-cos-v1 model. The vectors are stored on a pinecone database and are retrieved based on cosine similarity to the query vector.

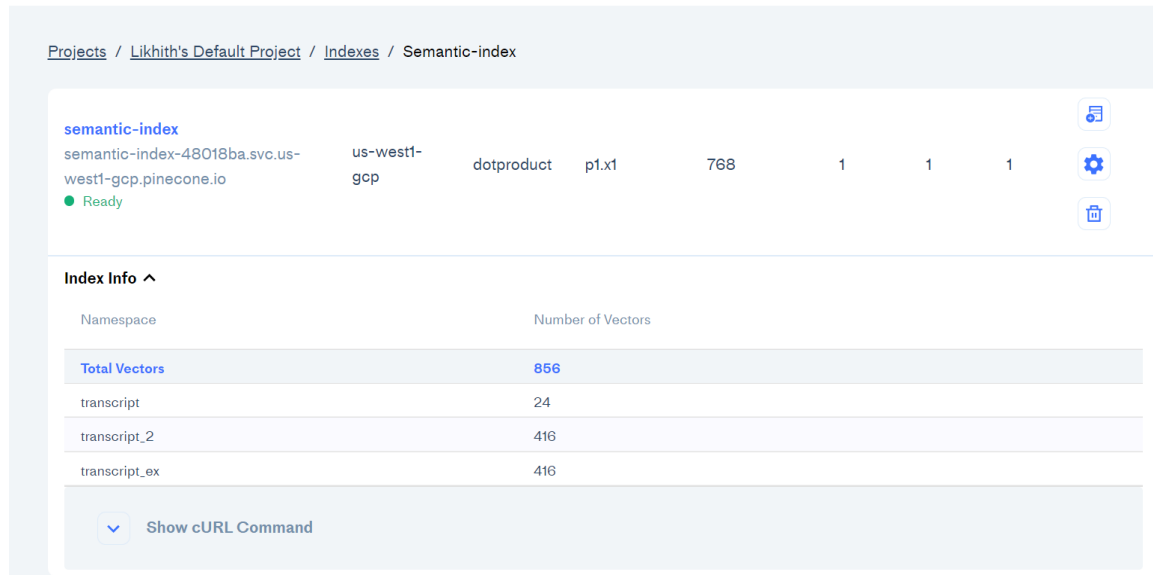


Figure 8.3: Overview of pinecone database

8.4 Title Based Segmentation:

The libraries used are numpy, pytesseract, threading, sentence_transformers and cv2 (Open CV). The video processing task is divided equally among a specified number of threads. Each thread reads a frame per specified interval (4 s) starting from its assigned start frame to end frame. The frame is cropped according to the position of the title, and the same is extracted using pytesseract.image_to_string(). The same steps are followed on the body part of the frame to extract the body-text. To avoid adding the same body text multiple times to a segment, cosine similarity is used to check if text to be added is different. It's done by checking if cosine similarity between prev-text vector and cur-text vector is less than specified threshold (0.80). Each title is mapped to start and end time (segment), which is based on when a title changes while reading frames. Once all the threads complete their part

of work, post processing of the collected data is done to join the segments with empty titles, remove overlapped segments, etc.,

8.5 Similarity Based Segmentation:

The libraries used are pytesseract, sentence transformers and cv2(OpenCV). The frames of video are being read at a rate of 1 frame per 4 second (0.25 fps). Using pytesseract.image_to_string, all the text present in the frame is extracted and preprocessed. The text is vectorized using a sentence transformers model and encoded into a vector of 768 dimensions. The dot product of the vector with the previous text vector(of previous frame)is taken. Using cosine similarity of two vectors, we segment them into separate categories if similarity < minimum similarity. If the similarity is \geq minimum similarity (usually 0.8) then we merge the text into one segment. After segmentation we merge those segments with length less than 40s to become one large segment with length ≥ 40 s.

8.6 Question-Answering and Timestamp Retrieving Model:

The libraries used are transformers and torch. Given the context and a question, top k answers along with text span is extracted from context and returned.

The models used are:

- i) distilbert-base-cased-distilled-squad
- ii) deepset/xlm-roberta-large-squad2

8.7 Implementation of Different Variations of Proposed Architecture:

To compare results and performances, different variations of proposed architecture are implemented and tested.

8.7.1 Base:

The whole video transcript is considered as a single document and is indexed. On query, top document is retrieved and passed to QA model to extract answer.

8.7.2 V1:

A video is divided into non-overlapping continuous segments based on slide-title/content-similarity. Transcripts corresponding to each segment are treated as individual documents and next things proceed as base. 101.srt -> 101_0.srt + 101_1.srt + 101_2.srt ...

8.7.3 V2:

This is similar to V1, but here the slide-texts are also extracted during the segmentation process. Both the transcript and extracted-text corresponding to each segment are treated as individual documents and are indexed in two separate indexes. Results from both the indexes are combined based on the assigned weights for each index on query, before proceeding.

101.srt -> 101_0.srt + 101_1.srt + 101_2.srt ... Index-1

101.mp4 -> 101_0.txt, 101_1.txt, 101_2.txt ... Index-2

8.7.4 V3:

A video is divided into non-overlapping continuous segments of 1 minute each. Rest is the same as the V2.

8.8 Metrics:

8.8.1 Confusion Matrix:

It is used to evaluate and compare video retrieval results comprised of actual_videoID and predicted_videoID. This is usually calculated for binary (true/false) values. But here we have multiple videoIDs in the dataset. So, here for each videoID, a confusion matrix is calculated by taking that specific videoID as true and rest as false. Then, the weighted average is taken to calculate the overall confusion matrix.

True Positive, TP: expected, predicted => count predicted

False Negative, FN: expected, not predicted = expected - TP

False Positive, FP: not expected, predicted => count predicted when not expected

True Negative, TN: not expected, not predicted = (total - expected) - FP

Above values are converted into percentage by multiplying them with 100 and dividing them by expected and not-expected counts depending on the value.

8.8.2 Effective Error:

It is used to evaluate and compare timestamp retrieval results. An error here indicates how far away a predicted answer is from the actual answer in the video. It could be in seconds or in minutes but is represented in terms of percentage of video to have uniformity and to address the fact that different videos have different duration. For example, a minute error in a 5-minute video is more significant than it is in a 30-minute video.

A user here, is presented with both the bestAnswer and longAnswer each of them having different errors. Sometimes bestAnswer will be closest to actual answer and other time it could be longAnswer. Since the choice to select the closest out of both lies with the user, Effective-Error is defined this way:

*bestAnswer_error = abs(actual_start - bestAnswer_start) * 100 / duration*

*longAnswer_error = abs(actual_start - longAnswer_start) * 100 / duration*

min_error = min(bestAnswer_error, longAnswer_error)

$$\text{max_error} = \max(\text{bestAnswer_error}, \text{longAnswer_error})$$
$$\text{Prob_to_pick_closest} = 0.5$$
$$\text{Effective-Error} = \text{min_error} * (\text{prob_to_pick_closest}) + \text{max_error} * (1 - \text{prob_to_pick_closest})$$

0.5 is taken as a probability that an average user chooses the closest answer. But in practice, this would be greater than 0.5 as he can make better decision based on the answer text showed and hence the lesser error. The Average Effective-Error vs No. of Answers graph provides us the better visuals to understand the change in error with the percentage of answers being considered, ignoring the rest as outliers.

CHAPTER 9

RESULTS AND DISCUSSION

9.1 YouTube Dataset Results:

The results tested on the YouTube dataset based on all the variations of the proposed architecture model are presented and discussed below:

9.1.1 Confusion Matrices, Effective-Error Graphs and Accuracy Table:

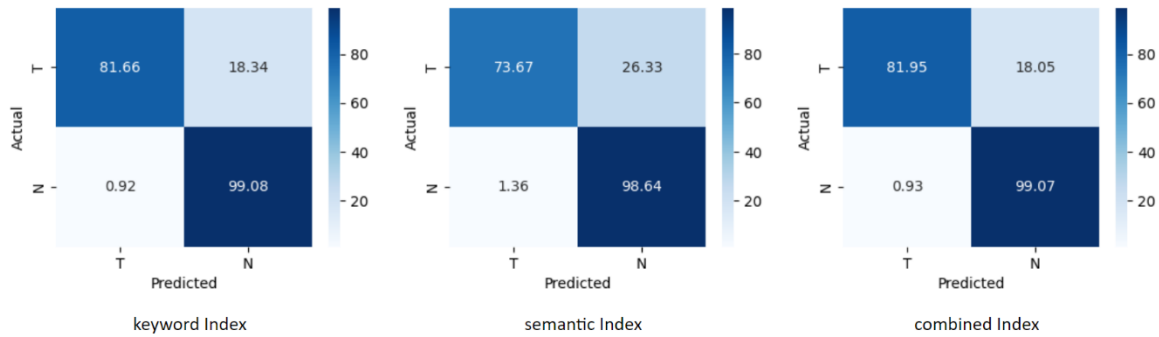


Figure 9.1.1a: Base- No segmentation, only transcript based index on Yt Dataset

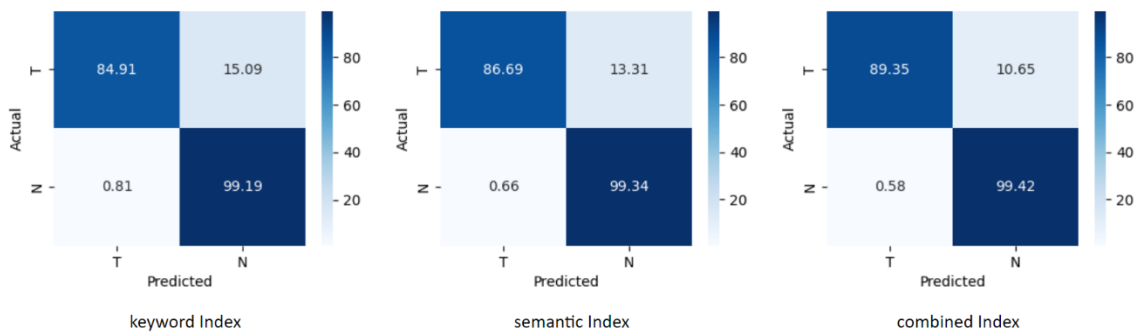


Figure 9.1.1b: Similarity based segmentation, only transcript based index on Yt Dataset

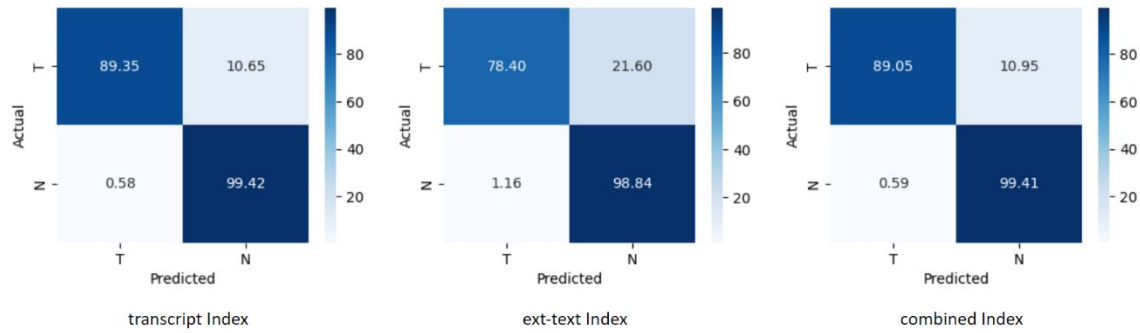


Figure 9.1.1c: Similarity based segmentation, transcript and extracted-text based index on Yt Dataset

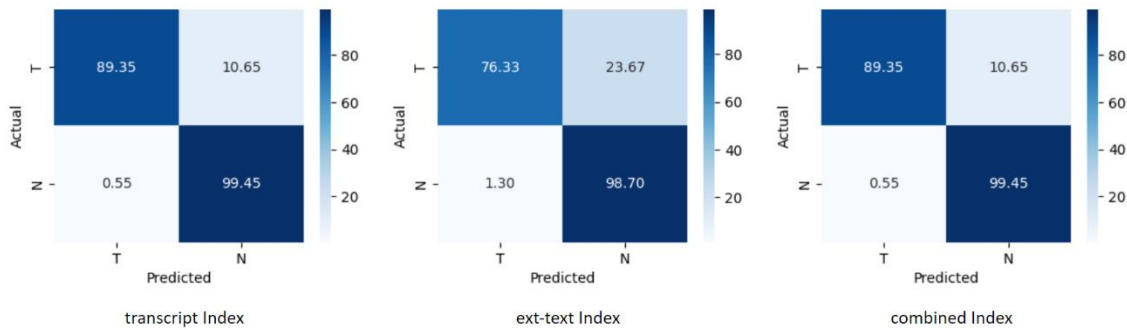


Figure 9.1.1d: 1-minute length segmentation, transcript and extracted-text based index on Yt Dataset

DistilBert Model: Effective-Error

- No assistance
- Base
- v1 and v2
- v3
- v3, no QA

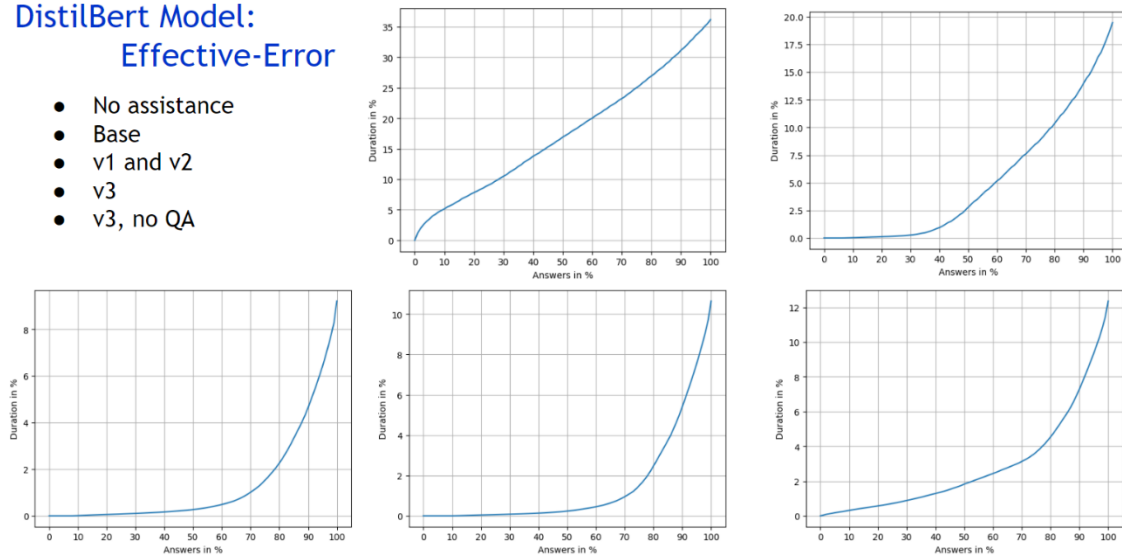


Figure 9.1.1e: DistilBERT Model: Effective-Error on Yt Dataset

Roberta Model: Effective-Error

- No assistance
- Base
- v1 and v2
- v3
- v3, no QA

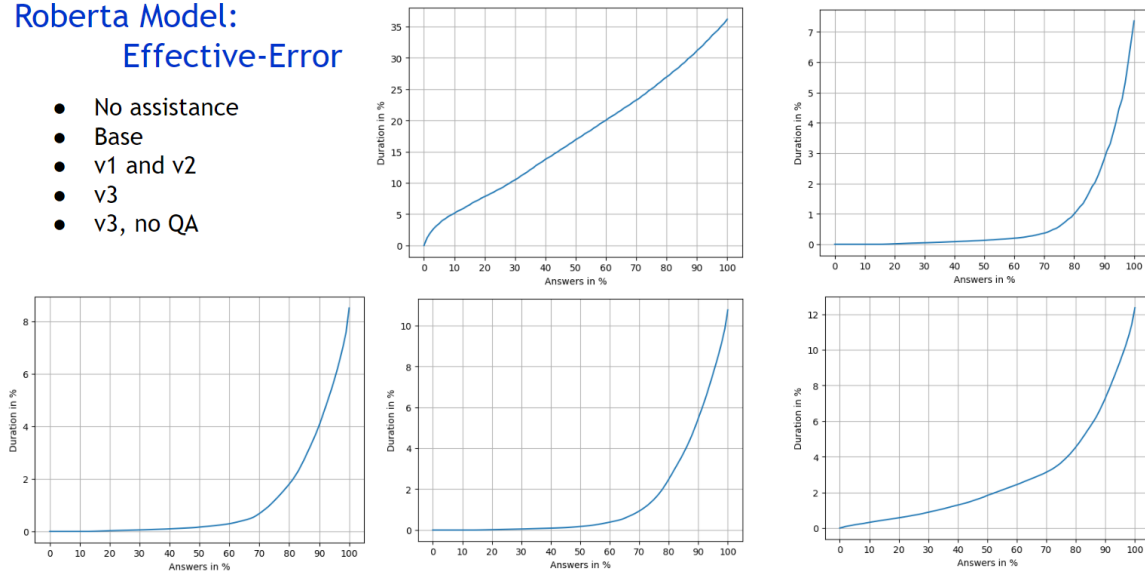


Figure 9.1.1f: RoBERTa Model: Effective-Error on Yt Dataset

Youtube Video Dataset	Accuracy (in percent)		
	Keyword Index	Semantic Index	Combined Index
Base Model	81.66	73.67	81.95
Similarity-based segmentation, Only transcript-based Index	84.91	86.69	89.35
	Transcript Index	Extracted Text Index	Combined Index
Similarity-based segmentation, Transcript and Extracted-text based Index	89.35	78.4	89.05
1-minute-length segmentation, Transcript, and Extracted-text based Index	89.35	76.33	89.35

Table 9.1.1: Accuracy of different models with Yt Dataset

9.1.2 Observations:

Here accuracy improved from 81.95% to 89.35% from base to v3. There is not much change observed in the accuracy from v1 to v3 variation (equal around 89%) while it was expected to increase. This could be due to the following reasons: The video segments in v1 had almost same length as segments in v3 i.e., around 40s-100s in v1, while v3 segments were around 60s. In v3, transcript index alone had much higher accuracy that there was very less room for any improvements. Semantic index gave very good results in v1 than in base variation, as the segments were made based on content similarity using the same text-to-vector model that was used in the semantic indexing.

Without any assistance, a user will be around 27% of video away from the desired part of the it for 80% of the questions assuming he/she starts watching video from the start. The best performance of RoBERTa-based model is just around 1% and the best performance of DistilBERT-based model is around 2.3% away, for 80% of the questions. For example, approximate error in results for a video of 30 mins would be 486s without any assistance, 18s with RoBERTa-based model and 42s with DistilBERT-based model.

9.2 Cloud Computing Dataset Results:

The results tested on the cloud computing dataset based on all the variations of the proposed architecture model are presented and discussed below:

9.2.1 Confusion Matrices, Effective-Error Graphs and Accuracy Table:

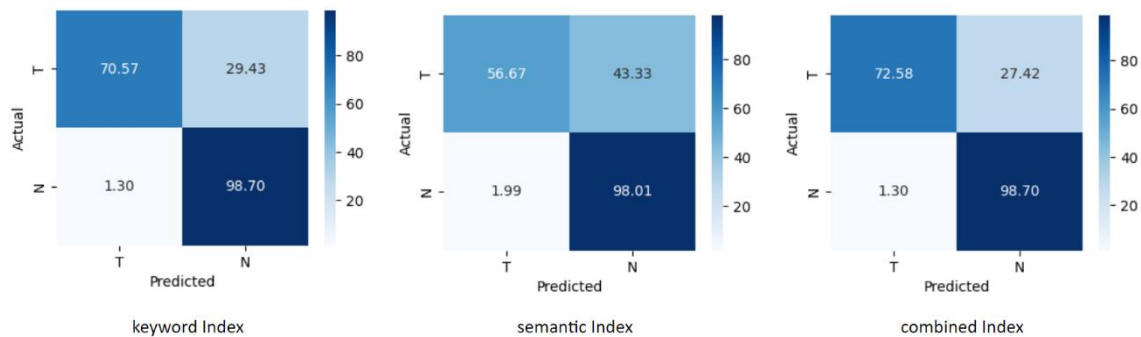


Figure 9.2.1a: Base- No segmentation, only transcript based index on CC Dataset

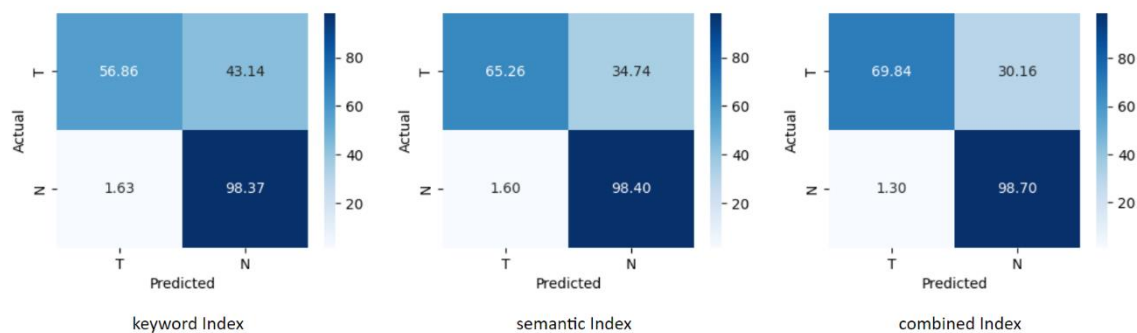


Figure 9.2.1b: Similarity based segmentation, only transcript based index on CC Dataset

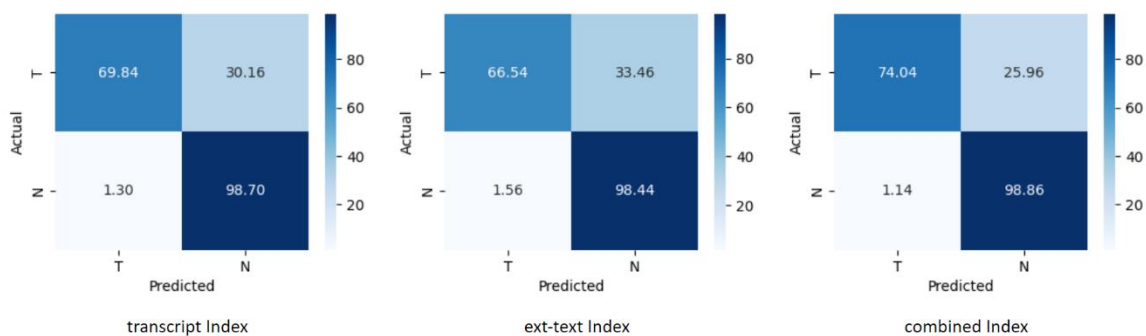


Figure 9.2.1c: Similarity based segmentation, transcript and extracted-text based index on CC Dataset

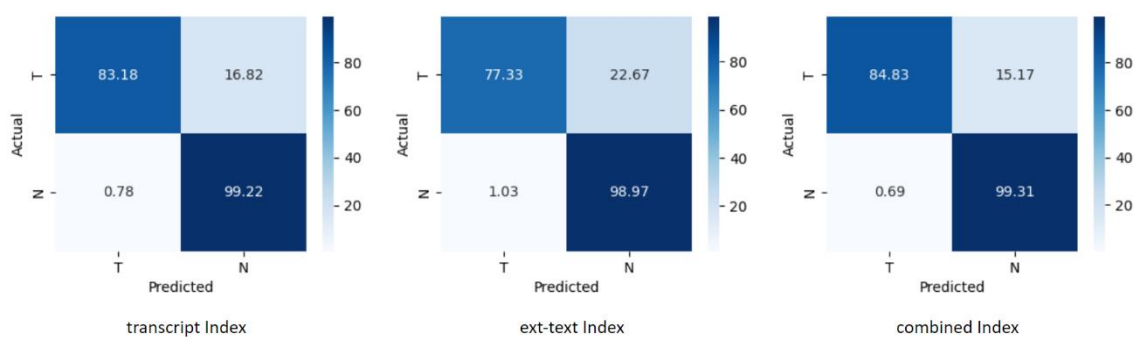


Figure 9.2.1d: 1-minute length segmentation, transcript and extracted-text based index on CC Dataset

DistilBert Model: Effective-Error

- No assistance
- Base
- v1 and v2
- v3
- v3, no QA

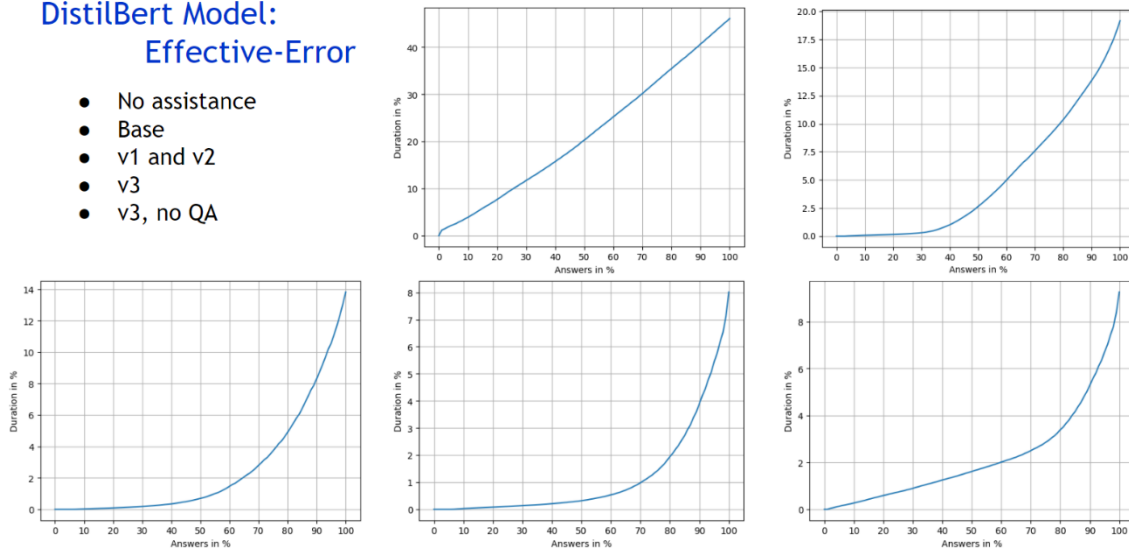


Figure 9.1.1e: DistilBERT Model: Effective-Error on CC Dataset

Roberta Model: Effective-Error

- No assistance
- Base
- v1 and v2
- v3
- v3, no QA

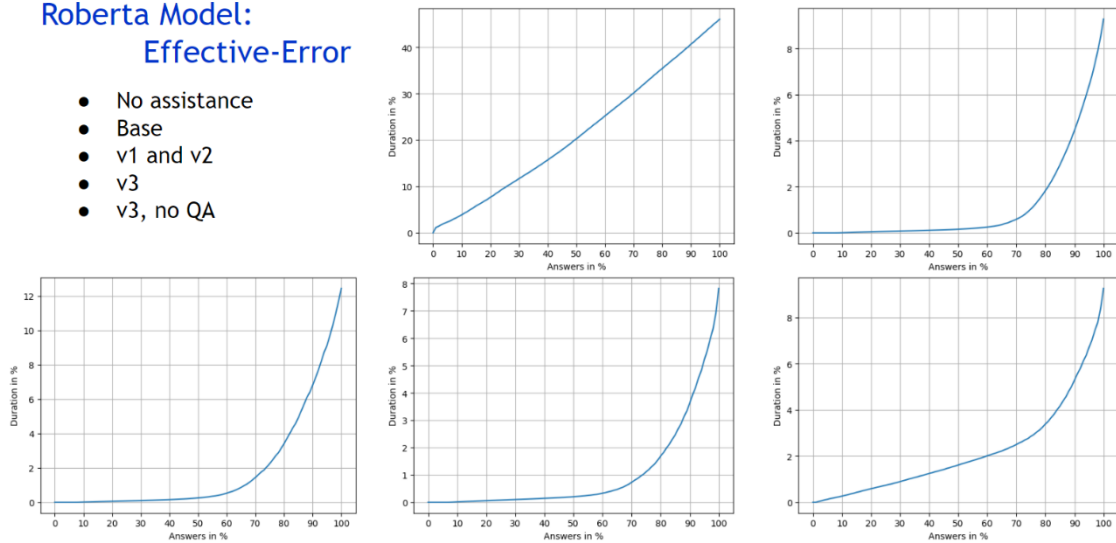


Figure 9.1.1f: RoBERTa Model: Effective-Error on CC Dataset

Cloud Computing Dataset	Accuracy (in percent)		
	Keyword Index	Semantic Index	Combined Index
Base Model	70.57	56.67	72.58
Title-based segmentation, Only transcript-based Index	56.86	65.28	69.84
	Transcript Index	Extracted Text Index	Combined Index
Title-based segmentation, Transcript and Extracted-text based Index	74.04	66.54	74.04
1-minute-length segmentation, Transcript, and Extracted-text based Index	83.18	77.33	84.83

Table 9.2.1: Accuracy of different models with CC Dataset

9.2.2 Observations:

Here accuracy improved from 72.58% to 84.83% from base to v3. Drop of accuracy by around 2.7% from base to v1 variation is observed, mainly because Keyword index accuracy dropped from 70.57% to 56.86% while it was expected to increase. This could be due to the mismatch in time of topic change on the slides and in the explanation (transcript). But the main reason is, segments in v1 vary from a length of a few seconds to almost 10-15 mins. Because of the presence of the same keywords and topics in multiple videos, segments with smaller length get more importance in the keyword index, even if the longer ones are more relevant. This is one of the huge drawbacks of TF-IDF based index.

To support this claim, further analysis was made and following are the results that validates our reason: Among 43.14% (100 – accuracy) false segment predictions made by the keyword index in v1 variation, around 93% were smaller than the true segments. It was

found that around 80% of them were at least 4.25 times smaller than true segments and at least 20% of the predicted segments were even 40 times smaller than the true segments (fig. 8). Anyways, when extracted text (slide text) is indexed along with transcript index, the combined result (v2 variation) outperforms the base by 4.2%.

Without any assistance, a user will be around 35.44% of video away from the desired part of the it for 80% of the questions assuming he/she starts watching video from the start. The best performance of RoBERTa-based model is just around 1.7% and the best performance of DistilBERT-based model is around 1.9% away, for 80% of the questions. For example, approximate error in results for a video of 30 minutes would be 638s without any assistance, 31s with RoBERTa-based model and 34s with DistilBERT-based model.

9.3 Overall Timestamp Retrieval Results:

	Youtube Video Dataset		Cloud Computing Dataset	
	Effective Error (in percentage of video length)			
Variation	80% of answers	100% of answers	80% of answers	100% of answers
No assistance	26.9	36.2	35.44	46.08
base	10.4	19.5	10.38	19.17
V1	2.26	9.21	4.9	13.8
V2	2.26	9.21	4.9	13.8
V3	2.48	10.65	1.94	8.0
V3, without answer extraction (no QA module)	4.55	12.36	3.4	9.26

Table 9.3a: DistilBERT Model: Effective Error Table

	Youtube Video Dataset		Cloud Computing Dataset	
	Effective Error (in percentage of video length)			
Variation	80% of answers	100% of answers	80% of answers	100% of answers
No assistance	26.9	36.2	35.44	46.08
Base	1.0	7.36	1.83	9.28
V1	1.8	8.5	3.4	12.4
V2	1.8	8.5	3.4	13.4
V3	2.48	10.76	1.69	7.83
V3, without answer extraction (no QA module)	4.55	12.36	3.4	9.26

Table 9.3b: RoBERTa Model: Effective Error Table

CHAPTER 10

CONCLUSION AND FUTURE WORK

We performed experiments with video transcript-based index and the segmentation of videos either based on text similarity of different slides or based on slide titles. Segmentation of videos into smaller parts of equal length, 1 minute was also done. We also considered using a secondary index based on texts extracted from the video frames. All these led to different variations of implementation architecture, each of which were evaluated for performance in keyword index, semantic index and combined index which involved results from both indexes and the timestamps retrieval results.

Combination of keyword index and semantic index together performs better than when they are taken individually for most of the times. Video retrieval results get better when videos are made into smaller partitions, preferably of same size. When the context length is more, Roberta based Question-answering module performs better than DistilBERT based module. However, DistilBERT tends to increase performance on segmented videos as the context length gets smaller. When videos are partitioned into smaller and equal segments, the most error-possible work gets shifted to Indexing module as, once after a right segment is chosen by the Indexing component on query, the max possible error by the Question-answering component would only be 3.3% for a 30-minute length video, with 1-minute segments.

According to us, the best approach would be to use v3 methods for indexing and base methods with RoBERTa based Question-answering module for answer and there by the timestamp extraction. However, since DistilBERT is faster than RoBERTa and takes very less space in comparison, v3 methods can be used with DistilBERT if the availability of computer resources is limited, without compromising much of the performance. Further, more experiments can be conducted with different other variations, for example, overlapped segments, noun substituted pronouns etc. to get more insights and we believe that this paper will provide a basis for that.

REFERENCES

- [1]Medida, Lakshmi Haritha, and K. Ramani. "An optimized E-lecture video retrieval based on machine learning classification." *Int. J. Eng. Adv. Technol. IJEAT* 8.6 (2019): 4820-4827.
- [2]H. Yang and C. Meinel, "Content Based Lecture Video Retrieval Using Speech and Video Text Information," in *IEEE Transactions on Learning Technologies*, vol. 7, no. 2, pp. 142-154, April-June 2014, doi: 10.1109/TLT.2014.2307305.
- [3]Lin-Qin Cai; Min Wei; Si-Tong Zhou; Xun Yan, "Intelligent Question Answering in Restricted Domains Using Deep Learning and Question Pair Matching" *IEEE Access* (Volume: 8), pp.32922 - 32934, 13 Feb 2020.
- [4]Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 2018 Oct 11.
- [5] }Anthony Colas, Seokhwan Kim, Franck Dernoncourt, Siddhesh Gupte, Zhe Wang, and Doo Soon Kim. 2020. TutorialVQA: Question Answering Dataset for Tutorial Videos. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5450–5455, Marseille, France. European Language Resources Association.
- [6]H. Yang, M. Siebert, P. Luhne, H. Sack and C. Meinel, "Automatic Lecture Video Indexing Using Video OCR Technology," 2011 *IEEE International Symposium on Multimedia*, 2011, pp. 111-116, doi: 10.1109/ISM.2011.26.
- [7]Ashwin, Siddharth S., et al. "Summarization of video lectures." *Artificial Intelligence and Speech Technology*. CRC Press, 2021. 149-158.
- [8]Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).
- [9] Sanh, Victor, Lysandre Debut, Julien Chaumond and Thomas Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *ArXiv abs/1910.01108* (2019): n. pag.

- [10] Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. "Unsupervised cross-lingual representation learning at scale." arXiv preprint arXiv:1911.02116 (2019).
- [11] Smith, Ray. "An overview of the Tesseract OCR engine." In Ninth international conference on document analysis and recognition (ICDAR 2007), vol. 2, pp. 629-633. IEEE, 2007
- [12] Ashwin, Siddharth S., et al. "Summarization of video lectures." Artificial Intelligence and Speech Technology. CRC Press, 2021. 149-158
- [13] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237
- [14] Song, Kaitao, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. "Mpnet: Masked and permuted pre-training for language understanding." Advances in Neural Information Processing Systems 33 (2020): 16857-16867.

WEB LINKS REFERRED:

<https://azure.microsoft.com/en-gb/services/cognitive-services/speech-to-text/?cdn=disable>

<https://hacks.mozilla.org/2018/09/speech-recognition-deepspeech/>

<https://aws.amazon.com/cloudfront/streaming/>

ORIGINALITY REPORT

14%

SIMILARITY INDEX

9%

INTERNET SOURCES

8%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Indian Institute of Science, Bangalore Student Paper	2%
2	arxiv.org Internet Source	2%
3	scholarbank.nus.edu.sg Internet Source	1%
4	www.ijeat.org Internet Source	1%
5	search.computer.org Internet Source	1%
6	aclanthology.org Internet Source	1%
7	Haojin Yang, Christoph Meinel. "Content Based Lecture Video Retrieval Using Speech and Video Text Information", IEEE Transactions on Learning Technologies, 2014 Publication	<1%

8

"An Optimized E-Lecture Video Retrieval based on Machine Learning Classification", International Journal of Engineering and Advanced Technology, 2019

Publication

<1 %

9

www.semanticscholar.org

Internet Source

<1 %

10

Hamid Hassani, Mohammad Javad Ershadi, Azadeh Mohebi. "LVTIA: A new method for keyphrase extraction from scientific video lectures", Information Processing & Management, 2022

Publication

<1 %

11

Submitted to CSU, Long Beach

Student Paper

<1 %

12

Nikola Milošević, Wolfgang Thielemann. "Comparison of biomedical relationship extraction methods and models for knowledge graph creation", Journal of Web Semantics, 2023

Publication

<1 %

13

Sandeep Varma, Arunanshu Pandey, Shivam, Soham Das, Soumya Deep Roy. "Chapter 14 Video Indexing System Based on Multimodal Information Extraction Using Combination of ASR and OCR", Springer Science and Business Media LLC, 2022

Publication

<1 %

14	Haojin Yang. "Automatic Lecture Video Indexing Using Video OCR Technology", 2011 IEEE International Symposium on Multimedia, 12/2011 Publication	<1 %
15	huggingface.co Internet Source	<1 %
16	ebin.pub Internet Source	<1 %
17	www.gwern.net Internet Source	<1 %
18	"ECAI 2020", IOS Press, 2020 Publication	<1 %
19	digitalcommons.unl.edu Internet Source	<1 %
20	Submitted to The University of Manchester Student Paper	<1 %
21	Konstantin Lomakin, Saif Alhasan, Gerald Gold. "Additively Manufactured Amplitude Tapered Slotted Waveguide Array Antenna With Horn Aperture for 77 GHz", IEEE Access, 2022 Publication	<1 %
22	"Digital Libraries at Times of Massive Societal Transition", Springer Science and Business	<1 %

23	Chinmaya Misra. "Content Based Image and Video Retrieval Using Embedded Text", Lecture Notes in Computer Science, 2006 Publication	<1 %
----	--	------

24	joelchoe.medium.com Internet Source	<1 %
----	--	------

25	www.semantic-web-journal.net Internet Source	<1 %
----	---	------

26	shreyansh26.github.io Internet Source	<1 %
----	--	------

27	www.iosrjournals.org Internet Source	<1 %
----	---	------

28	Ma, Di, and Gady Agam. "Enhanced features for supervised lecture video segmentation and indexing", Imaging and Multimedia Analytics in a Web and Mobile World 2015, 2015. Publication	<1 %
----	--	------

29	Submitted to The Hong Kong Polytechnic University Student Paper	<1 %
----	--	------

30	eudl.eu Internet Source	<1 %
----	----------------------------	------

31	krishikosh.egranth.ac.in	
----	--------------------------	--

Internet Source

<1 %

32

psasir.upm.edu.my

Internet Source

<1 %

33

"Advances in Information Retrieval", Springer
Science and Business Media LLC, 2020

Publication

<1 %

Exclude quotes On

Exclude bibliography On

Exclude matches < 5 words