Toward the Automatic Assessment of Text Exercises

Jan Philip Bernius
Department of Informatics
Technical University of Munich
Munich, Germany
janphilip.bernius@tum.de

Bernd Bruegge
Department of Informatics
Technical University of Munich
Munich, Germany
bruegge@in.tum.de

***Abstract*—Exercises are an essential part of learning. Manual assessment of exercises requires efforts from instructors and can also lead to quality problems and inconsistencies between assessments. Especially with growing student populations, this also leads to delayed grading, and it is more and more difficult to provide individual feedback.**

The goal is to provide timely responses to homework submissions in large classes. By reducing the required efforts for assessments, instructors can invest more time in supporting students and providing individual feedback.

This paper argues that automated assessment provides more individual feedback for students, combined with quicker feedback and grading cycles. We introduce a concept for automatic assessment of text exercises using machine learning techniques. Also, we describe our plans to use this concept in a case study with 1900 students.

I. Introduction and Problem

Instructors face a large population of students in their courses. Students require feedback on their exercises to reflect on their progress [1]. The concepts of interactive learning [2, 3] helps to increase the interaction between instructors and students but also increases the workload for instructors. Software engineering students need to learn constructive and creative capabilities. It is important for the instructor to facilitate the problem-solving learning process. Concrete problem-solving strategies are taught in paradigms, accepted by the profession [4]. Each paradigm provides a set of problem-solving exercises. These are usual textual exercises that involve the application of problem-solving techniques.

Exercises are a proven method to train higher cognitive skills including the acquisition of domain-specific knowledge, analysis and design methods and the evaluation of the results. Trivial exercises, such as multiple-choice quizzes, do not stimulate higher cognitive skills and do not reflect engineers daily work [1].

Exercises help students to learn, understand and apply a paradigm. A student needs feedback to reflect and improve on their solution to the exercise. Text exercise assessment causes time-intensive efforts with instructors, preventing them from spending time on improving their lectures, having discussions with their students or update exercises to incorporate technology evolution.

Increasing student populations make it harder to keep assessments fair and at equal quality. Students do not benefit from quantitative feedback alone [5]. Qualitative feedback helps students to improve.Splitting assessment efforts with multiple instructors can lead to inconsistencies. Providing timely or instant feedback in a large class is hard [6]. Waiting for feedback delays the students learning progress and hinders interactive learning. We strive toward a system to provide automated text assessments based on instructor feedback decreasing student feedback waiting times.

This paper is structured as follows: Section I introduces the domain and outlines the problems with the current correction process for text exercise. Our vision is described in Section II in the form of a visionary scenario. Section III describes the assessment workflow of a possible implementation and Virtual One-To-One, a machine learning based mechanism for providing individualized feedback for students in large classes. Section V discusses applicability and limitations of the system. We present related work in Section VI. Section IV proposes our evaluation approach, and Section VII concludes the paper.

II. Visionary Scenario

The following scenario describes how we envision to improve the assessment of text exercises:

Anna and Tom are students participating in a software engineering course. During a lecture, the instructor starts an in-class text exercise to be completed in the assessment system. Anna and Tom both submit a solution to the system. The instructor starts manually assessing a set of submissions selected by the system. The system asks the instructor to assess Annas solution. The instructor provides a score and a comment explaining his assessment. After receiving the assessment, the system decides to assess Toms solution automatically based on the assessments provided previously. Anna and Tom get individual feedback for their solution to reflect on their learning progress.

Tom is not satisfied with his submission after receiving his feedback. He decides to improve his work and resubmits a refined version of his solution. The system automatically assesses Toms resubmission and provides a new assessment. Tom is now satisfied with his assessment and fished the exercise.

III. Assessment Workflow

In a first prototypical implementation, we extend the ArTEMiS system, already capable of assessing programming and modelling exercises automatically [1, 7], by adding semi-automated text assessment. A student submits his solution for

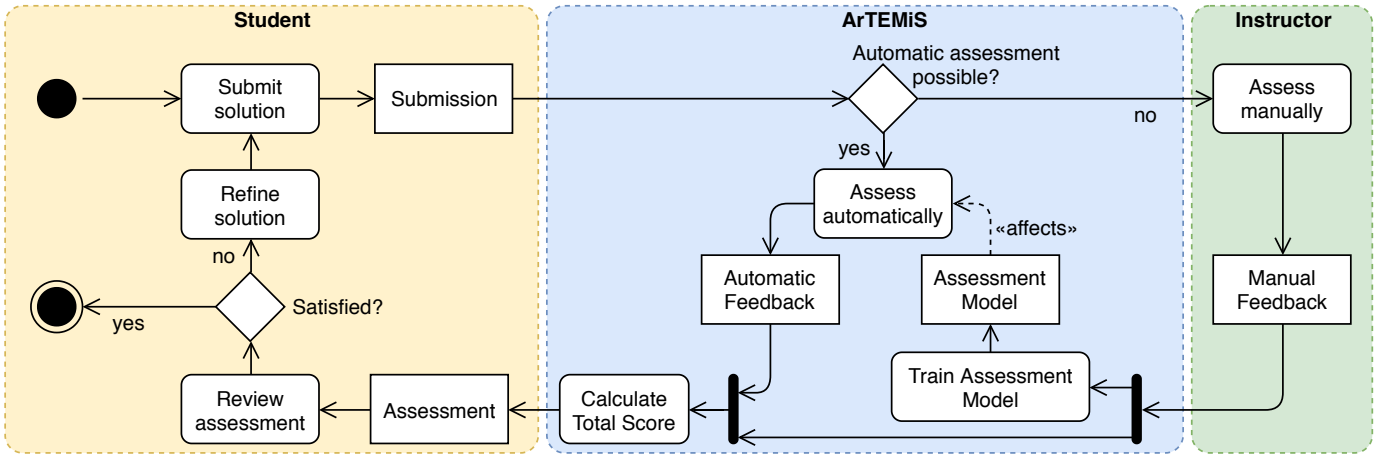

Fig. 1. Automatic assessment workflow, considering manual and automatic assessment.

a text exercise to the ArTEMiS system. The activity diagram in Fig. 1 depicts the assessment workflow. The system supports two means of assessment: Manual assessment provided by the instructor (Section III-A) and automatic assessment generated by the system based on an assessment model (Section III-B). ArTEMiS decides which assessment method is required for each submission based on the quality of the assessment model. Both means of assessment provide a set of Feedback Items.

The assessment of the submission is a composition of all feedback items. The final score is the sum of all feedback scores (see Fig. 2). Student review the assessment of their submission. If they are not satisfied, they can submit a refined solution for assessment, enabling continuous interactive learning [1] with text exercises.

A. Manual Assessment

ArTEMiS selects text exercise submissions for manual assessment by instructors if the assessment model does not allow for a confident assessment. Instructors are used to grading exercises using a set of rubrics. A rubric defines a set of traits of the students' submission, which are evaluated based on a scale [9]. Rubrics can exist in different levels of detail, such as only listing aspects of the assignment or defining different scoring levels. If instructors do not define a rubric beforehand explicitly, they build a rubric in their mind while assessing.

Instructors break down a submission into blocks and match each block with a rubric. As illustrated in Fig. 3, instructors define text blocks themselves as a phrase, sentence or paragraph by selecting a piece of text as they see fit. They assess each block quantitatively and qualitatively using a score and a feedback comment (see *Feedback* in Fig. 2).

B. Automatic Assessment

ArTEMiS assesses submissions automatically, if the quality of the assessment model allows for a confident assessment. The assessment model is trained based on the manual assessments of text blocks provided by instructors. Fig. 4 depicts the automatic assessment process. For automatic assessment, submissions need to be broken down to text blocks automatically, first. Second, a vector representation of the text blocks is calculated as an input value for further computations. Third, the assessment needs to be generated for each text block.

A first, simple approach is using sentences as text blocks. We split submissions into sentences using delimiter characters (. : ? !) or line breaks. In a later stage, we plan on applying techniques such as topic modelling for text block calculation if the simple approach does not provide sufficient results. All text blocks need feedback to complete an assessment.

ArTEMiS calculates a vector representation for each text block. Therefore, blocks are translated into a multi-dimensional vector space, following the word2vec algorithm

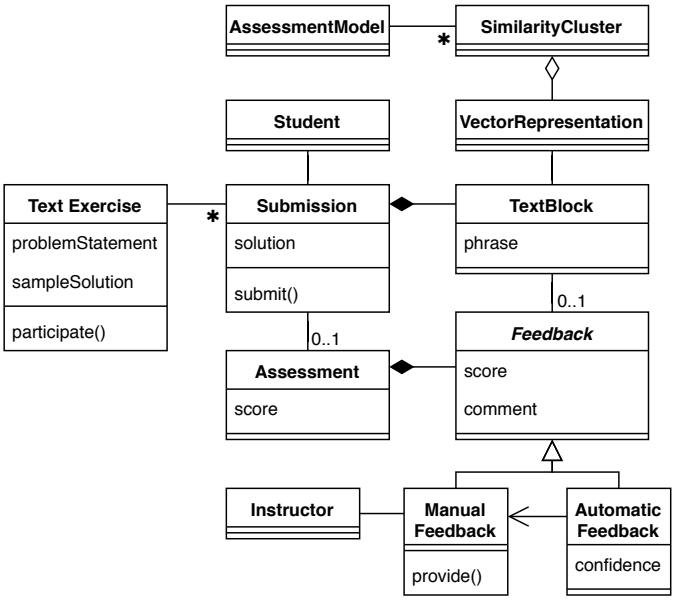

Fig. 2. The relevant entities in the system are depicted in a class diagram. A student creates a submission for a text exercise. An assessment is a composition of multiple feedback items referencing text blocks. A feedback item can be a manual or automatic feedback item. An instructor provides manual feedback. Automatic feedback items are a proxy [8] for manual feedback items. A similarity cluster aggregates the vector representations of text blocks. The assessment model consists of many similarity clusters.

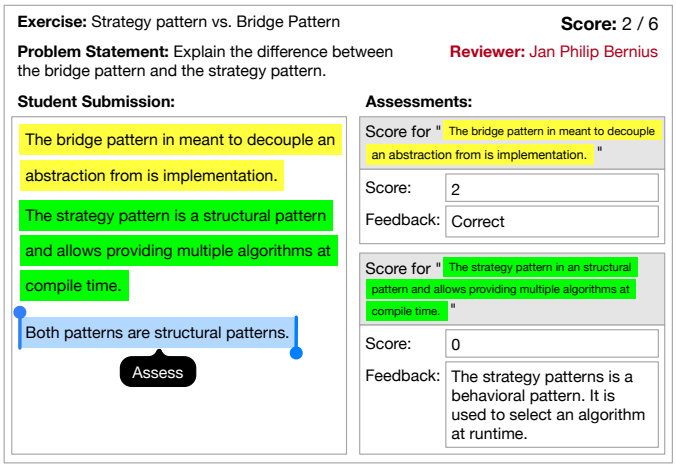

Fig. 3. Assessment of student submission for problem statement "Explain the difference between the bridge pattern and the strategy pattern." Example question taken from an EIST exam. Instructors define text blocks to build up their assessment. Each block is assessed with a score and a feedback text. The total score is based on all feedback items in the assessment.

[10, 11] and its doc2vec extension for sentences and paragraphs [12]. The algorithm can employ different strategies to calculate one-hot word vectors.

Using the resulting vector representation, we use cluster analysis to detect clusters of submission blocks [13] from all submissions of the same exercise. These clusters list the different statements submitted by all students as a part of their solutions.

Our primary assumption is that a single feedback item can be valid for text blocks from multiple submissions. Feedback for text blocks within the same similarity cluster can be applied to other nodes within the same cluster. This allows the system to provide VIRTUAL ONE-TO-ONE feedback: Real instructor feedback is applied to equivalent text blocks in a new submission automatically. ArTEMiS chooses a previously assessed text block located closely in the same similarity cluster, the nearest neighbour. The instructor feedback is selected for the new submission and ArTEMiS creates an automatic feedback item, a proxy for the manual feedback item (see Fig. 2).

If a cluster does not contain a manual feedback item, the system decides that an automatic assessment is not possible and requests a manual assessment from the instructor.

IV. EVALUATION APPROACH

We plan to conduct a case study to evaluate the automated assessment quality in the *Introduction to Software Engineering* (EIST) lecture taught at the Technical University of Munich to 1900 students. Students in the course complete weekly homework exercises. We will use the system for text exercise submissions and assessments in two stages.

As the first stage, we conduct a shadow test using our prototypical implementation. The learners submit their solution to a text question using our system. Instructors establish a truth set by assessing all submissions manually. Automatic assessment is not used during this stage. The truth set will be used for quantitative evaluation of the automatic assessment accuracy by comparing automatic assessments with the corresponding manual assessment.

> **Hypophysis 1:** Automatic assessments of text exercises following the presented concept produce results identical to manual assessments with an accuracy greater than 85%.

In a qualitative study, we will interview the instructors to analyze the block-based assessment concept (Sec. III-A), and its applicability to grading and providing feedback.

> **Hypophysis 2:** The assessment concept allows capturing all feedback necessary for assessment of text exercises. No information is lost compared to traditional assessment.

In the second stage, we will conduct a second study in a later EIST lecture to evaluate the complete automatic assessment workflow. We will evaluate how many manual assessments are needed to generate accurate assessments and the effects on assessment time.

> **Hypophysis 3:** Employing automatic assessment can save more than 50% in total required assessment time for all submissions. The assessment time per submission will increase compared to paper-based assessments.

A qualitative study with student interviews assesses the usefulness of automated feedback for them. Further, we want to understand students feeling toward automatic feedback.

V. DISCUSSION

We discuss applicability, limitations and implications of automatic text assessment. Feedback generated following the concepts introduced in this paper can only be as good as the feedback provided by the instructor. The system supports the assessment process by automating the repetitive process involved in assessing text submissions.

Grading based on automatic assessment leads to ethical problems. It is unclear whether non-native language or special figures of speech could lead to decreased scores. Applications in grading should be preceded by an extensive evaluation of assessment quality. While applications in grading are out-of-scope for this paper, we propose application in a two-phase grading process only. We intend to apply the system as a learning-support system. The generated feedback should help students during their learning progress and should not be used during a grading process.

The applicability of the described systems depends on the variety of possible solutions. Exercises with a variable answer space require more knowledge for assessment, increasing the complexity. The system focuses on assessing exercises from the lower spectrum of the revised Bloom's Taxonomy: Remember, Understand, Apply and Analyze [14]. Exercises of the given categories provide a lower variability of possible solutions and therefore limit the number of similarity clusters. Exercises from the categories Evaluate and Create are out of scope for this paper.

The design of the system allows for a hybrid assessment approach. A future system could combine manual and automatic feedback to further reduce the efforts for instructors. This could be especially useful if a certain aspect of the solution

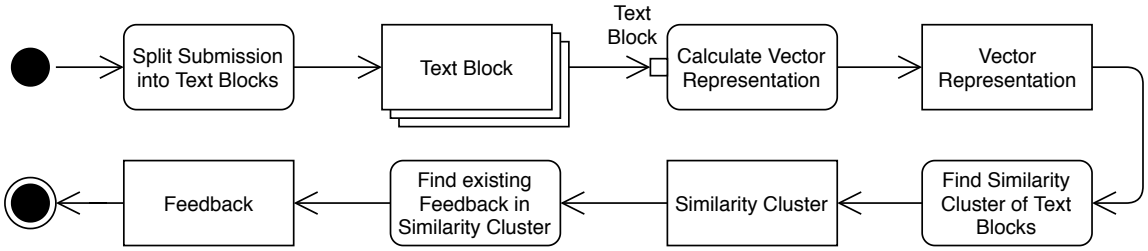

Fig. 4. The automatic assessment process. Zoomed into the "Assess automatically" activity in Fig. 1.

has a larger variability. A possible example is an exercise asking for two definitions and a comparison of the terms. The variability for the definitions is small, but the variability for the comparison part is larger. A hybrid approach allows instructors to focus the manual assessment on the comparison part, as soon as the definitions can be assessed confidently.

VI. Related Work

Kiefer and Pado suggest a system to simplify the grading process presenting responses to instructors in a sorted manner [15]. Submissions are sorted by similarity with a defined sample solution. Terms used in both the sample solution and the submission are highlighted. The tool supports instructors during the grading process but does not automatically assess submissions. The only criterion is the sample solution. Instructor assessments are not considered for the following submissions.

Wolska et al. and Basu et al. suggest a grading process where instructors grade submissions sorted by clusters of similar submissions for exercises in the domains of German as a foreign language [16] and the United States Citizenship Exam [17]. They propose clusters of entire submissions, compared to the text block based clustering approach presented in this paper. Basu et al. introduce grading of an entire cluster of submissions as a single action [17].

Gradescope Inc. offers its tool Gradescope, a commercial solution for grading assistance and "AI-assisted Grading". Their core product offers a rubric based grading system, allowing instructors to define a set of scores with feedback comments per exercise. Instructors manually select rubrics for each submission. Changes to the scores and comments in a rubric are applied to previously assessed submissions. The "AI-assisted Grading" feature creates groups of submissions (compare with similarity clusters), allowing the instructor to select rubrics for the entire group of submissions, similar to the approach of Basu et al. [17]. The automatic creation of groups is limited to multiple-choice and fill-in-the-blank exercises. It does not offer an automatic grouping of text questions.

These works focus on traditional exam assessment. The primary objective is an accelerated grading process, rather than providing feedback through comments. The focus of our approach is primarily providing more qualitative feedback to students on homework and in-class assignments.

VII. Conclusion

Assessments of text exercises require time-intensive efforts from instructors today. We argue that an automated process to generate Virtual One-To-One feedback can reduce assessment efforts for instructors and increase the amount of feedback for students. The system should use machine learning techniques to detect text blocks of the same meaning in submissions and automatically link real instructor feedback to equivalent blocks.

References

[1] S. Krusche and A. Seitz, "Increasing the Interactivity in Software Engineering MOOCs - A Case Study," in *31th Conference on Software Engineering Education and Training*, 2019.

[2] D. Kolb, *Experiential Learning: Experience As The Source Of Learning And Development*. Prentice Hall, 1984, vol. 1.

[3] S. Krusche, A. Seitz, J. Börstler, and B. Bruegge, "Interactive Learning: Increasing Student Participation through Shorter Exercise Cycles," in *19th Australasian Computing Education Conf.* ACM, 2017, pp. 17–26.

[4] T. S. Kuhn, *The Structure of Scientific Revolutions*. University of Chicago Press, 1996.

[5] P. Sadler and E. Good, "The Impact of Self- and Peer-Grading on Student Learning," *Educational Assessment*, vol. 11, no. 1, pp. 1–31, Feb. 2006.

[6] G. Jerse and M. Lokar, "Providing Better Feedback for Students Using Projekt Tomo," in *1st ISEE Workshop*, 2018, pp. 28–31.

[7] S. Krusche and A. Seitz, "ArTEMiS - An Automatic Assessment Management System for Interactive Learning," in *49th Technical Symposium on Computer Science Education*. ACM, 2018.

[8] B. Bruegge and A. Dutoit, *Object-Oriented Software Engineering Using UML, Patterns, and Java*, 3rd ed. Prentice Hall, 2009.

[9] V. J. A. Barbara E. Walvoord, *Effective Grading: A Tool for Learning and Assessment in College*, 2nd ed. Jossey-Bass, 2009.

[10] J. Mitchell and M. Lapata, "Vector-based Models of Semantic Composition," in *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2008, pp. 236–244.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *CoRR*, vol. 1301.3781, 2013.

[12] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *31st International Conference on Machine Learning*, vol. 32, 2014, pp. II–1188–II–1196.

[13] N. Bansal, A. Blum, and S. Chawla, "Correlation Clustering," *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, Jul. 2004.

[14] D. Krathwohl, "A revision of bloom's taxonomy: An overview," *Theory into Practice*, vol. 41, no. 4, pp. 212–218, 2002.

[15] C. Kiefer and U. Pado, "Freitextaufgaben in Online-Tests – Bewertung und Bewertungsunterstützung," *HMD Praxis der Wirtschaftsinformatik*, vol. 52, no. 1, pp. 96–107, Feb. 2015.

[16] M. Wolska, A. Horbach, and A. Palmer, "Computer-Assisted Scoring of Short Responses: The Efficiency of a Clustering-Based Approach in a Real-Life Task," in *Advances in Natural Language Processing*. Springer, 2014, pp. 298–310.

[17] S. Basu, C. Jacobs, and L. Vanderwende, "Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading," *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 391–402, 2013.