Direct automated feedback delivery for student submissions based on LLMs

MAXIMILIAN SÖLCH, Technical University of Munich, Germany

FELIX T.J. DIETRICH, Technical University of Munich, Germany

STEPHAN KRUSCHE, Technical University of Munich, Germany

Receiving timely and personalized feedback is crucial for students' learning progress and motivation. However, providing such feedback poses a significant challenge in education, particularly as student numbers have steadily increased in recent years. This growth has made it difficult for tutors and professors to deliver individualized feedback to each student, resulting in a time-consuming, repetitive, and often manual task that contributes to a heavy workload for educators.

This paper presents DAFeeD, a large language model (LLM)-based approach for providing automated formative feedback on student submissions. The exercise-independent approach can be applied to various domains, such as programming, text, or modeling exercises. By incorporating exercise-specific information, such as the problem statement and grading instructions, into the prompt, the feedback is tailored to the exercise context.

We implemented this approach in an open-source reference implementation named Athena, which is integrated with the learning platform Artemis. To evaluate the effectiveness and efficiency of the approach, we conducted a controlled study with students. The results show ...

CCS Concepts: • **Social and professional topics** → **Student assessment**; • **Applied computing** → **Education**.

Additional Key Words and Phrases: Do, Not, Us, This, Code, Put, the, Correct, Terms, for, Your, Paper

1 INTRODUCTION

In the current educational landscape, providing timely and effective feedback to students remains a significant challenge. Traditionally, students must wait for course tutors or professors to review their submissions and provide feedback. This process can be time-consuming, often requiring students to arrange meetings and wait for available time slots, which are not always convenient or immediate. Similar it is timecosuming and tedious for professors and tutors to provide asynchronous feedback via email or other communication channels. The inherent delays and scheduling difficulties make this approach not scalable, especially in courses with a large number of students.

These limitations hinder students' learning progress and motivation. The waiting period for feedback interrupts the learning flow, causing students to lose momentum and potentially disengage from the subject matter. Additionally, the limited availability of tutors and professors means that not all students receive the individualized attention they need to improve their understanding and skills. This situation underscores the necessity for a more efficient and scalable feedback system that can provide continuous support to students without the constraints of traditional methods.

In this paper, we present an approach for generating automated feedback on student submissions using large language models (LLMs) to address these challenges. The approach is independent of the exercise type and can be applied to various domains, such as programming, text, or modeling exercises. We implemented the approach in an open-source reference implementation called Athena, connected to the learning platform Artemis through which students submit their solutions and receive feedback. To validate the effectiveness and efficiency of the approach, we tested it in a controlled environment. We collected quantitative and qualitative data to evaluate students' perceptions of the approach and the overall performance of the reference implementation. The results show

The subsequent sections of this paper are organized to provide a comprehensive understanding of the research. Section 2 provides an overview of related work. Section 3 details the concept and methodology of Direct Automated Feedback Delivery (DAFeeD). Section 4 describes the reference implementation of DAFeeD, called Athena, including a general overview, details on the used prompts, and the system architecture. Section 5 presents the evaluation results, Finally, Section 6 concludes with a summary of findings and discusses future research directions to enhance automated feedback systems.

2 RELATED WORK

The necessity and challenges of providing effective feedback in educational settings have been widely recognized. This section overviews prior work in three interrelated areas: automated feedback systems, the application of large language models in educational feedback, and feedback delivery mechanisms within learning platforms.

2.1 Automated Feedback Systems

todo [2]

2.2 Large Language Models in Educational Feedback

2.3 Feedback Delivery in Learning Platforms

3 APPROACH: DIRECT AUTOMATED FEEDBACK DELIVERY (DAFEED)

DAFeeD employs large language models to deliver automated feedback on student submissions, designed to complement traditional teaching methods and provide additional support. Figure 1 illustrates the continuous feedback workflow that DAFeeD facilitates, enabling students to receive feedback at any time, thereby eliminating the need to wait for responses from human tutors or course professors.
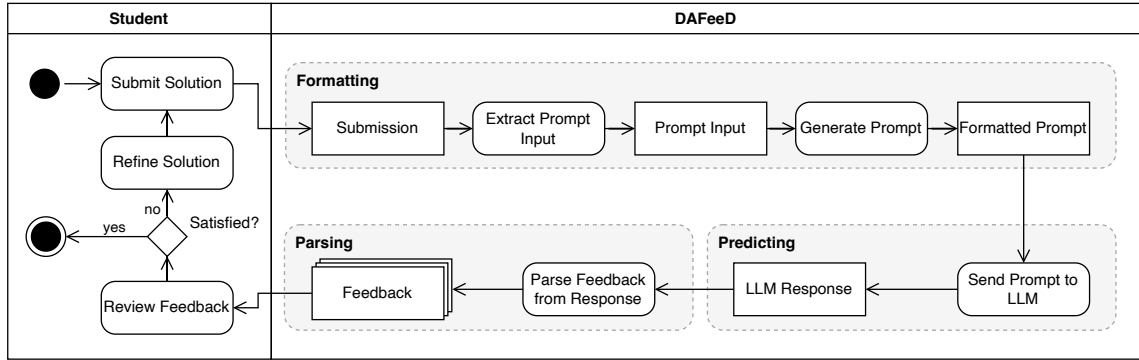

Fig. 1. Workflow of direct automated feedback delivery for students' submissions (UML Activity Diagram)

DAFeeD can provide feedback on various aspects, such as the correctness of the code, the quality of the code, and the performance of the student. Once the student submits their solution, DAFeeD initiates a three-stage process to generate natural language feedback.

The first stage, called *Formatting*, takes the student's submission and extracts the submission content, problem statement including learning objectives, and any possible grading instructions the instructor defines. This extracted information represents the prompt input. During the prompt generation step, a predefined prompt template is filled with the prompt input data, resulting in a formatted prompt.

In the second stage, called *Predicting*, the formatted prompt is sent to a large language model (LLM), which generates a response that includes detailed feedback for the student.

The final stage, *Parsing*, takes the LLM response, which comes in the JSON format, and parses feedback items from it. In addition to the feedback text, the feedback object also contains reference information indicating the part of the submission it pertains to. For programming exercises, this includes the file name and line number of the relevant code snippet to which the feedback refers. For text exercises, the reference information includes only the sentence or word range the feedback refers to.

All of the feedback is then returned to the student for review. If the student is satisfied with the feedback, the process concludes. Otherwise, the student can refine their solution and resubmit it, initiating the DAFeeD process anew.

This iterative process is designed to motivate students to continuously learn and experiment with their solutions, resulting in improved performance.

4 REFERENCE IMPLEMENTATION: ATHENA

We incorporated DAFeeD into a reference implementation named Athena, which is seamlessly integrated with the learning platform Artemis. Through Artemis, students can submit their solution and review the feedback.

When submitting their solutions on Artemis, students have the option to request direct automated feedback by clicking a newly added button. This feedback request is then sent to Athena, provided the student has not reached their feedback request limit for the exercise. Course instructors can customize the number of allowed feedback requests per exercise according to their preference. A status visualization informs students about their feedback request state. Once Athena generates the feedback and sends it back to Artemis, the student can review it in a modal window on Artemis, as depicted in Figure 2.
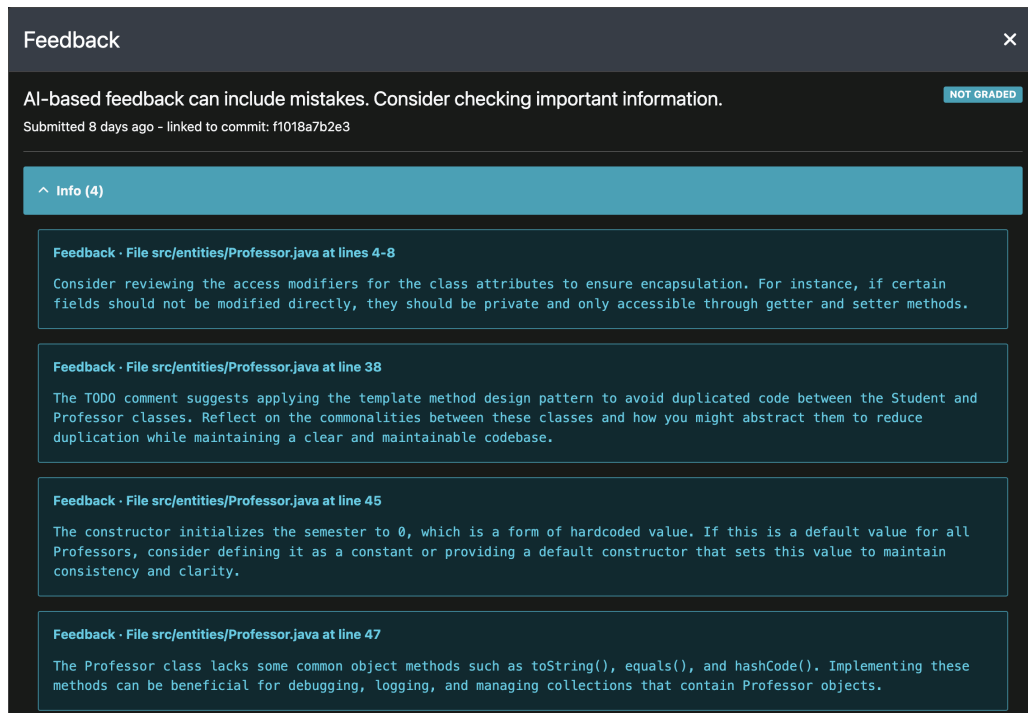

Fig. 2. Visualization of the feedback how students see it in Artemis.

4.1 Prompts

4.2 Feedback Generation

4.3 Architecture

Athena is deployed in production alongside the learning platform Artemis, which serves up to more than 2000 students per course. Consequently, the reference implementation must satisfy additional non-functional requirements such as performance, scalability, maintainability, and usability. To meet these requirements and to support feedback generation for multiple exercise types while allowing for future extensibility, we adopted a modular architecture, as illustrated in Figure 3.

The *Module Manager* handles all incoming requests, verifies authorization, and forwards them to the appropriate modules. The *ProgrammingLLM* module manages programming exercises and executes the three-stage DAFeeD process,
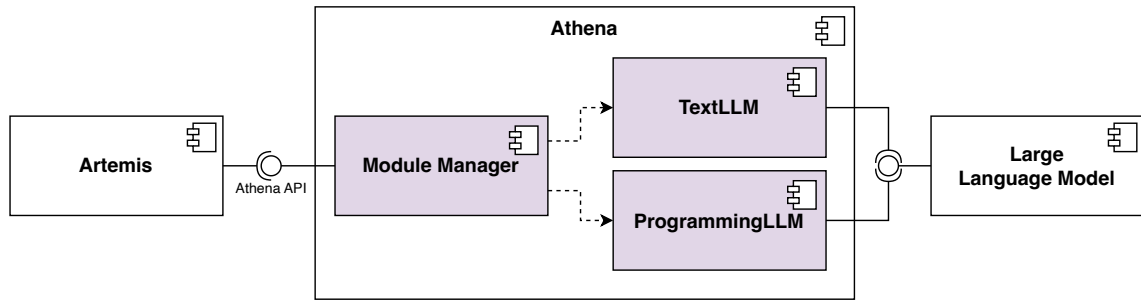
Fig. 3. Top Level Archietecture of the reference implementation Athena (UML Deployment Diagram)

which includes formatting, predicting, and parsing. Similarly, the *TextLLM* module is optimized for text exercises and follows the same process.

Athena's system design is independent of any specific learning management system (LMS) as it provides a REST API, documented using the OpenAPI standard[1]. This independence allows Athena to be integrated with various LMS platforms, such as Moodle[2].

Athena currently connects to OpenAI models hosted in a private Azure cloud to ensure that student data is not used for training models, maintaining privacy. Additionally, the system can be configured to use open-source models like Llama[3] or Mistral[4], either self-hosted or cloud-based.

To meet performance and scalability requirements, Athena and its modules are deployed within a Kubernetes cluster[5]. Kubernetes, in conjunction with Athena's modular architecture, allows the system to scale each module independently. For example, additional instances of the programming module can be instantiated when a new programming exercise is released. Furthermore, Kubernetes provides out-of-the-box load balancing and self-healing capabilities, ensuring that if a module crashes, it is automatically restarted.

5 EVALUATION

In this section, we outline the methodology employed to validate the effectiveness of the proposed DAFeeD approach including the reference implementation Athena. The conducted evaluation represents the treatment validation stage of the design science methodology proposed by Wieringa [3]. In this stage, the proposed solution — DAFeeD — is evaluated in a controlled environment, and the collected data is utilized for the refinement and improvement of the solution.

We begin by describing the research questions, followed by the study design and the results. Subsequently, we outline the limitations of the evaluation and discuss the implications of the findings.

5.1 Research Questions

With this study, we want to answer the following research questions about direct automated feedback delivery:

RQ1 How do students perceive the effectiveness of direct automated feedback?

---

[1]https://www.openapis.org
[2]https://moodle.org
[3]https://llama.meta.com
[4]https://mistral.ai
[5]https://kubernetes.io

RQ2 How does the availability of direct automated feedback affect student engagement and motivation?

RQ3 Do students feel more comfortable requesting automatic feedback than asking a human tutor or the course professor?

RQ4 How do students perceive the usability and helpfulness of DAFeeD?

5.2 Study Design

To gain comprehensive insights into students' perceptions of the newly introduced DAFeeD concept, we employed a survey-based approach. Initially, study participants tested the new feedback feature on a sample exercise using the Artemis platform in a controlled environment at the university. The study conductors were present throughout to answer questions and provide support as needed. Following this hands-on experience, participants were asked to complete a survey hosted on the community version of the open-source survey tool LimeSurvey[6]. This survey aimed to gather their opinions on direct automated feedback and collect feedback on their overall experience with the feature.

We invited students from current courses at the university to participate in the study via direct messages and conducted the study with a total of 20 participants. The study uses a mixed methods approach, combining quantitative and qualitative data collection methods. The participants were a mix of undergraduate and graduate students from various disciplines, including computer science, information systems, and games engineering.

All survey questions, except for the introductory demographic queries and the final, voluntary free-text responses, employ a 5-point Likert scale [1] and are mandatory. The survey commences with a series of introductory demographic inquiries, including the student's current study program, ongoing degree pursued, and academic semester. Additionally, respondents are asked to provide information regarding the hardware employed, including details on the operating system and web browser used. The final question in this group asks participants to describe their programming experience.

Following the demographic section, the survey explores participants' opinions on the impact of direct automated feedback on student engagement and motivation. Subsequently, the survey queries participants on their comfort with the feedback source, specifically whether they prefer receiving feedback from a human tutor or from the DAFeeD feature. The survey then investigates aspects related to participants' perceived effectiveness of the DAFeeD feature. In the final question group, participants assess the overall usability and helpfulness of the DAFeeD feature.

5.3 Results

5.4 Limitations

5.5 Discussion

6 CONCLUSION & FUTURE WORK

Future work includes enhancing the visualization of feedback, such as grouping and color coding the feedback items to make it easier to differentiate between critical feedback items, suggestions for improvement, and positive feedback. A high priority will be on further improving the overall quality of the feedback provided. We also aim to extend the implementation to support direct automated feedback for the remaining exercise types of Artemis. Another crucial step is to test direct automated feedback in a real-world setting by utilizing this feature in an actual course. This will allow us to collect comprehensive data to thoroughly evaluate the impact on student performance and motivation.

---

[6]https://www.limesurvey.org

REFERENCES

[1] I Elaine Allen and Christopher A Seaman. 2007. Likert Scales and Data Analyses. *Quality progress* 40, 7 (2007), 64–65.

[2] Marcelo Guerra Hahn, Silvia Margarita Baldiris Navarro, Luis De La Fuente Valentin, and Daniel Burgos. 2021. A Systematic Review of the Effects of Automatic Scoring and Automatic Feedback in Educational Settings. *IEEE Access* 9 (2021), 108190–108198. https://doi.org/10.1109/ACCESS.2021.3100890

[3] Roel J. Wieringa. 2014. *Design Science Methodology for Information Systems and Software Engineering.* Springer Berlin Heidelberg.