Assessment Assistive System

A CAPSTONE PROJECT REPORT

Submitted in partial fulfillment of the
requirement for the award of the
Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE ENGINEERING

by

Sai Krishna (17BECD7094)

Under the Guidance of

DR. Prof BKSP Kumar Raju

SCHOOL OF Computer Science Engineering
VIT-AP UNIVERSITY
AMARAVATI- 522237

March 2021

CERTIFICATE

This is to certify that the Capstone Project work titled "**Assessment Assistive System**" that is being submitted by Sai Krishna**(17BCD7094)**is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. Prof BKSP Kumar Raju
Guide

The thesis is [✓ satisfactory] / unsatisfactory

Approved by

PROGRAM CHAIR
B. Tech. CSE-DA

Geetha H.
DEAN
School of Computer Science Engineering

TABLE OF CONTENTS

CHAPTER 1

Abstract

The sheer volume of answer-sheets, when coupled with essay(long) answers makes the process of evaluation strenuous which in turn compromises precision and objectivity of the evaluator, Existing research in digital evaluation tries to automate the process entirely for which to work as intended the approach of questioning and the corresponding answers are required not to skew beyond a degree relative to the training data which would restrict the ability of teachers to assess the students thus this system would only work for evaluating MCQ Questions. Assessment Assistive system takes the approach of assisting the teacher by automating the aspect of evaluation which fundamentally makes it grueling for the teachers, which is to find the important sub-sections in the answer-sheets based on the internal-rubric teacher has set for the particular question. An Extractive Question answering Model fine-tuned with the vocabulary of a particular subject when given a set of sub-questions as a rubric would highlight the parts of the essay which the teacher is likely to consider to score it making the job evaluator to evaluate instead of search, system also gives the option of adding “Ideal answers” which is a set of answers perfectly aligning with the rubric of the question, which the evaluator can dynamically add to the system for a given question and doing so would generate a average semantic similarity score between all the ideal answers for the question and the answer paper

being evaluated allowing the evaluator to take into consideration how the current answer compares to the set of ideal answers.

CHAPTER 2

Introduction

As Data/content is being generated at ever-increasing rates trying to grab our attention more efforts are being put into content summarization and aggregation as a consequence of which NLP is making better and better strides in comprehending Text and its applications in domains which are in dire need of automating mundane tasks are becoming a reality, with the abundance of answer-papers and time constraints evaluation is a process which has been identified early on in the Digital Education revolution as a domain in need of automation, Natural Natural Processing combines with Deep Learning in recent times have achieved near-human level performance in Tasks such as Text Classification, sentiment analysis and entity recognition, However challenges involved in the process of evaluation do not align with the standard NLP tasks, which does not make the formulation of the problem obvious. Prior work done in this domain to approach this problem have chosen to automate the task entirely which dis-regards the

expertise a teacher brings to the table and overestimating the capabilities of NLP models because evaluation fundamentally is not just about measuring how close the current answer is to some ideal answer rather it is to check weather the current answer states all the facts and covers all the points expected by evaluators internal rubric with respective to the question based on this definition of evaluation the approach chosen is to highlight the sections of the answer sheet based on teachers rubric, which are likely to be considered by the evaluator to sore the paper and the system also calculates the average semantic similarity between the current answer sheet and “Ideal answers” set which the teacher is allowed to dynamically add and come up with a score indicating how close is the current answer semantically to answer which the evaluator considered were perfect,   hence making the approach assistive/suggestive rather than automotive which still keeps the evaluator in the loop making the process more reliable and its implementation more pragmatic.

.

CHAPTER 3

Literature Survey

Before being in a position to evaluate the viability of prior research or work in this domain, Understanding the The process of evaluation is of importance, like many other tasks it can be decomposed into different steps as following

1) Extracting the Information from the document
2) Evaluating the merit of the answer
3) Scoring/Grading the answer

Previous Attempts or Research work conducted to address the bottlenecks in the process of evaluation took the approach of automating the entire process on the basis that NLP as a field in the recent years has been demonstrating ground-breaking performance such as surpassing human level performance in tasks such as text classification, sentiment analysis, entity recognition however the ability of these models which facilitates such performance only accounts for the first step in the process of evaluation listed above which is “extracting the information from the document" and the methods further used as proxy for step2 and step3 were classification but what the actuals steps entails is beyond the formulation of classification frameworks.

The second Step, “evaluating the merit of the answer” requires human/machine to have a deep understanding of both the domain and also the ability to extrapolate different ways in which the same piece of information could be conveyed, It is a challenging part for the machine to put forth the exact inference of the information conveyed which varies with the type of questions and the answers[1]

The third step which is “scoring/grading answer” is dependent not only, on the” keywords used and context of the phrases conveying meaning” [1] but it also depends on to what extent they have been addressed in the answer and the evaluator also seem to award marks on how how effectively an answer is being written along with to what extent each point is discussed.

Though the NLP models are in a position to automate the first step of the evaluation process in its entirety, subjective nature of the second and third steps of evaluation is what makes automated graded systems ineffective based on the following reasoning the system developed leverages the capabilities of NLP by automating the first step but leaves the second and third step to the expertise of the evaluator and as such embodying an assistive nature of sorts.

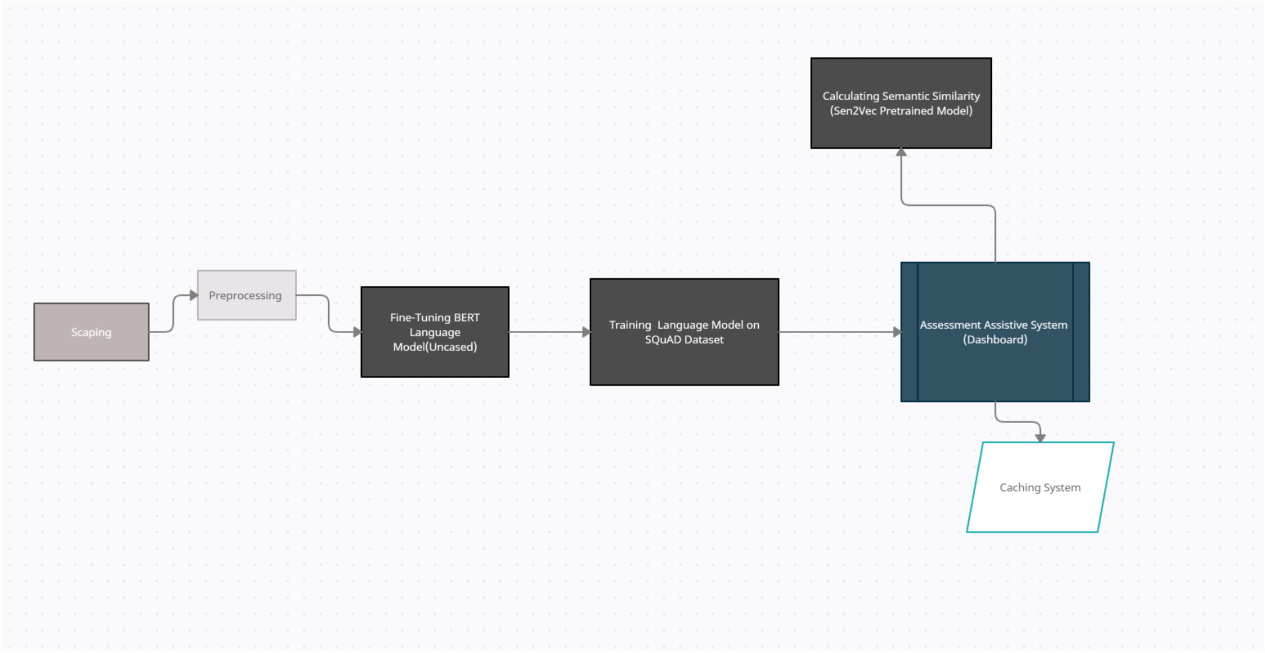
CHAPTER 4

Proposed Work

Current Approach embodied by the Assessment assistive system is inspired from looking at the problem of evaluation of checking if the answer-sheet includes all the points from the evaluators rubric instead of making a semantic comparison between the current-answer and a proxy for an ideal answer As assistive approach has been considered to be more pragmatic the task of the system comes down to automating the bottleneck in the process evaluating which is to search for sections of answer-paper which are most likely related to the sub-questions in the rubric

An extractive Question Answering Model is being used to find the sub-text in the answer sheet which most likely is addressing questions in the rubric, BERT-uncased Language Model which is fine tuned on the vocabulary of the subject the exam is being conducted on to improve the ability of the model make sense of the words being used in the answer-paper and this language model is then trained on the SQuAD Dataset[2]

Along with the main task of highlighting the text, system also consists of feature which allows the evaluator to add “Ideal Answers” which are near perfect answers in terms of consisting all the elements the rubric requires and a sentence to vector pretrained model transforms the current answer and all the ideal answer to a specific question into vectors and computes the semantic similarity scaled between 0 to 100 indicating the how similar the current answer is in comparison to the ideal answer which the evaluator dynamically added


Calculating Semantic Similarity (Sen2Vec Pretrained Model)
Scaping
Preprocessing
Fine-Tuning BERT Language Model(Uncased)
Training Language Model on SQuAD Dataset
Assessment Assistive System (Dashboard)
Caching System

Scraping(4.1)

Data Required for the Language Model to be trained is a text corpus related to the subject on which the test is being conducted, Which means for every new domain or subject the model would be used on it has to be fed text corpus related to that domain, Making the job data-intensive for this approach to be practical, Scrapers have been developed which when fed links(urls) related of a few selected websites such as Geeks for Geeks, Tutorial Points, Java-T-Point would recursively scrape every article present on that subject on the respective websites, Parse the text out of the HTML template and save it into text files

Preprocessing(4.2)

Since the text is being directly scraped from the internet feeding it to the any model in its raw form would not yield appropriate results, Following are the preprocessing steps taken

1. All the hyperlinks and non-alphanumeric content is removed
2. All the numbers are converted into numbers in word form
3. Text is converted into lower case

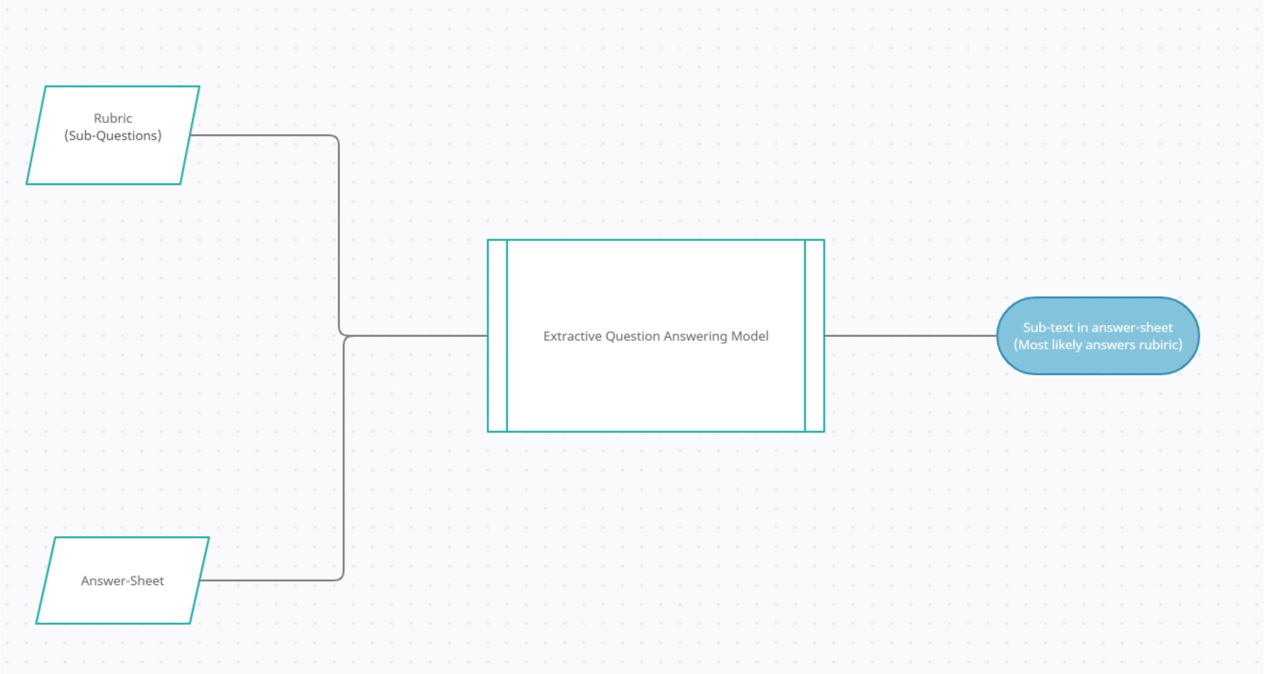
4. Commonly used “STOP-words” are removed
5. Stemming and lemmatization is performed
6. All the empty spaces, line breaks are removed

Fine Tuning BERT Language Model-uncased(4.3)

BERT Uncased Language Model is pretrained on Wikipedia Dataset which consists of generic words but since this language model (after further training) has to perform well on Assessment’s Answer papers related to a particular subject , Which are more likely to consists of technical words of the respective domain using a Model which consists of Vocabulary of general words is not the optimized approach Hence the pre-trained model is further fine tuned on domain specific text corpus which consists of technical words of the respective subject which allows the model to emphasize more on semantic importance of technical words which otherwise would not have been given.

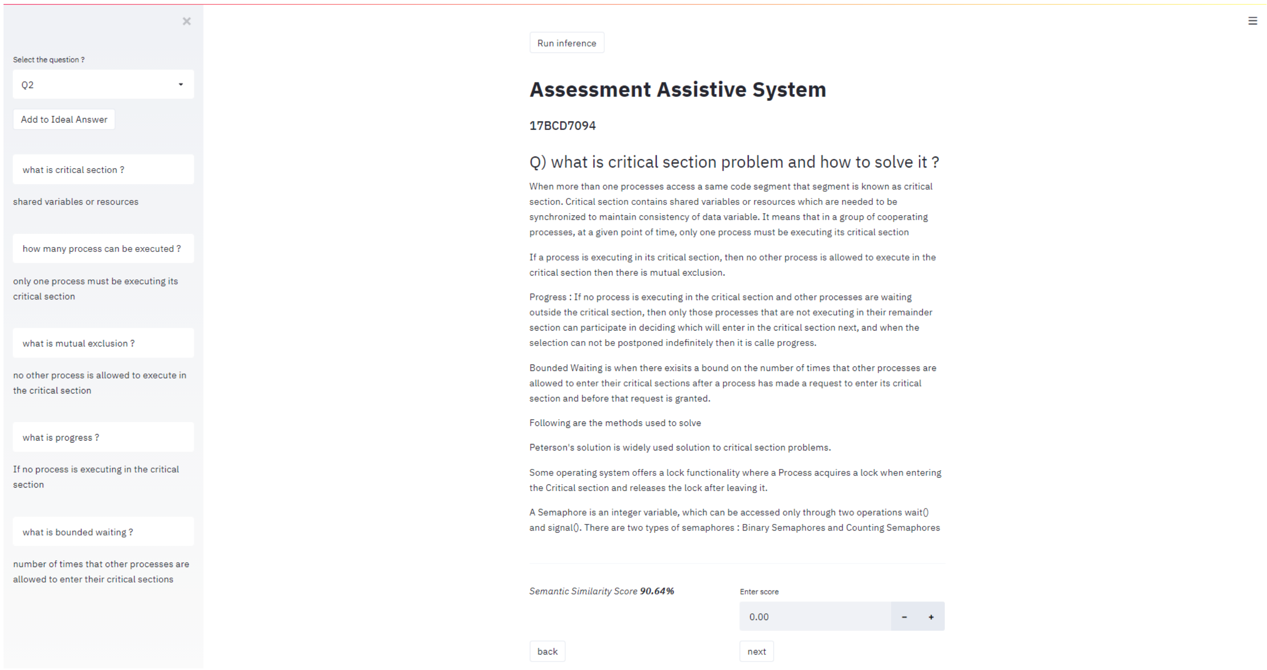
Training the BERT uncased Model on SQuAD dataset for Extractive Question Answering(4.4)

SQuAD (Stanford Question-Answering Dataset) is a Dataset used to train Models for finding the answer to the Question in given Document, The structure of the dataset makes it a Great Candidate for our system as the Rubric provided by the evaluator is also a set of questions.


Rubric
(Sub-Questions)
Extractive Question Answering Model
Sub-text in answer-sheet
(Most likely answers rubiric)
Answer-Sheet

After Training Our Model both on Domain Specific Vocabulary and SquAD dataset, Model is capable of Identifying Sub-Text in the answer sheet of the student Given a Rubric(set of sub-questions), Reducing the amount of time teacher has to spend searching for that part manually.

Dashboard-Assessment Assistive System(4.5)



what is critical section ?

shared variables or resources

how many process can be executed ?

only one process must be executing its critical section

what is mutual exclusion ?

no other process is allowed to execute in the critical section

what is progress ?

If no process is executing in the critical section

what is bounded waiting ?

number of times that other processes are allowed to enter their critical sections

Assessment Assistive System

17BCD7094

Q) what is critical section problem and how to solve it ?

When more than one processes access a same code segment that segment is known as critical section. Critical section contains shared variables or resources which are needed to be synchronized to maintain consistency of data variable. It means that in a group of cooperating processes, at a given point of time, only one process must be executing its critical section

If a process is executing in its critical section, then no other process is allowed to execute in the critical section then there is mutual exclusion.

Progress : If no process is executing in the critical section and other processes are waiting outside the critical section, then only those processes that are not executing in their remainder section can participate in deciding which will enter in the critical section next, and when the selection can not be postponed indefinitely then it is calle progress.

Bounded Waiting is when there exisits a bound on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.

Following are the methods used to solve

Peterson's solution is widely used solution to critical section problems.

Some operating system offers a lock functionality where a Process acquires a lock when entering the Critical section and releases the lock after leaving it.

A Semaphore is an integer variable, which can be accessed only through two operations wait() and signal(). There are two types of semaphores : Binary Semaphores and Counting Semaphores

Semantic Similarity Score ***90.64%***

A Dashboard is built which acts as an interface for the evaluator to navigate on Student’s Answer papers, Extractive Question Answering Model which is trained is embedded into the dashboard for inference, Evaluator is required to setup up the rubric for each question in the form of set of sub-questions which then enables the evaluator to utilize the inference provided by the model of extracting the parts of Answer-Sheet which are most likely to address the question in the rubric which then Teacher can use to evaluate the answer

Semantic Similarity(4.6)

System allows Evaluators not only to add more than one answers which they deem to perfectly fit the rubric but it also allows teachers to add answers dynamically (ie: If evaluator assumes the paper he/she is evaluating to be perfect then there is an option in the dashboard which allows teachers to add that paper to the ideal set of answers), Having a set of ideal answers to a specific questions enables the evaluator to view how semantically similar the current answer is in-comparison to all the ideal answers, This allows evaluator to mechanically judge to what degree does the current answer align with the set of ideal answers(chosen by the evaluator) which if its the only metric evaluator would be suing to score would have been a bad metric but when married with the objective evaluation based on how well the current paper answers questions in the rubric makes it an enhancement,

To measure the semantic similarity a pre trained Sen2Vec(Sentence to vector) Model is used to convert the current answer paper and the ideal answer papers into vectors and then average cosine similarity is computed between the current answer paper and ideal answers, which gives out the score scaled between 0 to 1 and is presented on dashboard between 0-100(as a percentage similarity)

Cache System(4.7)

As the Question Answering Model is heavy(Has many layers with huge vocabulary) Inference time of the model(ie Taking too much time identifying sub-text in the answer-sheets), The vert reason for introducing this system is to make the process of evaluation fast and smooth but if the time model takes to generate its output is higher than the teacher would take to manually go through the the papers then the purpose of the system is defied, to avoid which a cache system is built into the dashboard which gives the options to run inference on all the papers of all the students before the evaluator starts to correct papers so that when the teacher starts the process of evaluation the output of the mode can simply be loaded from the cache instead of running real-time, Cache also allows evaluators to save any changes made to the answer papers, So if any further changes have been made to the answer papers such as adding new papers or editing already existing one's the cache system will detect that changes have been made and run the model on the new version of the paper and save the results for the new version

CHAPTER 5

Conclusion

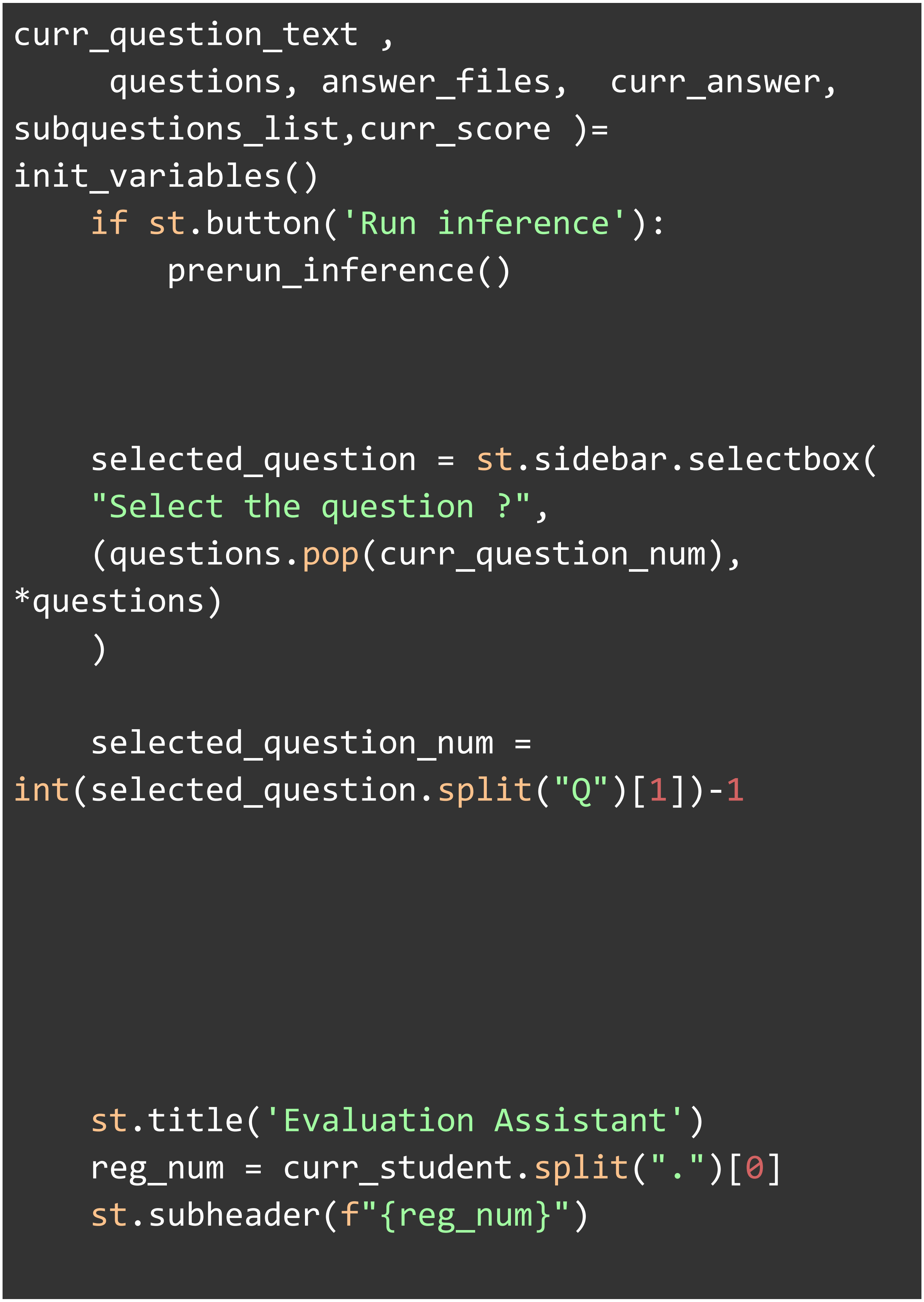
The approach chosen and the methods used to implement assessment assistive system are based on cognitive modeling which is to develop solution based on how the underlying process is done manually and then identifying the aspects or steps of the process which can be deemed mundane and have potential to be automated, The current system identifies that finding the sub-text in the answer paper is the mundane aspect of the evaluation process and the current state of the NLP Models to comprehend text with expectational ability facilitates this step to be automated,

Since the goal is to assist the evaluator in scoring the paper with ease and speed as a consequence feature such as semantic similarity has been included which provides a score between 0-100 as to how similar the current answer sheet is to the answer-sheets the evaluator or the teacher deemed to be ideal based on the rubric

However the way in which current rubric is designed which is a set of questions restricts the evaluator to break-down the aspect of the answer he/she wants to find into sub-questions Perhaps if and when NLP models are capable of identifying Sub-text given an abstract description would allow the evaluator to come up with effective ways of designing the rubric

Appendix

```
import streamlit as st
import numpy as np
import pandas as pd
import os


from constants import QUESTIONS_Dir,
ANSWER_Dir, QUESTIONS_File, SUB_Question_File,
SCORE_Dir, IDEAL_Dir , SESSION_File


from helper import write_session
,init_variables,update_score,
estimate_similarity, get_key, highlight_text,
add_ideal_answer, add_highlights

from preload_cache import prerun_inference

if __name__=='__main__':



    (curr_question_num,
curr_student_num,curr_question,curr_student,
```



```
curr_question_text ,
     questions, answer_files,  curr_answer,
subquestions_list,curr_score )=
init_variables()
    if st.button('Run inference'):
        prerun_inference()



    selected_question = st.sidebar.selectbox(
    "Select the question ?",
    (questions.pop(curr_question_num),
*questions)
    )

    selected_question_num =
int(selected_question.split("Q")[1])-1







    st.title('Evaluation Assistant')
    reg_num = curr_student.split(".")[0]
    st.subheader(f"{reg_num}")
    st.header(f"Q) {curr_question_text}")


    keys = list()
    highlight_positions = list()

    for subquestion in subquestions_list:
        resp = get_key(curr_question_text ,
curr_answer, subquestion)
        keys.append(resp["answer"])

highlight_positions.append((resp['start'],
resp['end']))


    markdown_writer =
add_highlights(curr_answer,
sorted(highlight_positions, key=lambda x:
x[0]) )
    st.markdown(markdown_writer)

    add_to_ideal = st.sidebar.button("Add to
Ideal Answer")

    if(add_to_ideal):
        add_ideal_answer(QUESTIONS_Dir,
curr_question, IDEAL_Dir, curr_answer)
```

```
    for question, key in
zip(subquestions_list, keys):
        st.sidebar.text_input("",question)
        st.sidebar.write(key)




    st.markdown("___")
    col21, col22 = st.beta_columns(2)

    raw_semantic_score =
estimate_similarity(QUESTIONS_Dir,
curr_question, IDEAL_Dir, curr_answer)
    if(raw_semantic_score):
        semantic_sim_socre =
round(raw_semantic_score*100, 2)
        col21.markdown(f"*Semantic Similarity
Score **{semantic_sim_socre}%** *")

    else:
        col21.markdown("No ideal answers added
!")


    if(curr_score):
        updated_score =
col22.number_input('Enter score', value =
float(curr_score))

    else:
        updated_score =
col22.number_input('Enter score')

    if(updated_score and
updated_score!=curr_score):
        update_score(QUESTIONS_Dir,
curr_question, SCORE_Dir, curr_student ,
updated_score)




    col1, col2 = st.beta_columns(2)
    back_bool = col1.button("back")
    next_bool = col2.button("next")

    if(back_bool):
        if(curr_student_num==0):
            st.text("This is the first file")
        else:
            curr_student_num-=1
            write_session(SESSION_File,
curr_student_num, selected_question_num)

    if(next_bool):

if(curr_student_num==len(answer_files)-1):
            st.text("This is the last file")
        else:
            curr_student_num+=1
            write_session(SESSION_File,
curr_student_num, selected_question_num)


if(selected_question_num!=curr_question_num):
        write_session(SESSION_File,
curr_student_num, selected_question_num)
```

References(5.1)

[1]Nandini, V., Uma Maheswari, P. Automatic assessment of descriptive answers in online examination system using semantic relational features. J Supercomput 76, 4430–4448

[2] rajpurkar2016squad, Pranav Rajpurkar and Jian Zhang and Konstantin Lopyrev and Percy Liang SQuAD: 100,000+ Questions for Machine Comprehension of Text

BIO Data

Name: Sai Krishna Manthena
Mobile Number: 8099210858
Email: saikrishna.manthena@vitap.ac.in
Perminent Email: saimanthena13579@gmail.com