Worcester Polytechnic Institute

Machine Learning Implementation in Education Technology

A Major Qualifying Project Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science in Computer Science

By

John Bulman

Rui Huang

Connor Paisner

Zonglin Peng

Alp Piskin

Date: March 6th, 2019

Advisor:

[illegible]

Prof. Neil T. Heffernan

Abstract **2**

Acknowledgments **2**

Introduction **3**

1.1 QUICK-Comments 3

1.2 ASSISTments 3

1.3 Platform 4

Methodology and Results **5**

2.1 Software Design 5

2.1.1 User Interface 5

2.1.2 User Testing 6

2.1.3 HTTP Request 9

2.1.4 HTTP Response 10

2.2 QUICK-Comments Server and APIs 11

2.2.1 Baseline Predictive Model 11

2.2.2 Python Server 12

2.3 Data Processing 13

2.3.1 Data Pre-processing and Analysis 13

2.3.2 Data Logging 14

2.4 Preliminary Research 15

2.4.1 Word Embedding Method 15

2.4.2 Sentence Embedding Method 16

2.4.3 Comments Clustering 16

2.4.4 Generic and Specific Comments Analysis 17

2.5 Logging 20

2.5.1 Python API Logging 20

Conclusion **21**

References **22**

Appendices **23**

Appendix A: Screenshots of Past Iterations 23

Appendix B: Table of Authors 24

Appendix C: API Server Code (Python Flask) 25

Appendix D: Comment Clusters 27

Abstract

The goal of this project is to use machine learning to provide constructive feedback for student open responses. We are working with the ASSISTments team and their platform to help teachers give more helpful and timely feedback to their students. Using a dataset of open responses and corresponding grades, we were able to use machine learning to develop a model that can give what the most likely grade for an essay might be. Our model gives three suggestions for a given student response, and the teacher can pick the most appropriate one. We will further develop this using user testing with teachers that want to try out new features in ASSISTments.

Acknowledgments

We would like to thank our advisors Neil Heffernan and Anthony Botelho for guiding us through this process. We would also like to thank Christopher J Donnelly and the other faculty working with ASSISTments for their support.

Introduction

1.1 QUICK-Comments

QUICK-Comments (QC) is a webpage in ASSISTments where it runs an AI-powered program that will suggest personalized feedback comments to teachers for student essays. Current math textbooks ask students to explain their thinking 3 to 4 times on each night's homework, but teachers often do not have time to give detailed feedback on these problems. To improve the grading speed and to enrich details of feedback, QC aims at providing the teachers a number of comments based on the students' answers. Then, the teacher can edit and send the feedback to their students. QC page integrates directly with the ASSISTments platform and relies on machine learning and natural language processing (NLP) techniques to suggest responses for teachers to send to their students. This project works in a manner similar to Google Smart Reply.

1.2 ASSISTments

ASSISTments is an online educational platform for students to learn and practice different topics with problem sets. It provides a framework to develop modular sequences of problems and deploy them at a large scale, and is used by middle and high schools across the nation in varied environments. Students can use the platform in a classroom aspect and teachers can use it to view their progress or utilize the Problem Set Builder to design problem sets for deployment. The modularity this platform provides is a highly desired feature that many of the teachers use. "I love the rich data that I have to start my class. I write most of my own curriculum through ASSISTments which allows me to personalize the assignments to my classes and school." (Hinkley, William). William Hinkley is one of the many teachers that uses the

ASSISTments platform for classroom teaching. Him and many other teachers value the website as a tool to easily help them learn more about how their students are performing and what the reasons behind that are.

1.3 Platform

The ASSISTments platform is used by teachers for a variety of reasons. It is used as a way to give automatic feedback to students regarding their level of understanding of different problems, as well as to track students progress. ASSISTments allows the teachers and students to understand how well certain skills are known. Teachers also use it to find common misconceptions in the classroom, allowing them to refine their teaching style. To use the system, the teacher first assigns a student, or group of students, a specific problem set ID which corresponds to a sequence of questions. The correctness of the questions is based on the number of attempts and hints the student uses to get the right answer. In general, the question is marked as correct if the student answered it correctly on the first try. If the student does not get the question right on the first try, they may still get partial credit if they answer correctly depending on the grading rules the teacher has in place.

Methodology and Results

2.1 Software Design

ASSISTments is currently working on a project called The New Generation (TNG) of the website and our goal was to create a separate QUICK-Comments page under the TNG project as an extension to the essay grading function. The main feature that was implemented was the feedback system. The feedback system uses a predictive model to generate three comments to an open-ended answer text. The comments are displayed as selectable feedback to the teacher on their grading page. If the teacher decides to use the suggestion, they can simply click on it. It is possible for them to write their own feedback or modify the existing suggestion that was selected. This mainly aims to accelerate the grading process for teachers, to gather data regarding the grading habits to make improvements to the page, and to analyze the model's findings to gather insight about the process of essay grading.

2.1.1 User Interface

One major aspect of this project was the building of the new QUICK-Comments page. We started by building off of the existing essay grading page. The grading page is a page that teachers currently use to grade open response questions in ASSISTments. The page consists of a table where teachers can input a grade and a comment for each student's answer to the open response question. We expanded on this original essay grading page by adding a new column to the table to hold our three suggested messages for each student. We also made it so clicking on a suggested message would put that message in the teacher feedback box, where the teacher would then have a chance to edit it.

In addition to the suggested message column, we also initially added diagnosis and action columns. The purpose of the diagnosis column was to classify the student's response into broad categories. For example, the diagnosis column might label the student as confused if it seemed like they did not understand the question. The purpose of the action column was to give the teacher the option to assign the student another problem set if it looked like they did not understand the question. We added these columns because they appeared in the original QUICK-Comments grant, but we later removed them as we realized we would not have time to fully implement these features. This could be a potential further improvement for this project.

2.1.2 User Testing

Over the course of our project, we met with a few teachers to get feedback on the user interface. In our first meeting with these teachers, we focused on how they would want to submit their comments to students. We came up with four different designs for a submission column (see Figure 1) that we presented to the teachers. Our first design was to just have a submit button on each row so that teachers could submit their comments to each student as they finished writing them. Our second design was identical to the first with the addition of a "submit all" button that would give teachers the option to submit all of their comments at once rather than one at a time. Our third design replaces the submit buttons on each row with checkboxes and has "select all" and "submit selected" buttons. This design would allow teachers to submit many comments at once without necessarily having to submit them all. Our fourth and final design was a combination of first and third designs. This design included a submit button on each row, if the teacher wanted to submit their comments as they finished writing them, and checkboxes on each row, if the teacher wanted to submit many comments at once.

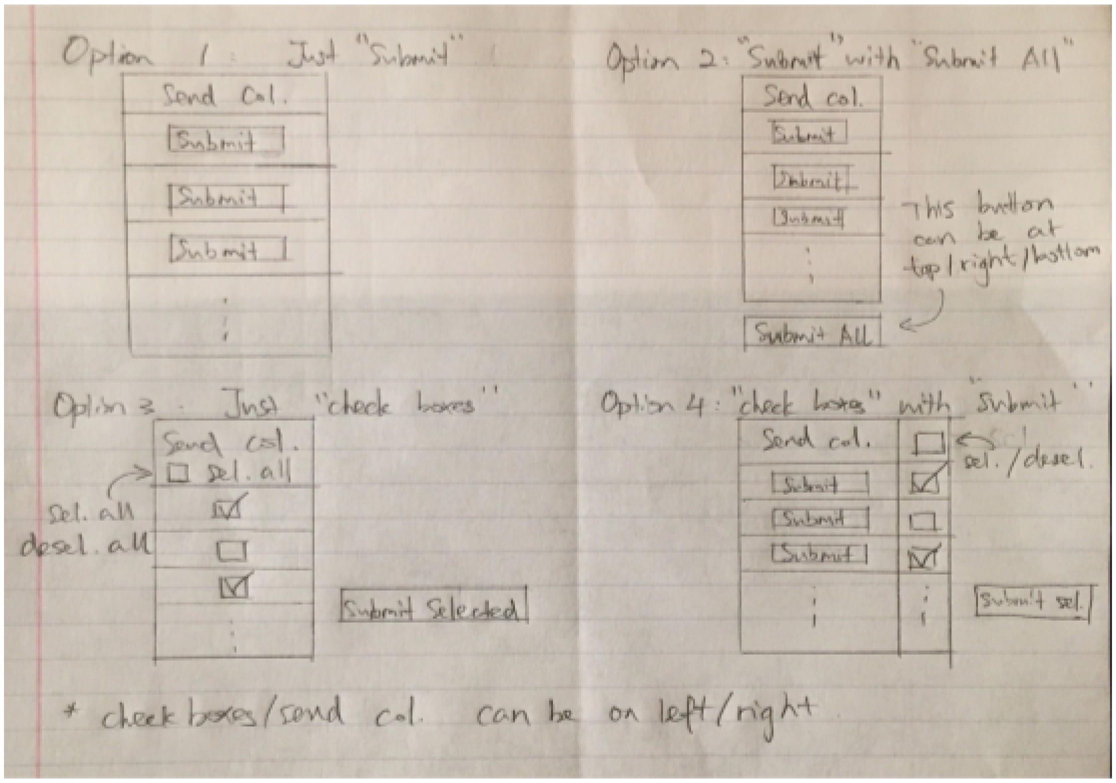
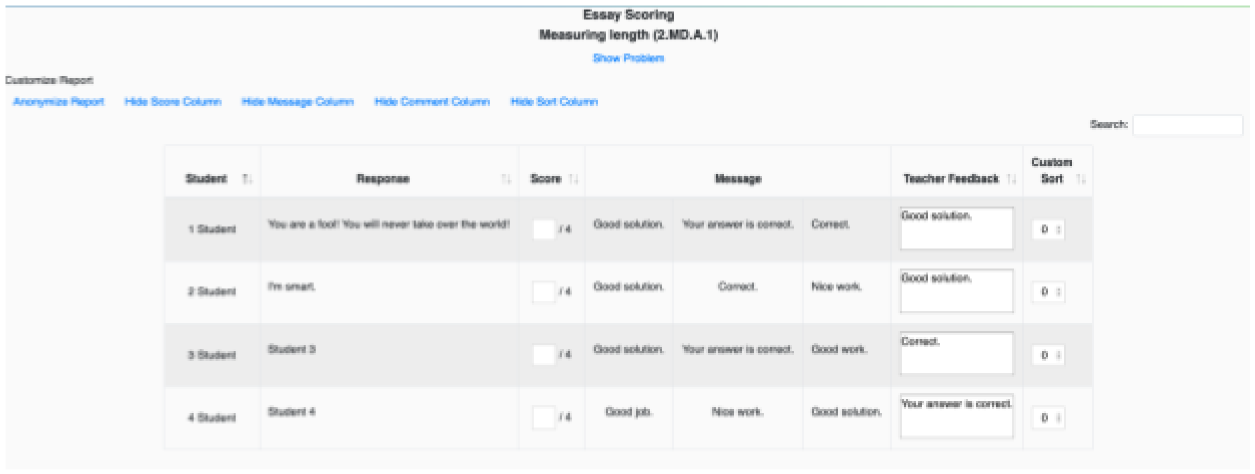


Figure 1: Design Options

When we presented them with these four designs, the teachers ultimately chose a fifth option. They decided that it would be simplest to keep the existing method for submitting comments on the essay grading page. This meant getting rid of the submission column altogether and having comments automatically be submitted to the corresponding student whenever a teacher feedback box is unfocused after anything has been typed in it.

In our final meeting with the teachers, we asked them for general feedback on the design of the QUICK-comments page. After looking at our design (see Figure 2), the teachers came up with three major suggestions to improve the page. The first was to make the suggested messages look more like buttons so that clicking on them would be more intuitive. Their second suggestion was to add placeholder text to the teacher feedback boxes to inform teachers that they can either select a suggested message or type their own. The teachers' third recommendation was to add tooltips to some of the columns to explain how they work.


Essay Scoring
Measuring length (2.MD.A.1)
Show Problem

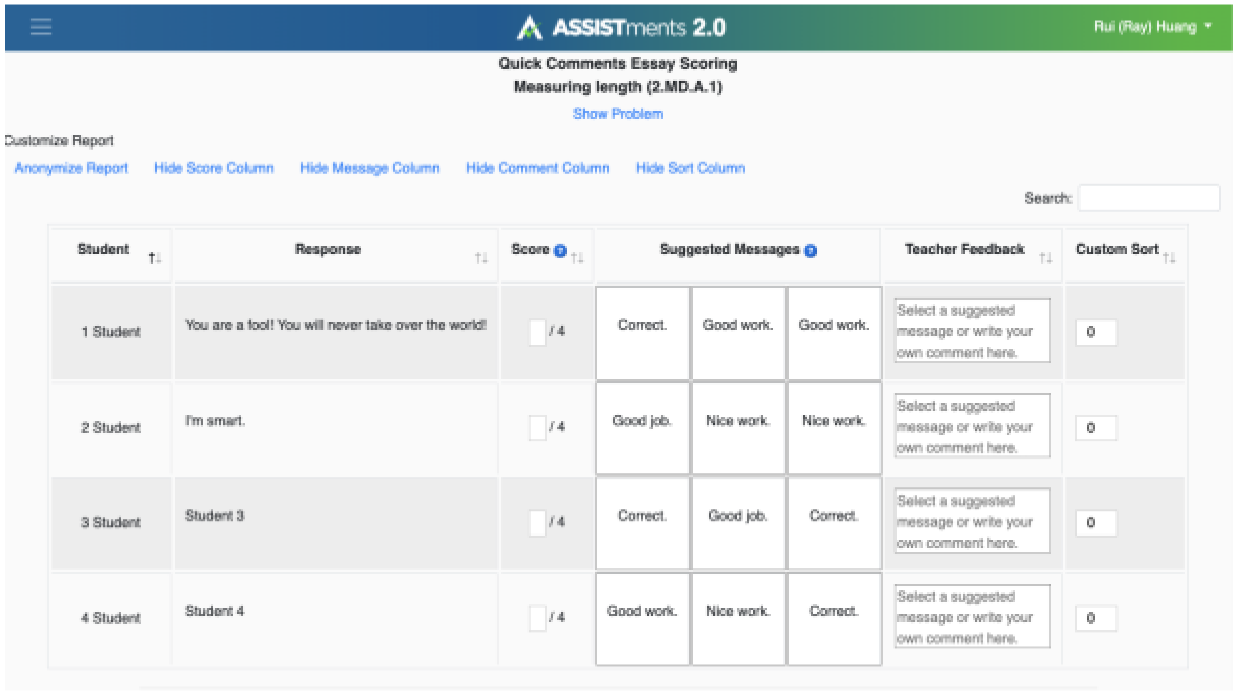
Customize Report
Anonymize Report   Hide Score Column   Hide Message Column   Hide Comment Column   Hide Sort Column

Search:

Student	Response	Score	Message			Teacher Feedback	Custom Sort
1 Student	You are a fool! You will never take over the world!	/ 4	Good solution.	Your answer is correct.	Correct.	Good solution.	0
2 Student	I'm smart.	/ 4	Good solution.	Correct.	Nice work.	Good solution.	0
3 Student	Student 3	/ 4	Good solution.	Your answer is correct.	Good work.	Correct.	0
4 Student	Student 4	/ 4	Good job.	Nice work.	Good solution.	Your answer is correct.	0

Figure 2: Initial Prototype

After the meeting, we updated the page to its current design based on the teacher feedback (see Figure 3). The suggested messages were made to look more like buttons by giving each of them a small gray border that turned into a larger black border when the mouse hovers over them. Placeholder text was added to each teacher feedback box that informs teachers that they can select a suggested message or type their own. Lastly, tooltips were added to the score and suggested messages columns that inform teachers on how to use them when hovered over.


ASSISTments 2.0

Rui (Ray) Huang

Quick Comments Essay Scoring

Measuring length (2.MD.A.1)

Show Problem

Customize Report

Anonymize Report Hide Score Column Hide Message Column Hide Comment Column Hide Sort Column

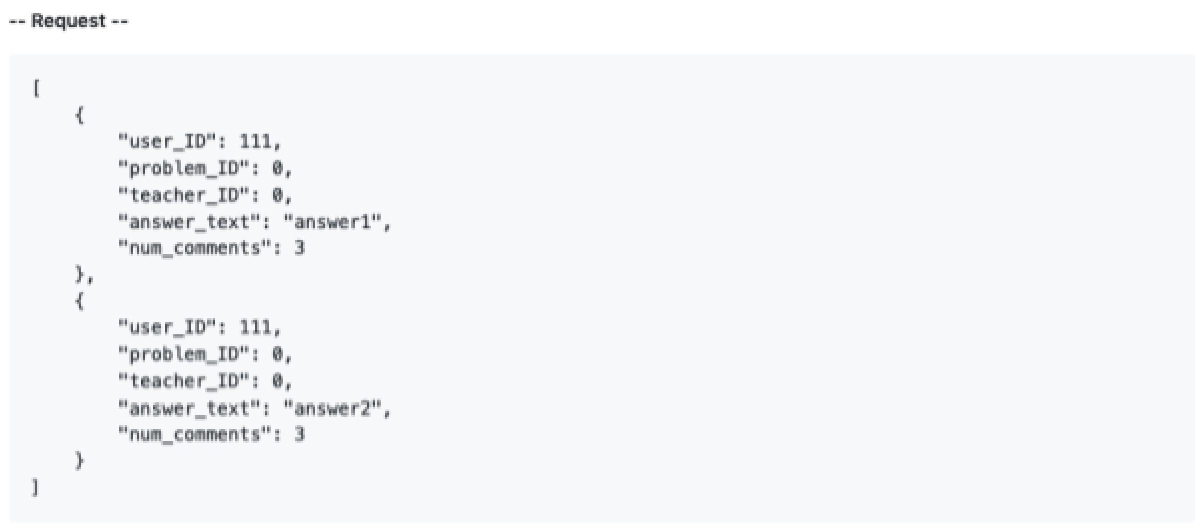
Search:

Student	Response	Score	Suggested Messages			Teacher Feedback	Custom Sort
1 Student	You are a fool! You will never take over the world!	/ 4	Correct.	Good work.	Good work.	Select a suggested message or write your own comment here.	0
2 Student	I'm smart.	/ 4	Good job.	Nice work.	Nice work.	Select a suggested message or write your own comment here.	0
3 Student	Student 3	/ 4	Correct.	Good job.	Correct.	Select a suggested message or write your own comment here.	0
4 Student	Student 4	/ 4	Good work.	Nice work.	Correct.	Select a suggested message or write your own comment here.	0

Figure 3: Final Design

2.1.3 HTTP Request

The project website interacts with the predictive model using a server API. When the website loads, it sends an HTTP request to the API including the student answers and the API responds with the generated comments. For the API, a Flask server was used to hold the model and respond to the website. The request consists of a small JSON package (see Figure 4) including the text, the student ID and the problem ID to stamp the answer, the teacher ID for logging purposes, and the number of comments requested in order to have multiple comment generation options in the future.



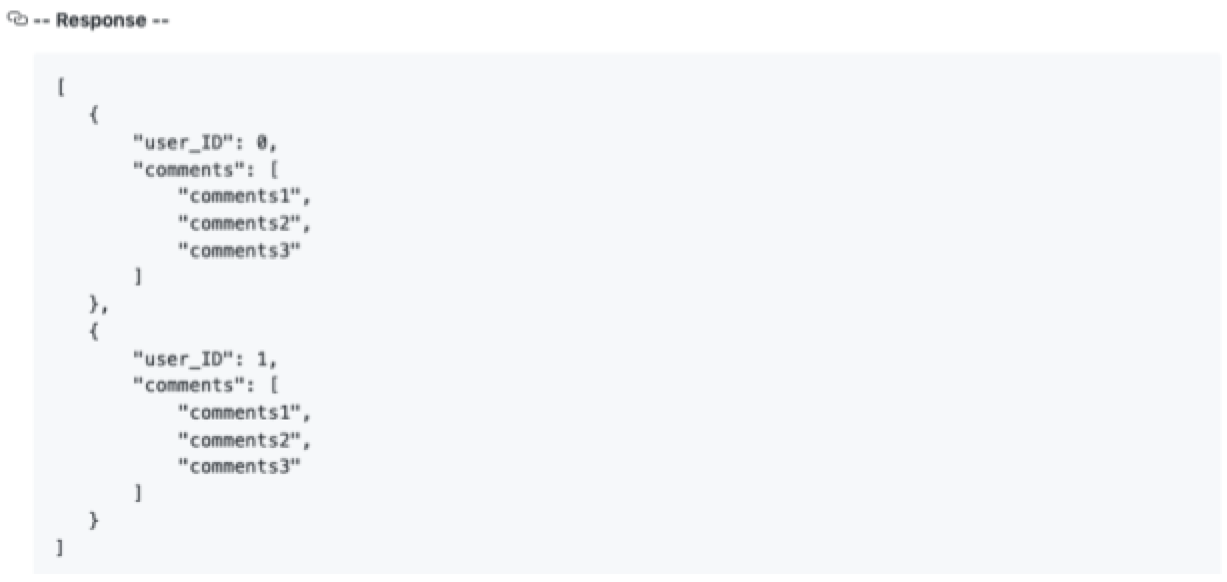
-- Request --

```
[
    {
        "user_ID": 111,
        "problem_ID": 0,
        "teacher_ID": 0,
        "answer_text": "answer1",
        "num_comments": 3
    },
    {
        "user_ID": 111,
        "problem_ID": 0,
        "teacher_ID": 0,
        "answer_text": "answer2",
        "num_comments": 3
    }
]
```

Figure 4: JSON Request (Example array of two answers)

2.1.4 HTTP Response

When the request is sent to the API, the answer text is parsed and sent to the model to predict the comments. Then the generated comments need to be coupled with the associated student user's ID to be sent back to the page.



-- Response --

```
[
   {
       "user_ID": 0,
       "comments": [
           "comments1",
           "comments2",
           "comments3"
       ]
   },
   {
       "user_ID": 1,
       "comments": [
           "comments1",
           "comments2",
           "comments3"
       ]
   }
]
```

Figure 5: JSON Response (Example array of two answers)

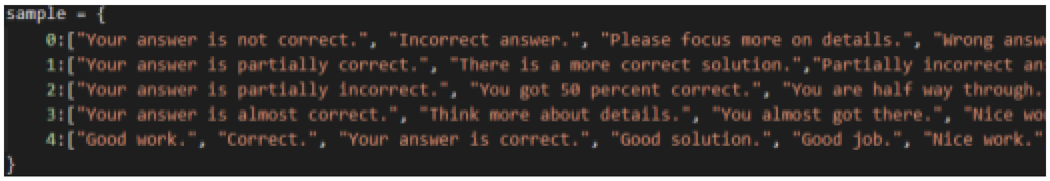
2.2 QUICK-Comments Server and APIs

A backend server was created to support text information transferred from the HTTP request and to generate comments for each of the student answers. For easy development and testing, this server is deployed separately from the TNG project. As a result, our software designs in the TNG communicates with the server by HTTP requests and responses in JSON format. Python and its Flask technology were selected to create the APIs and the infrastructure for or the predictive model. The APIs serves the purpose of taking the HTTP requests and sending an HTTP response via Flask. The infrastructure is to send inputs to the trained predictive model and receive the predictive results from the model.

2.2.1 Baseline Predictive Model

When the project was being developed, there was not a constructed model for our project to take inputs including student answers and predict comments. Therefore, a baseline model was implemented in order to generate unique and current comments in some degree. The baseline model was essential because it serves the purpose of not only testing but also exposing a better result in the demos to the users.

By that time, a simple neural network model had been created to predict grades based on student answer texts. The baseline predictive model, also known as the baseline model, is designed to use the grade predictions to select corresponding teacher comments from a library of comments. The library of comments is predefined and categorized by the grades from zero to four, where zero is the minimum score and four is the maximum. When the grade is predicted, a given number of comments are randomly selected from grade category as the predicted comments for the baseline model.



```
sample = {
    0:["Your answer is not correct.", "Incorrect answer.", "Please focus more on details.", "Wrong answ
    1:["Your answer is partially correct.", "There is a more correct solution.","Partially incorrect an
    2:["Your answer is partially incorrect.", "You got 50 percent correct.", "You are half way through.
    3:["Your answer is almost correct.", "Think more about details.", "You almost got there.", "Nice wo
    4:["Good work.", "Correct.", "Your answer is correct.", "Good solution.", "Good job.", "Nice work."
}
```

Figure 6: Library of comments partially captured from the code

The baseline model is able to provide different comments each time. But for further improvements,the library can be enriched by the commonly used teacher comments from the ASSISTments data. Secondly, Teacher IDs and Problem IDs can be taken into consideration so that for different teachers and different problems, different libraries of comments can be used for selections.

2.2.2 Python Server

The python server is a package of python scripts using Flask. To deploy the Python server, the developer needs to run the script on localhost, and then it can take HTTP requests. As discussed in the HTTP Request section, the IDs and corresponding student answer texts are sent to the API server. From there, the information will be handled in two stages before sending back to the TNG as the response. First, for each of the students, the answer text will be parsed into vectors (word and sentence embedding, explained in the Preliminary Research section) as one of the inputs to the baseline model. Next, the baseline model will take the parsed student text and generate the given number of comments to the corresponding problem ID. As per the design of problem ID in TNG, it is unique throughout the assignment. Therefore, the user ID will be the primary key in the TNG to match the suggested comments with the student answers. After the ID and the comments are generated by the Python server, the data is sent to the TNG in JSON format (see Figure 5).

2.3 Data Processing

The ASSISTments platform has been used by over thousands of teachers and it has a large database, storing relative information such as comment history or students' answers. From those existing archive, we can process and analyze the data and decide what should we build for QUICK-Comment before we start implementing it. And after finishing the product, we also need to log relevant information for future evaluation purposes.

2.3.1 Data Pre-processing and Analysis

During our data pre-processing stage, we fetched all the relevant data from the ASSISTments database, which includes millions of answers from open-response questions. We cleaned the data from the raw .csv file and prepared for the future uses:

The teacher's comments gives us an intuition for how should we generate suggested responses for teachers based on the student's answer. The analysis shows that when the answer is correct or near correct, words such as "Good job", or "Well done" are used frequently; whereas when the answer is incorrect, there is not a commonly used comment since teachers will usually provides student feedback on that specific problem.

And the student's answers shows their answer pattern: how frequently they uses math equations, what are the commonly used math symbols. Based on those symbols used, we can extract the math equations from their text and analyze the meaning of those equations, which will make our decision more accurate.

2.3.2 Data Logging

After we finish the main functionalities in QUICK-Comments page, we then need to build a data logging function (logger) that logs the information such as teacher interaction with the page, or the detailed information of the predictive model. After several discussions with the software developing team, we decide to create 5 new tables in the database, which can be described as the following Entity-Relationship diagram:

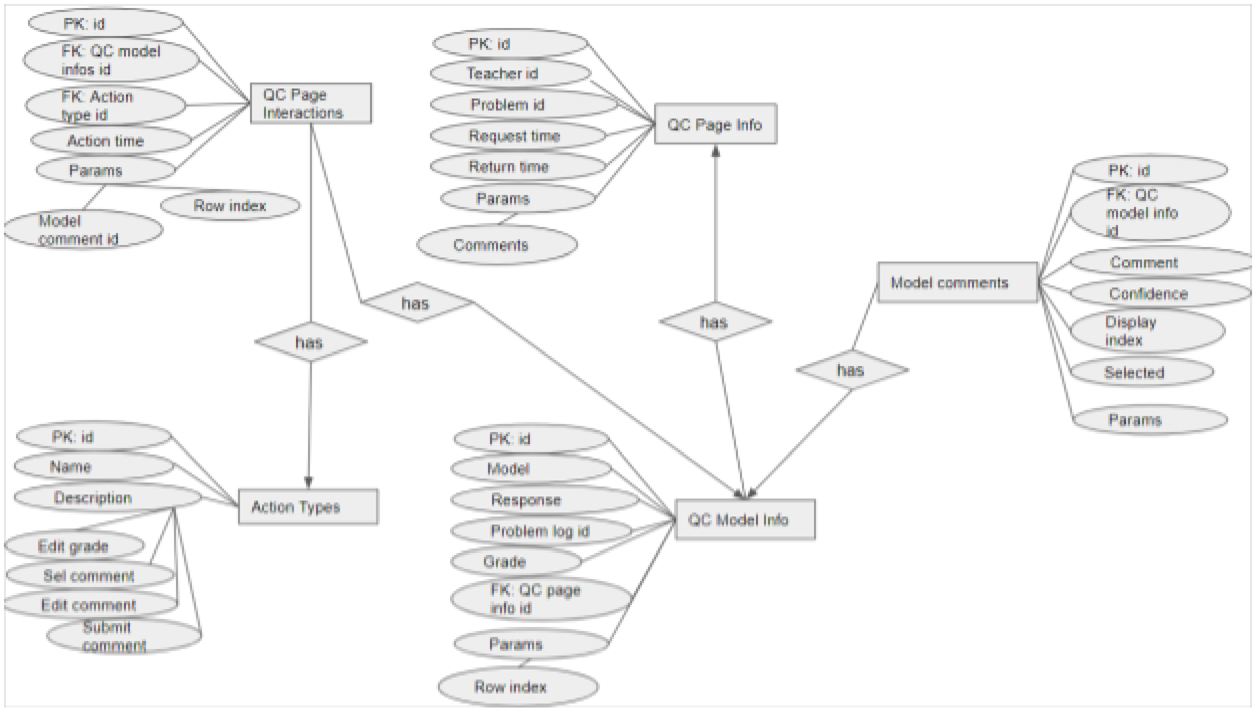


Figure 7: ER Diagram

- QC Page Interaction: record the teacher's interaction with the page
- Action Types: enumerates 4 possible teacher actions we want to log
- QC Page Info: record the page request and response information
- QC Model Info: record the corresponding model of the given problem
- Model Comments: record the detailed statistic model information

2.4 Preliminary Research

In the QUICK-Comments page, we need to generate comments based on the student answers, so we needed to look into natural language processing (NLP) methods with other feature learning techniques to train a proper statistical model for suggesting comments. After the analysis on the specific problem, we chose to use word embedding with comment clustering as our predictive model.

2.4.1 Word Embedding Method

Word embedding is commonly used language modeling method that maps words to an N-dimensional vector, and already has a number of specific implemented models, such as word2vec. The word embedding model we chose to use in our project is Global Vectors for Word Representation (GloVe) by the Stanford NLP Group. Like other word embedding methods, GloVe maps words to a vector, which can simplify the problem of analyzing the meaning of raw text to analyzing the meaning of a sequence of N-dimensional vectors. We used the 100 dimension pre-trained model based on the Wikipedia 2014 and Gigaword 5 datasets from the Stanford NLP site, which includes over 400,000 vocabulary words and characters. The following table shows the 5 most-frequently used words and characters and their corresponding 100 dimensional vectors:

Token	**100-D Vector Mapping**
the	[-0.038194, -0.24487 , ... , 0.27062]
,	[-0.10767 , 0.11053 , ... , 0.082577]
.	[-0.33979 , 0.20941 , ... , -0.028803]
of	[-0.1529 , -0.24279 , ... , 0.20664]
to	[-0.1897, 0.05002, ... , -0.1598]

2.4.2 Sentence Embedding Method

Similar to the word embeddings, sentence embedding is an alternative way to interpret the student answer text. Both word and sentence embedding are the methods to vectorize the students answers and to train the machine learning models discussed later. Because the student has answers to one problem, the embeddings are done for each problem. To do so, the words are tokenized and then translated into vectors as word embedding. Then, for each sentence, append all the 1x100 word representations into 1x100*N sentence representation where N is greater than 1. The representations of each sentence should be in the same length for clustering, therefore the longest length of the sentences of each problem is found and all the other sentence embeddings are padded with 0 to this length.

2.4.3 Comments Clustering

K-mean clustering was the method we chose to apply on the sentence embeddings. This part reaches results from the collaboration with Aaron G. Alphonsus, Johnathan A`Vant, and John A. Erickson. The possible optimal K was found from the Elbow Method by a given range of K values. From observation, a larger K has a lower distortion as shown in Figure 8.

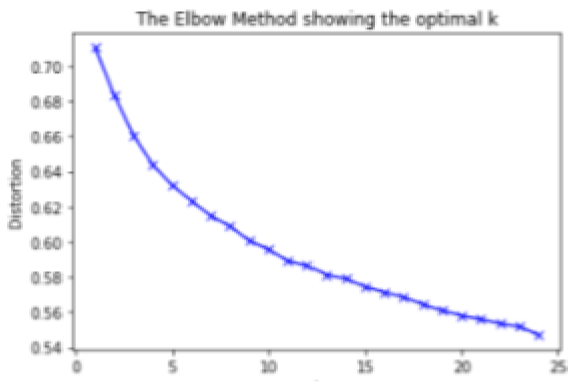


Figure 8: The optimal K from Elbow Method (Credit to Johnathan A`Vant)

From the pre-built Python clustering tool, the teacher comments are parsed into 13 clusters, shown in Figure 9. For each of the clusters, all teacher comments can be taken out for the analysis. An example is shown in Appendix D, which is the comments of cluster 12. In this example, all the comments have similar syntax and meanings. Therefore, the K-means clustering with sentence embedding could potentially be the appropriate method for this model.

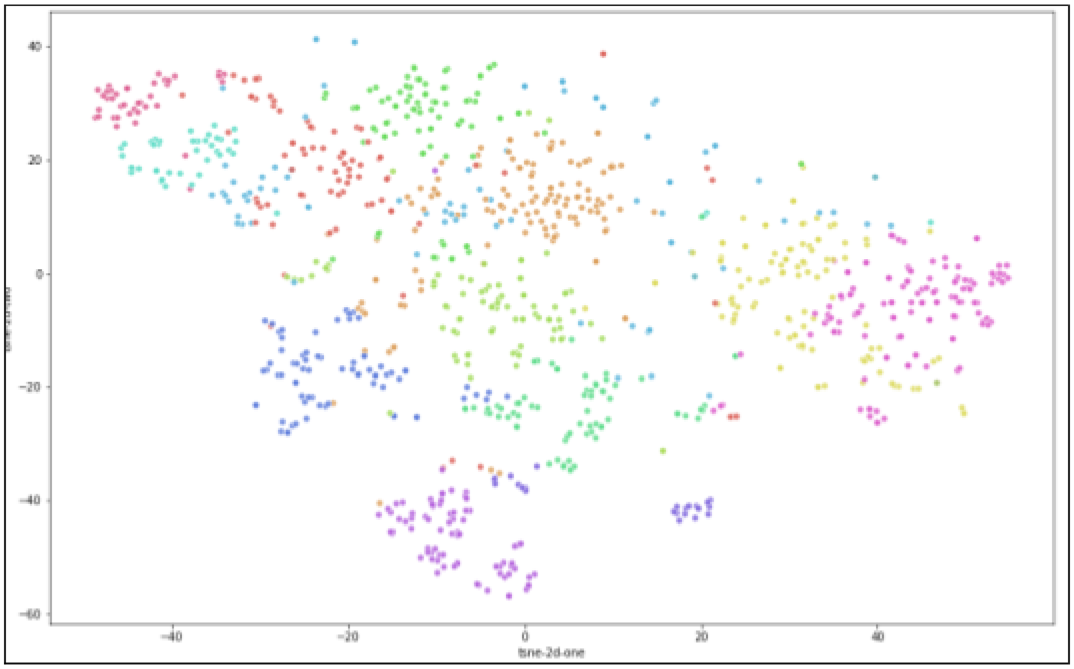
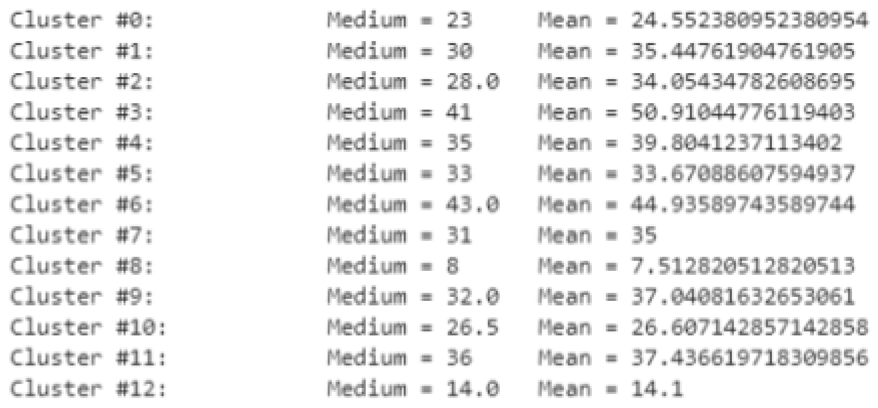


Figure 9: 13 Clusters (found from sentence embedding)

2.4.4 Generic and Specific Comments Analysis

One way to analyze the teacher comments is to find a split between the generic and the specific comments. A generic comment is defined as having a general instruction or piece of feedback that encompasses all students solutions to the problem, whereas a specific comment is defined to have a special feedback for one student for one problem. This analysis provides the length distribution of the comments, where the length of a comment is the number of words in the comment.

In the statistics in Figure 10, the medium and the mean of all the comments from each cluster are calculated. The median and mean of each cluster have approximate values indicating that the length distribution is normal. This is proven after Figure 10 is generated. Furthermore, it shows that the clustering method does not result in obvious skews.



```
Cluster #0:        Medium = 23     Mean = 24.552380952380954
Cluster #1:        Medium = 30     Mean = 35.44761904761905
Cluster #2:        Medium = 28.0   Mean = 34.05434782608695
Cluster #3:        Medium = 41     Mean = 50.91044776119403
Cluster #4:        Medium = 35     Mean = 39.8041237113402
Cluster #5:        Medium = 33     Mean = 33.67088607594937
Cluster #6:        Medium = 43.0   Mean = 44.93589743589744
Cluster #7:        Medium = 31     Mean = 35
Cluster #8:        Medium = 8      Mean = 7.512820512820513
Cluster #9:        Medium = 32.0   Mean = 37.04081632653061
Cluster #10:       Medium = 26.5   Mean = 26.607142857142858
Cluster #11:       Medium = 36     Mean = 37.436619718309856
Cluster #12:       Medium = 14.0   Mean = 14.1
```

Figure 10: Comment length statistics of the clusters

From observation, both median and mean can potentially be used as the split for this analysis. In the example cluster in Appendix D, the median and mean are both 14. For comments below the length of 14, they match the definition of generic comments and for the comments above this length, they match the definition of specific comments. Therefore, the median and mean could possibly be used to separate these two kinds of comments.

This observation needs further validation and proof. Generic and specific comments can be used to enrich the library of teacher comments in the baseline predictive model to get better results.

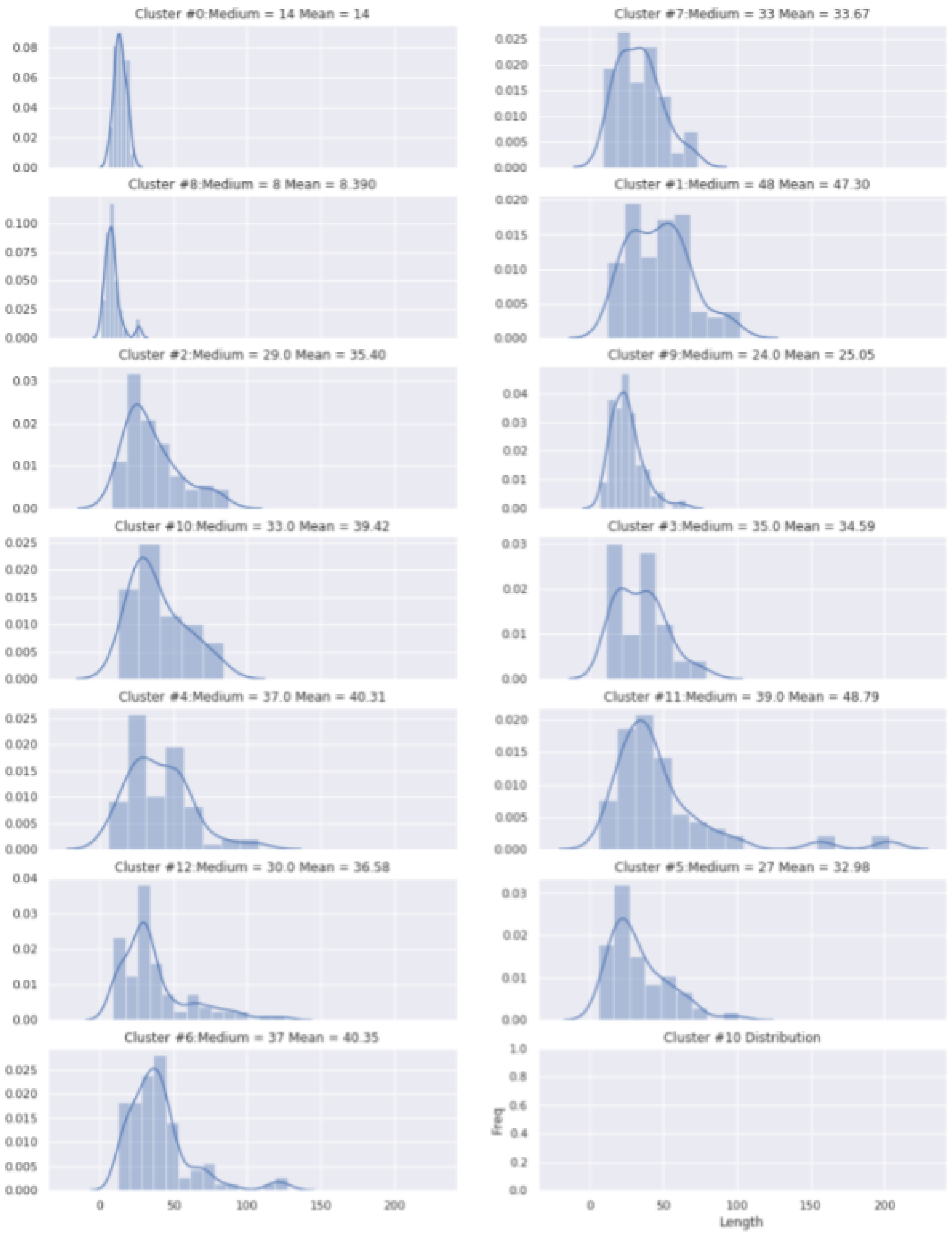


Figure 11: Comment length distribution of the clusters.

2.5 Logging

One of the most valuable tools in user driven production code is server side logging. Keeping track of important information, such as errors, timestamps of key events, etc., can prove to be useful when debugging unforeseen issues.

2.5.1 Python API Logging

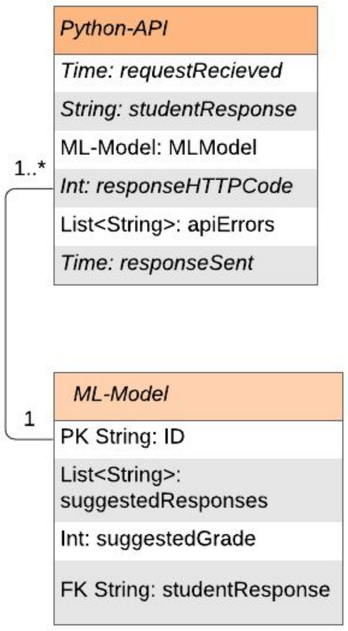


Figure 12: An ERD of the Python API Logger

Figure 12 depicts the schema of the information that is tracked by the logger in the Python API. First, the logger keeps track of the sent and received request times in order to calculate how long each request takes to process. It also keeps track of a *ML-Model* object which contains information about what models were used to grade the student responses. Finally, if any errors occur, these are logged as well to make debugging the production code easier. The next feature of this logger will be to write this data to one of the production databases rather than storing the information locally.

2.5.2 Future Logging

The logging for the entirety of this project was unfortunately not finished. This mostly due to the sheer amount of data that needs to be kept track of. If this project is to be continued, one of the things to help debug issues will be making this logger functional on a full stack level. Currently, only the API has a design, but client and server side information need to be tracked as well.

Conclusion

QUICK-Comments is a web page that will be able to expedite teacher grading processes and streamline classroom grading. We were able to develop a fully functional prototype that can grade open responses directly on the ASSISTments platform. When this project is carried forward, the webpage could be user tested to get empirical data as to how much time QC can actually save in a classroom setting. Combined with feedback on the user interface from teachers, this data can be used to make significant improvements over each iteration of the project. Being able to see which features succeed in making the process faster and which do not, would allow the project to be fine-tuned in order to maximize teacher grading efficiency. With the further improvements, QUICK-Comments will be able to help make teachers more efficient and move learning to a more smart classroom setting.

References

Pennington, J., Socher, R., & Manning, C. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

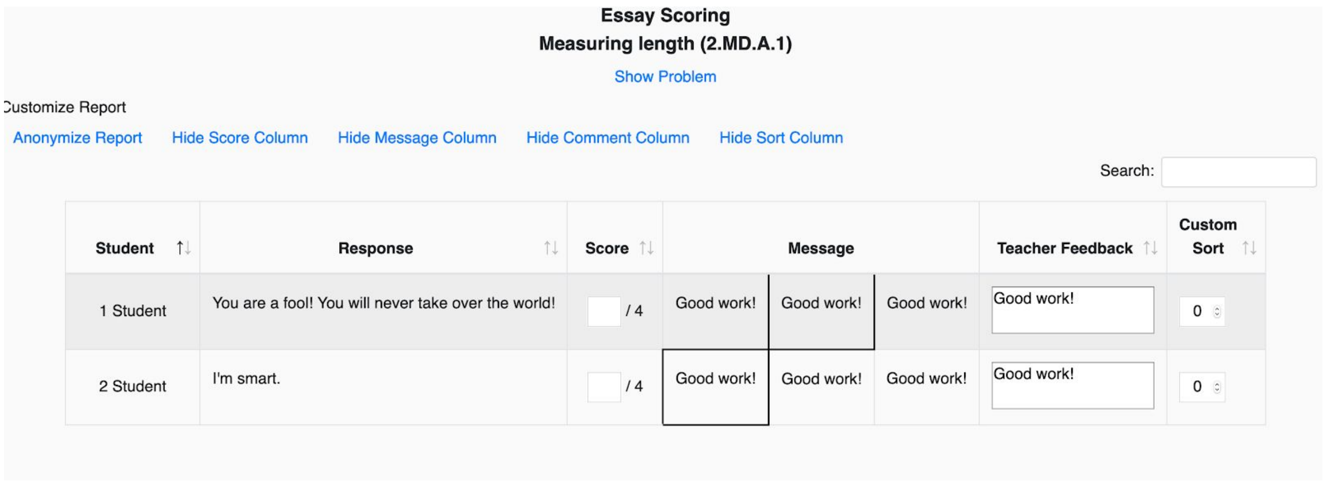
Bolukbasi, T., Chang, K. W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In Advances in neural information processing systems (pp. 4349-4357).

Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., ... & Ramavajjala, V. (2016, August). Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 955-964). ACM.

Hinkley, W. (n.d.). Free Education Tool for Teachers & Students. Retrieved from https://new.assistments.org/.

Appendices

Appendix A: Screenshots of Past Iterations

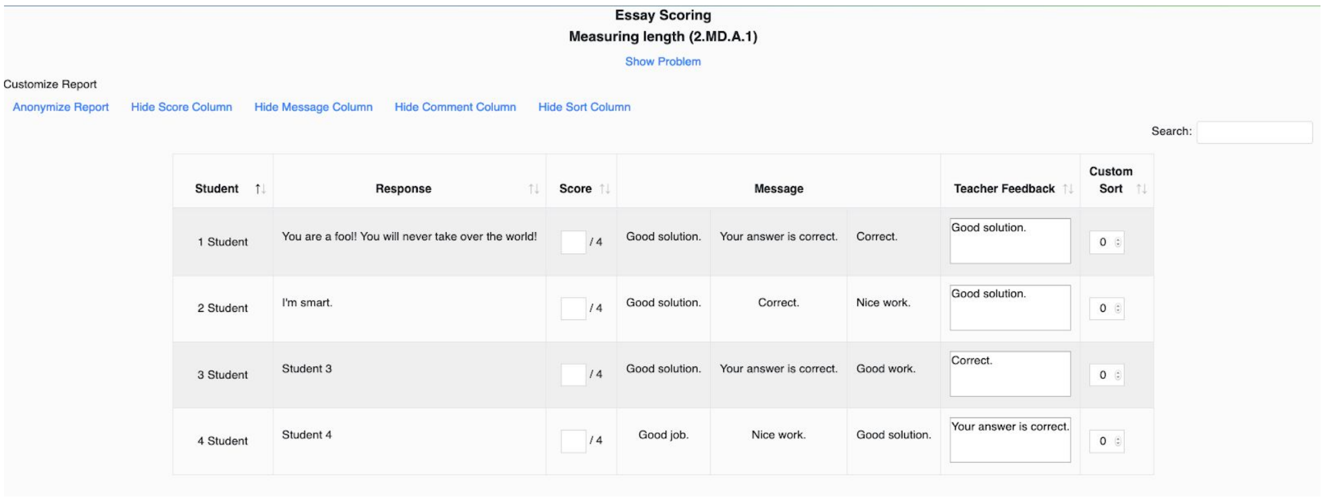


Essay Scoring
Measuring length (2.MD.A.1)
Show Problem

Customize Report
Anonymize Report | Hide Score Column | Hide Message Column | Hide Comment Column | Hide Sort Column

Search:

Student	Response	Score	Message			Teacher Feedback	Custom Sort
1 Student	You are a fool! You will never take over the world!	/ 4	Good work!	Good work!	Good work!	Good work!	0
2 Student	I'm smart.	/ 4	Good work!	Good work!	Good work!	Good work!	0



Essay Scoring
Measuring length (2.MD.A.1)
Show Problem

Customize Report
Anonymize Report | Hide Score Column | Hide Message Column | Hide Comment Column | Hide Sort Column

Search:

Student	Response	Score	Message			Teacher Feedback	Custom Sort
1 Student	You are a fool! You will never take over the world!	/ 4	Good solution.	Your answer is correct.	Correct.	Good solution.	0
2 Student	I'm smart.	/ 4	Good solution.	Correct.	Nice work.	Good solution.	0
3 Student	Student 3	/ 4	Good solution.	Your answer is correct.	Good work.	Correct.	0
4 Student	Student 4	/ 4	Good job.	Nice work.	Good solution.	Your answer is correct.	0

Appendix B: Table of Authors

Content	Author(s)
Abstract	John
1. Introduction	Zonglin & John
1.1 QUICK-Comments	Zonglin & John
1.2 ASSISTments	Alp & John
1.3 Platform	Alp & John
2. Methodology and Results	Alp, Connor, Rui, & Zonglin
2.1 Software Design	Alp
2.1.1: User Interface	Connor
2.1.2: User Testing	Connor
2.1.3: HTTP Request	Alp
2.1.4: HTTP Response	Alp
2.2 QUICK-Comments Server and APIs	Zonglin
2.2.1 Baseline Predictive Model	Zonglin
2.2.2 Python Server	Zonglin
2.3 Data Processing	Rui
2.3.1 Data Pre-processing and Analysis	Rui
2.3.2 Data Logging	Rui
2.4 Preliminary Research	Rui & Zonglin
2.4.1 Word Embedding Method	Rui
2.4.2 Comments Clustering	Zonglin
2.4.3 Generic and Specific Comments Analysis	Zonglin
2.4.4 Sentence Embedding Method	Zonglin
3. Conclusion	John

Appendix C: API Server Code (Python Flask)

```
"""QUICK Comments RESTful APIs
Give comments based on the given student answers
and the number of comments
"""
import os
import numpy as np
import json as js
from flask import Flask, request, render_template, jsonify, session, Markup
from trained_model.grade_prediction_function import predict_grade, predict_comment


__author__ = "Zonglin Peng, Rui Huang, and Meghana K V"
__copyright__ = "Copyright 2019, The ASSISTments Project"
__license__ = "GPL"
__version__ = "1.0.1"
__status__ = "Production"




'''MACRO'''
app = Flask(__name__)
app.secret_key = os.urandom(20)




'''HELPER'''
def get_comment_prediction(answer, num_comments, problem_ID, teacher_ID):
   comment_prob = predict_comment(answer, num_comments, problem_ID, teacher_ID)
   return comment_prob




'''API CALL'''
@app.route("/comment/", methods=['POST'])
def commnent_me():
   response_list = []
   raw_req = request.get_data().decode("utf-8") # get JSON as string and decode
   # print(raw_req) # DEBUG
```

```
    # print(type(raw_req)) # DEBUG
    req = js.loads(raw_req) if(type(raw_req) == str) else raw_req
    # parse req
    for json in req:
        response_dict = {}
        problem_ID = json["problem_ID"] # not plog_ID
        teacher_ID = json["teacher_ID"]
        user_ID = json["user_ID"]
        answer_text = json["answer_text"]
        num_comments = json["num_comments"]
        # answer_text = Markup(answer_text).striptags()
        if not answer_text:
            print("NOTE: No answers are passed in")
          session["answer"]  =  answer_text.strip('<p>').strip('</p>')  #  remove paragraph
tags
        # get comments
           comment_prob  =  get_comment_prediction(answer_text,  num_comments,  problem_ID,
teacher_ID)
        # parse data
        response_dict["user_ID"] = user_ID
        response_dict["comments"] = comment_prob
        response_list.extend([response_dict])
    # parse into JSON
    session["comments"] = js.dumps(response_list)
    print(session["comments"])
    return session["comments"]




@app.route("/_cleanup")
def cleanup():
    session.clear()


@app.route('/')
def hello_world():
    return 'Hello World!'


'''MAIN'''
if __name__ == '__main__':
    app.run()
```

Appendix D: Comment Clusters

- - - - - Cluster 12 - - - - -

length 6 : I can show you in class.

length 6 : Please use this space for help.

length 9 : Show me some work and I can help you.

length 10 : If you provide your work here I can help you.

length 10 : I can help you if you give me useful feedback.

length 11 : I can encourage you if you provide some useful feedback here.

length 11 : I can help you if you provide some useful feedback here.

length 11 : I can help you if you give me some useful feedback.

length 11 : I can help you if you show me some work please?

length 11 : I can help you if you provide what you tried here.

length 11 : I can help you if you show me what your thinking?

length 12 : I can help you if you privide your work on this here.

length 12 : I can help you if you give me some workable feedback here.

length 12 : I can help you if you could provide some useful feedback here.

length 12 : I can help you if you could provide some useful feedback please.

length 13 : I can help you if you provide your thoughts on this problem here.

length 14 : Please make an attempt to answer all questions. Reviewing scale drawings would be helpful.

length 14 : Please make an attempt to answer all questions. Reviewing scale factors would be helpful.

length 14 : Please make an attempt to answer all questions. Reviewing proportional relationships would be helpful.

length 14 : Great start, but please make sure you fully describe how you determined the answer.

length 14 : I can help you if you provide some of your thinking on this problem.

length 15 : I can help youif you provide some of the ideas you tried on this here.

length 15 : Great - make sure to turn in your paper copy to your teacher for feedback.

length 15 : Please make an attempt to answer all questions. Reviewing ratios and equations would be helpful.

length 15 : Please make an attempt to answer - use your resources to seek assistance if needed.

length 15 : I can help you if you provide some of your work here that you tried.

length 16 : Please make an attempt to complete each problem. It will help to review ratios & proportions.

length 16 : Please make an attempt to complete each problem. It will help to review ratios and proportions.

length 16 : I can help you if you provide some of the ideas you tried on this here.

length 16 : You should type your explanation or take a picture of your work and upload it here.

length 16 : I can help you here or in class if you give me something to work with.

length 16 : Please make an attempt to answer all questions. Reviewing relationships between scaled copies would be helpful.

length 16 : Please make an attempt to complete each problem. It will help to review ratios and proportions."

length 18 : Please make an attempt to respond to every problem. Use your resources or ask for assistance if needed.

length 19 : Be sure to show me your work/explanation during class so that I can give you feedback on this problem.

length 19 : Be sure to show me your paper before/during class so that I can give you feedback on your answer.

length 20 : I can help you if you can provide me with some work or some idea of what you tried here?

length 20 : Be sure to show me your paper before/during class so that I can check your work and give you feedback.

length 20 : Be sure to show me during/before class so that I can check your work and provide feedback to your response.

length 23 : I can help you if you could provide some useful feedback please.I can help you if you could provide some useful feedback please.