NC STATE UNIVERSITY

ReadMe2Video

"Turning readmes into compelling visual stories"

Generative AI for Software Engineering

Aditya Chitlangia | Firasat Hussain Mohammed | Nikhilesh Cherukuri | Pankaj Manoharlal Thakur

Agenda

1. Problem Statement
2. Focused Challenge
3. State of the Art
4. Key Idea
5. Experiment Setting
6. Experiment Result
7. Findings
8. Conclusion
9. Lessons Learnt

A picture speaks a **thousand** words.
A video speaks **1000x24fps** words.

Problem Statement

Readmes are not always:

1. Intuitive
2. Easy-to-understand
3. Organized and Navigable
4. Fun and compelling

Focused Challenges

1. Collection of course description-specific README Dataset.

2. Generating Domain specific accurate summaries from the dataset

3. Non - existence of open source text to video tool

State of the Art Solutions

1. For Categorizing Contents of Github README

Categorizing the Content of GitHub README Files [2]

- Employs an SVM-based classifier on the manually collected annotated dataset to achieve an F1 score of 0.746

- Limitations: The precision, recall and F1 score can be improved. The solution does not focus on domain specific README files

2. For summarizing Large Document Files

Hybrid Long Document Summarization Using C2F-Far And ChatGPT: A Practical Study [3]

- Combines C2F-FAR's ability to extract important sentences from texts and ChatGPT's zero-shot ability to summarize and paraphrase texts. The results measured against the existing automated evaluation metrics are as good as those written by humans, especially when the ROUGE-1 score is taken into account.

- Limitations: Critical issues were identified in the human evaluation in terms of text coherence, faithfulness and style. ChatGPT is not open source

Key Idea

What we are trying to achieve?

Summarizing a domain specific readme into a predefined template using open source LLM

Creating a keyword driven video pipeline for the generated summary.

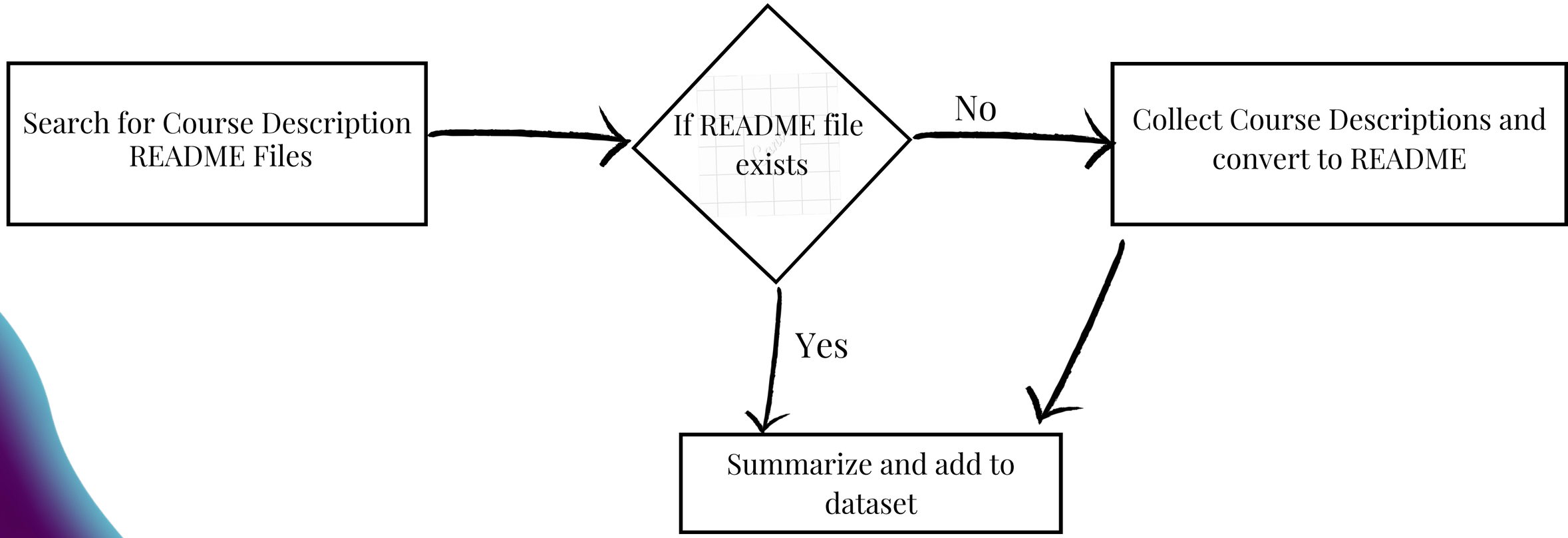
Experiment Setting

Leading Questions?

1. How can we collect our own dataset?
2. What open-source models will give us the desired results?
3. What evaluation metric would be the most suitable for our problem statement?

Dataset

We collected our own Dataset.



README Template

ReadMe2Video

Description

{a brief description of the course}

Course Content

{syllabus of the course and a brief of topics that will be covered}

Grading

{info regarding the grading, homeworks and grades, exams, midterms etc}

Prerequisites

{prerequisites of the course if any}

Office hours

{time and days at which students can meet the professor}

Location/time

{location and time of the course}

Class Structure

{include if there are going to be seminars, lectures, homeworks, assignments etc}

Teaching assistant

{info/additional info: any info related to the teaching assistant's office hours, and other info related to teaching assistant}

Communication

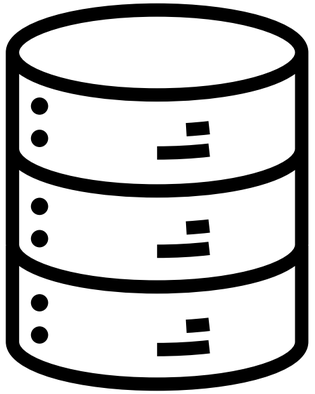
{Any communications, ways to reach the professor or any links or grps they can join}

Additional info

{any additional info relevant to the course and course success criteria}

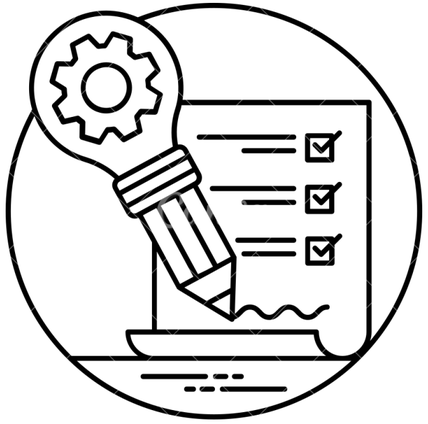
Our Project in Stages

Stage 1



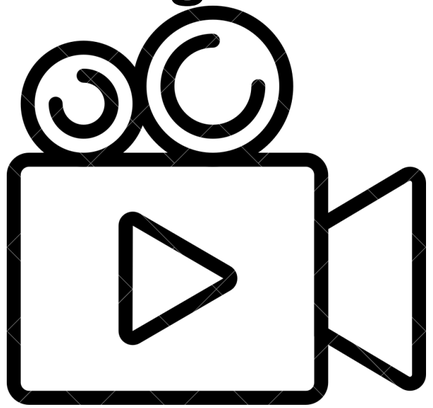
Dataset Collection

Stage 2



Summary Pipeline

Stage 3



Video Pipeline

Workflow

Falcon 7B
{Pre-Trained}

Instruction Tuning

ReadMe

Summarized Text

Post-Processing

Formated Ouput

- Sentence
- Keywords
- Metadata

Video Generation Pipeline

COMBINING
- Key-Words video frames
- Audio Narration

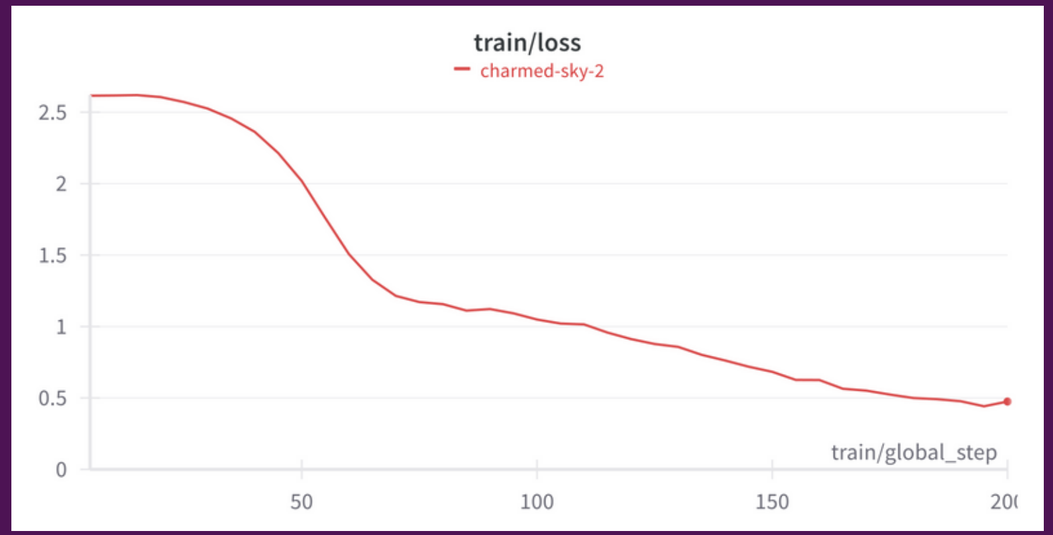
Visual Representation of the ReadME

Instruction Tuning

- Fine-tuned on 225 quality data samples
- Used Falcon 7B with fp16 precision
- Converted lm_head and small parameters to fp32 for stability and trained lora adapters on it.

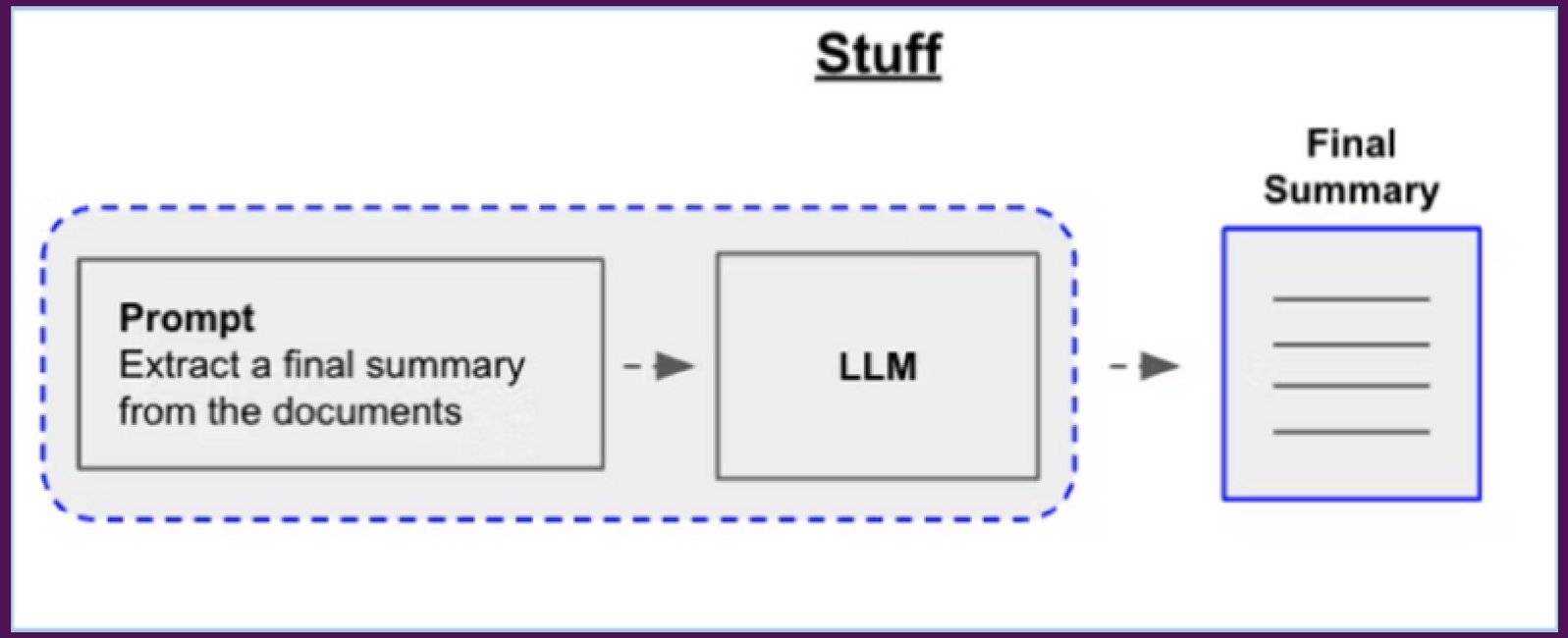
Experiment Result

- Trained for 260 epochs on a 32GB GPU
- Final loss - 1.274, Our loss didn’t converge completely.



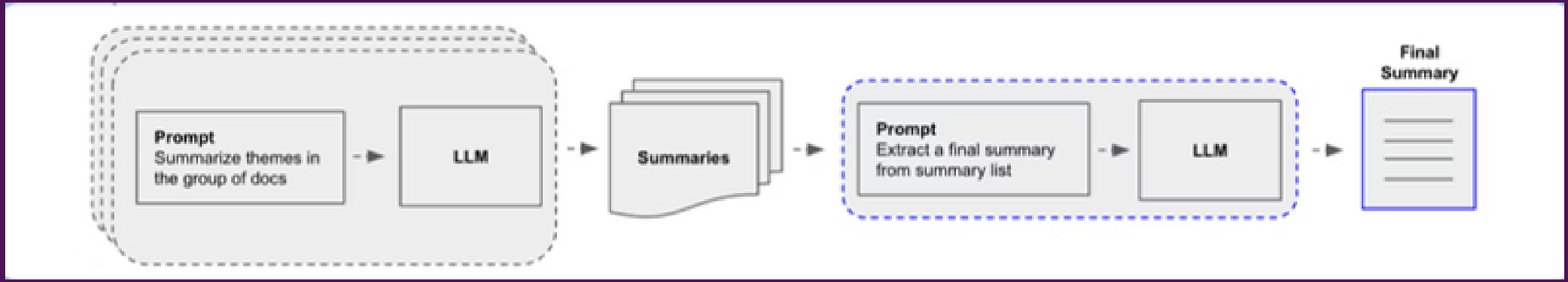
Summary Generation

- **Stuff method**
  - For small to medium-sized readmes directly input the readme text in the prompt so that the model has access to the entire context



Summary Generation

- **Map Reduce method**
  - For larger readmes used, map-reduce method where we divide the readme into smaller chunks and generate summaries for each chunk and finally reduce all of them into a single final summary
  - The results with this method are half baked and sub optimal.



Models

GPT - 4

Dataset Collection

falcon 7b

Summarization

Evaluation Metric

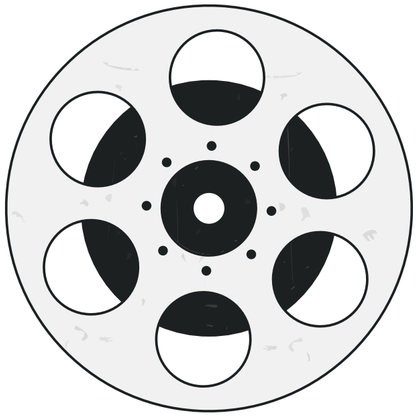
Peer Eval

BLEU Score: 0.417

- We have relied majorly on human evaluation for our custom summarization use case
- Pre defined metrics such as the BLEU Score is not useful as the summary generated by the model varies each time and the score is not able to encapsulate the semantically similar summaries generated.
- This effects the scoring on the final video composed

Video Pipeline

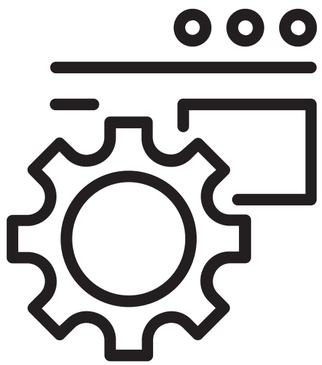
MoviePy library to implement the timeline



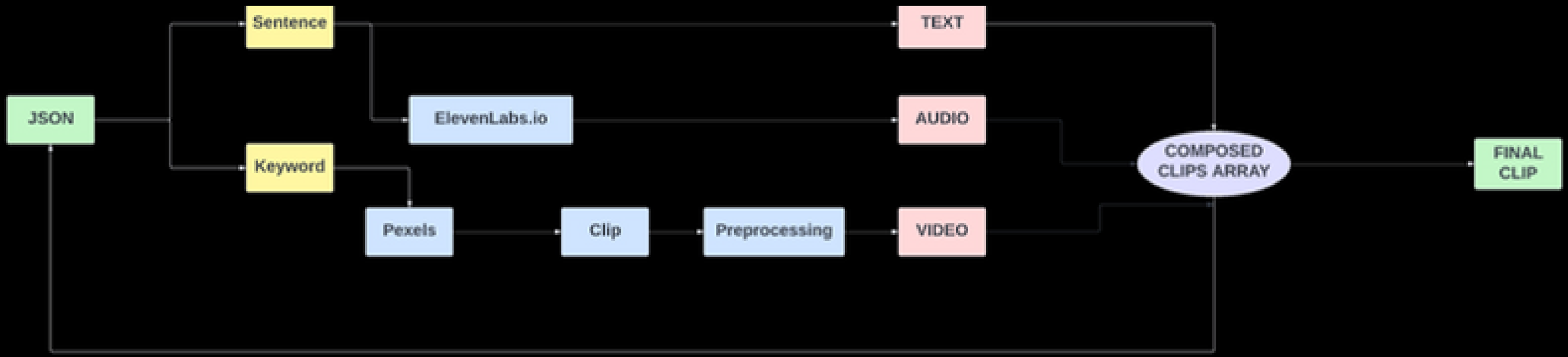
Post Processing Output

Processing the output generated from the model for the video pipeline

```
{ ‘Clip’:
            { ‘Sentence’ : ‘…’
              ‘Keyword’ : ‘…’
            }
}
```



- The keyword is generated by the model
- Used to search for a relevant video using Pexels API
- Audio is generated by ElevenLabs.io
- Combined clip is composed by the pipeline


JSON
Sentence
Keyword
ElevenLabs.io
Pexels
Clip
Preprocessing
TEXT
AUDIO
VIDEO
COMPOSED CLIPS ARRAY
FINAL CLIP

Syllabus: Automated Software Engineering

CSC 591-021 (4579) SP Topic CSC (Lecture)
CSC 791-021 (5548) Advncd Topics CSC (Lecture)

We 4:30PM - 7:15PM
01231 Engineering Building 2

Overview

This subject is a peek under the hood of under the hood of data mining, optimization, theorem proving and all the other tricks of automated software engineering.

This is a project-based class where students will use any scripting language they like to build and extend their own AI tools for software engineering. 500 level students will work in groups.

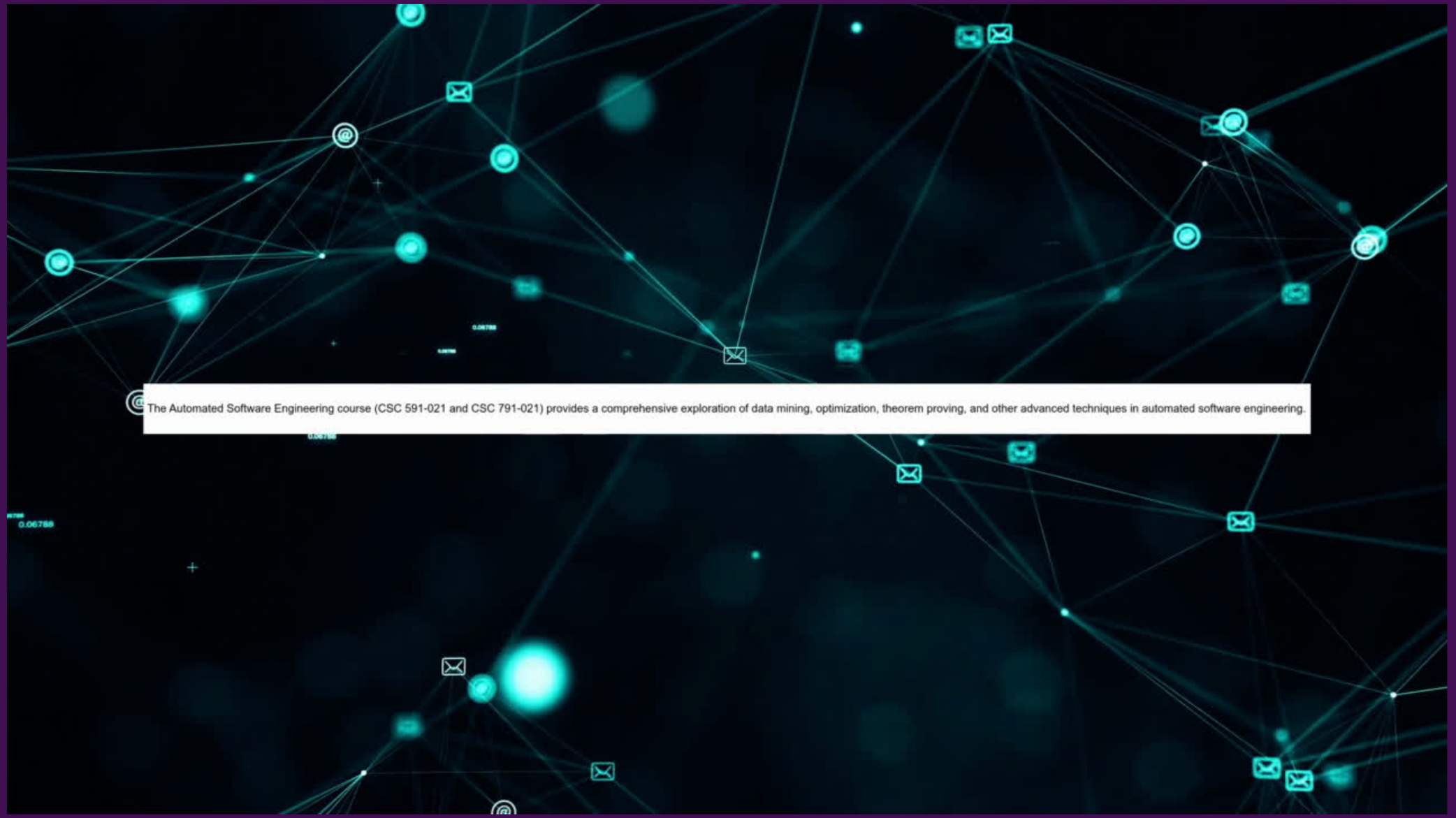
Topics covered: theorem provers, data miners, multi-objective optimizers, semi-supervised learning, fairness/accountability/trust.

Grading: 40% exams; 21% small homeworks, 39% for large end-of-term project.

Textbook: none

Topics

- scripting
- clustering
- optimization
- data mining
- theorem proving


The Automated Software Engineering course (CSC 591-021 and CSC 791-021) provides a comprehensive exploration of data mining, optimization, theorem proving, and other advanced techniques in automated software engineering.

Evaluation

- The tricky part of this project is evaluation
- Traditional methods don’t necessarily encapsulate the subjective nature of keywords generated and the final video itself

Hence a different mode of evaluation

Evaluation

A peer review based evaluation metric

Google Form

PROVIDE

- Original ReadMe
- Summarized JSON
- Rank the LLM generated keyword
- Rank the final video

} USER WILL SCORE

Findings

- Keyword generation from each sentence is not aptly relevant
- The clips associated with a specific keyword relies heavily on the Image API's database.
- Efficient refinement of a pre-trained open-source Falcon base on a high-quality custom dataset proves effective, even when working with a limited number of samples.

Lessons Learnt

- The quality of the inference summary relies significantly on the parameters employed in fine-tuning and inferencing the model. Selecting the appropriate parameter set is crucial for producing high-quality output.

- Human Evaluation stands as one of the most effective evaluation methods for the generated summaries for our domain specific usecase.

Conclusion

- Our approach, employing a Large Language Model for text summarization and keyword generation, is suitable for application-based, domain specific pipelines, including video generation.

- Our pipeline generates commendable video which serves as an effective interface for learning in a given course syllabus's specific README.

Future Work

- Streamlining keyword fine-tuning with image search to construct a high-quality, domain-specific clips dataset.
- Getting human feedback for the 3 videos we have(Baseline, GPT summary and falcon summary)
- Concluding the end-to-end pipeline(video and summarization pipelines) for web application hosting to transform it into a functional tool.

References

[1] Prana, G.A.A., Treude, C., Thung, F. et al. Categorizing the Content of GitHub README Files. Empir Software Eng 24, 1296–1327 (2019). https://doi.org/10.1007/s10664-018-9660-3

[2] Lu, G., Larcher, S. B., & Tran, T. (2023). Hybrid Long Document Summarization using C2F-FAR and ChatGPT: A Practical Study. ArXiv. /abs/2306.01169

[3] R. Deepa and K. N. Lalwani, "Image Classification and Text Extraction using Machine Learning," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 680-684, doi: 10.1109/ICECA.2019.8821936.