A Grading Tool to Simplify the Process of Grading Papers

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Andrew Song

Spring, 2022

Technical Project Team Members

Andrew Wang

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Rosanne Vrugtman, Department of Computer Science

A Grading Tool to Simplify the Process of Grading Papers

CS4991 Capstone Report, 2022

Andrew Song
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
andrewsong.j@gmail.com

Abstract
Grading papers can often involve writing and typing the same things again and again, leading to a time-consuming process. To address this problem, our research team created a web application to simplify and streamline the paper grading process. Designed with a React frontend and Django backend, the application enables users to annotate online submissions quickly with pre-set comments that are submitted by inputting the assignment's rubric into a separate interface contained in the application. After about three months of work, our results consisted of a largely functional prototype application. In the future, to improve the application, more features can be implemented and more testing with annotating and overall bug-fixing could be performed.

1. Introduction

In a time when we are utilizing mobile or web applications and platforms more than ever in the field of education, how can we utilize these apps and platforms when it comes to one of the most time-consuming tasks in the classroom: grading? While many applications have been created to simplify/automate the process of grading assignments that have pre-determined answers, a different approach is required for assignments that have open-ended answers, such as papers and essays. One approach is to create an interface that speeds up the process of writing feedback by taking advantage of the idea that professors oftentimes find themselves giving similar feedback over and over. With this in mind, our team aimed to simplify the grading process and create an app to save professors much time and effort.

2. Related Works

One of the earliest works in this area was by Golovchinsky, et al, (1998). This work created a method of handwritten digital annotations enabling users to directly and freely annotate documents with digital ink. While our application does not use free form to create annotations, this early work is a sort of foundation for future digital annotation tools.

Another early paper in this area by Bullock, et al. (1998) is also one of the most significant. The paper details the effects of several annotation interface designs on the user experience during annotation. The ideas in this paper offer inspiration for our own annotation interface design.

Other works creating annotation applications more specifically for grading include the JANE system introduced by Hahn, et al (2007). The JANE system uses machine learning to learn annotation patterns, saving time for users while annotating. While this system uses artificial intelligence to create annotation shortcuts, our system uses direct user input to create shortcuts.

3. System Design

To detail our system design, we split up the section into 4 parts: 1) a review of the system architecture, 2) the requirements for our web application, 3) the key components of our application, and 4) the challenges that were involved with development.

3.1. Review of System Architecture

The system was a full stack web application that used the Reactjs library to design the frontend and the Django framework to organize the backend. Thus, the frontend was written largely in javascript and the backend was written in python. For data operations, the database used during development was an SQLite database while the database used during production was a PostgreSQL database. For deployment of the application, we hosted the backend on Heroku and hosted the frontend on a service called Netlify.

3.2 Web Application Requirements

When we first designed the web application, the most important requirement was having an interface that supported annotating a paper. This interface would need to allow the user to highlight a piece of text and use a sidebar, on the left side of the interface, to click and create their comments. Additionally, the annotations that were created would need to be visible next to a button, on the right side of the interface, that displayed details about the annotation made. Further, in order to make a list of pre-set comments that the user could click and add as an annotation, there would have to be a separate interface where users could either upload or create their own rubric for the assignment that they would be grading. Outside of the annotation interface, the application also needed to be able to parse a separate website where students submitted their work so that the submissions could be displayed inside our annotating interface.

3.3 Key Components

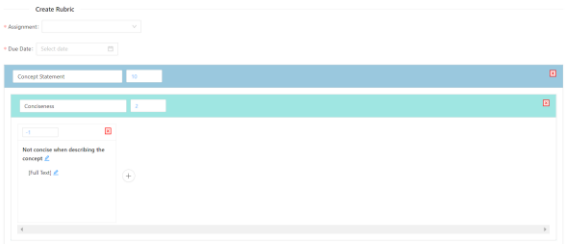

Figure 1: Rubric Creation Interface

In the rubric creation interface, users can create the rubric from pre-made comments so that users can click them easily when annotating. Each comment exists under two levels. In Figure 1, the first level is the Concept Statement which amounts to a total of 10 possible points. Within the Concept Statement section, the Conciseness makes up 2 of those 10 points. Finally, the comment for lacking conciseness is under the Conciseness section, and users can take off 1 point for this.

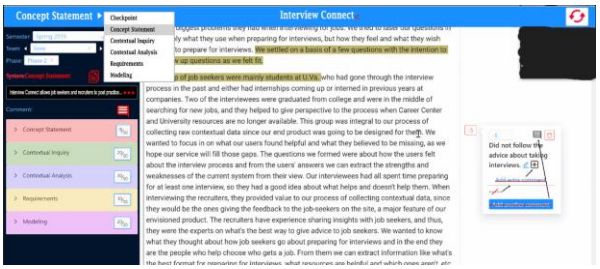

Figure 2: Overview of Annotation Interface

Figure 2 shows the annotation interface of the application at the center of the page is the paper/submission. This submission is scraped from the submission site and displayed into an iframe at the center. In there, users can highlight text, and create an annotation using the sidebar at the left. On the left, each of the rubric elements appear in a list, and can be selected once the text is highlighted to add a comment.

There are also a number of other features that are a part of this interface. First,

although five points were taken off for this particular comment, this point total can be adjusted if the grader wants. Further, the actual text of the annotation can be changed by clicking the edit icons at the end of the text. In the top left corner of the page, the grader can navigate between the groups to easily select a new paper to grade and annotate.

3.4. Challenges

There were a lot of difficulties in creating the annotation interface. One of the main ones was getting the paper displayed correctly inside the iframe. There were times when formatting and html elements would be lost when displaying the paper into the iframe. To address this, we did some processing on the html of the paper, but we still ran into occasional, albeit much more minor, losses of formatting. Additionally, getting the annotation box aligned in the correct spot along with the spot of the highlighted text was a challenge. Eventually, we were able to get the boxes located in the correct spots, but occasionally, the location would be off by a little bit.

4. Results

After about three months of work, we were able to create a working prototype of the grading application. However, it was not a full-fledged application ready for adaptation in the classroom. All of the initial key requirements were satisfied, but a number of smaller, additional features were not implemented. There were also some minor bugs in the application. Professor Apostolellis used the grading tool application briefly with his TAs at the beginning of the Fall 2021 semester and tested it. Despite some missing features and small bugs, the grading process itself was quicker and faster. At the end of our time working on the project, we documented the process of what we were doing, what we had done, and what the current problems were so that future workers would more easily understand the current state of the application.

5. Conclusion

Through this project, we have created a prototype and model for an annotation application that saves time for teachers while grading papers. One of the most important elements of this project is the mechanism through which the application aims to save time—by allowing professors to input their rubrics for a given assignment so that pre-written annotations may be generated and easily added to submissions. Additionally, the annotation user interface designed in this project may serve as a model for future related works. Ultimately, we have potentially created the foundations for tool that teachers can use to save time while grading papers.

6. Future Work

The next phase for this project would involve further testing and debugging. Our prototype has been minimally tested and some bugs have been found in testing. Future work would also involve implementation of various functionalities for the tool that have not yet been completed. One important functionality is the ability to create different types of accounts (e.g. students, teaching assistants, and professors) and put them into groups to support group work. Additionally, a number of interface specifications have not yet been realized, thus, future works would build on the user interface of this project.

7. Acknowledgements

I would like to thank Professor Apostolellis and Andrew Wang for their help and collaboration throughout the development process.

References

Schilit, B.N., Golovchinsky, G., & Price, M.N. (1999). Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. *Proceedings of CHI '98*, 249-256. ACM Press, 1998.

Tomanek, K., Wermter, J., & Hahn, U. (2007). Efficient Annotation with the Jena ANnotation Environment (JANE). *Proceedings of the Linguistic Annotation Workshop*, 9-16. Association for Computational Linguistics.

Wojahn, P.G., Neuwirth C.M., & Bullock B. (1998). Effects of Interfaces for Annotation on Communication in a Collaborative Task. *CHI98: Human Factors in Computing Systems, Conference Proceedings*, 456–463, ACM 1998.