

Combining Statistical and Semantic Data Sources for the Improvement of Software Engineering Courses

Michael Koch, Markus Ring, Florian Otto, Dieter Landes
Coburg University of Applied Sciences and Arts

Friedrich-Streib-Straße 2
96450 Coburg, Germany

{michael.koch, markus.ring, florian.otto, dieter.landes}@hs-coburg.de

Abstract

The research project EVELIN aims at incrementally improving the quality of software engineering education. To this end, software engineering courses and their contents regularly need to be evaluated in order to provide meaningful feedback to lecturers with on the quality and its evolution. While statistical analyses of evaluation questionnaires with closed scaled questions can easily be done by most evaluation tools, open questions still need to be analyzed manually. Computer-based approaches to text analysis are still in their infancy and do not take advantage of software engineering domain knowledge. Although students' feedback is important, it is only one data source among many others such as transcribed interviews, exam statistics, or a course's entire context information. We argue that the combination of these data sources – especially natural language ones – in conjunction with innovative and semi-automated analysis techniques from, e.g., data mining and natural language processing will open up new opportunities for the improvement of software engineering courses.

Keywords

Software engineering education, multi-source analysis, natural language processing

1. Introduction

Software engineering offers a huge variety of methods and tools to design and realize complex software systems. Yet, software development also requires skilled individuals. Therefore, software engineering education and its improvement are paramount for being able to develop complex software systems successfully.

The research project EVELIN (Experimental improVEment of Learning software engINeering) tries to clarify which competencies students should possess and which didactical approaches are suitable to foster particular competencies. This includes continuous evaluation, reflection, and iterative adjustment and enhancement of software engineering courses.

Even though students' feedback – e.g. obtained by standardized questionnaires – is important for measuring educational quality, it is not the only data source to be considered. On the one hand, there are many statistical factors, like pass rate of a course, average, median, and standard deviation of exam grades, and scores of

a given exam question. On the other hand, there are additional semantic data sources such as interview transcripts of graduates and industry partners, knowledge bases, or free-text feedback reports from students. It is also necessary to observe the evaluations' context with respect to positive or negative impact, as described in [2].

We aim at creating meaningful insights into the quality and quality evolution of a course by combining currently isolated heterogeneous data sources such as the ones mentioned above. In particular, we are searching for better ways to analyze and integrate natural language data sources for the iterative improvement of software engineering courses.

2. Related Work

Data mining approaches like [8] only focus on scaled questions when trying to find correlations between students' feedback and the lecturers teaching performance. Yet, they tend to neglect qualitative data sources.

Recent approaches like [1] try to analyze text-based questions, e.g. by determining the number of positive and negative statements related to different teaching aspect categories. While the classification of positive and negative statements is done automatically, the assignment to the corresponding teaching aspect categories is done manually.

Learning management systems may offer additional statistical data about students, their demographic background, their learning behavior, and engagement [5]. Especially in combination with online quizzes and exams, very detailed analytics are possible.

3. Methodology

It seems to be insufficient to focus only on a single data source like standardized questionnaires or drop-out quotes in order to measure the quality of a given course, especially when multiple additional sources are already available.

Due to legal and privacy reasons and our focus on in-situ courses, we can neither observe individual learning process of students at this level, nor do we have access to their personal data and social background, even though many promising research attempts on the latter are under way, trying to integrate data from social media platforms in the course evaluation.

Nevertheless we have access to exam questions and overall statistical parameters on results. Still, these data are very useful for analyzing the success of a course because exams and their questions may be linked to specific teaching goals and competencies.

Questionnaires and interview transcripts of different stakeholders are around for many years and evolve slowly, but constantly over time. Questionnaires contain a variety of open and scaled ques-

tions about the lecturer, the teaching method, the course material, the workload imposed on students, and possible improvements. Interviews aim at uncovering which competencies and skills are expected from students. These data sources are anonymous but can be attributed to the same student across multiple courses or project feedbacks by using an alphanumeric key. This offers the possibility of generating trend analyses over an extended period of time while respecting the students' privacy. Since a couple of semesters, we also collect post-mortem feedback reports of students in a software engineering capstone project. These short essays contain individual experiences including personal difficulties they encountered, gains, and feelings.

While common evaluation tools support statistical evaluations of scaled questions, only a few approaches like [1] and [7] tackle the computer-based evaluation of open questions. Qualitative feedback can be transformed into quantitative feedback by analyzing the relationships between statistical data and students' written responses [9].

Generic algorithms for text mining use word statistics and word correlations to identify meaningful propositions. Due to our focus on software engineering education, better results and a broader range of applications will be likely when domain knowledge is actually incorporated as guidance into text mining. To provide this domain knowledge we need to combine the terminology of software engineering with the terminology of educational processes, forming a suitable ontology for our needs. Promising sources of domain knowledge are – among many others – SWEBOK [3], the IEEE glossary of software engineering terminology [6], as well as ontologies for aspects like teaching programming. Manual or semi-automatic reviews of lecture notes should also be considered as additional knowledge source [4].

Even though the combination of multiple data sources – especially natural language based ones – is recommended and necessary to obtain a complete picture of a course, quantitative scores taken alone should not be underestimated and contain useful information about the relationships between students' feedback and teaching performance as described in [8].

Inspired by the approach described in [1], we want to automatically identify positive and negative statements of text-based feedback answers and assign them to a suitable category in the educational context. As a basis, the predefined categories of [1] may be used, namely Course in General, Instructor, Assessment, Material, Delivery, Equipment, Program, Schedule, and Teaching. While categories are assigned manually in [1], we want to automate this step by using available domain knowledge of the specific course. At least we want to automatically suggest a category by using domain knowledge – e.g. in form of a software engineering education ontology – and provide the ability of learning from manual assignments, especially when the system was not able to automatically suggest a suitable category. These cases may be also interesting with respect to revealing new rating aspects that were neglected or overlooked before. A fine-grained hierarchical categorization system based on the mentioned ontology may support a drill-down analysis, helping to find causalities for successes or failures, when necessary.

The categorized positive and negative statements may be used to supplement, validate, or even automatically adjust the statistical information gained from closed questions.

In addition, the detection of statements in feedbacks and exams that deviate significantly from the average or median can be very useful on multiple levels of the course evaluation.

4. Summary and Future Work

Teaching software engineering is difficult and needs to be continuously improved to stay current. However, this requires careful analysis of a variety of data sources, many of which are in textual format. In this contribution, we analyzed the potential that lies in of available course evaluation data sources and outlined how these sources might be combined with each other to yield meaningful insights into the quality of a course. In particular, we sketch how data mining might help to automatize this process.

In the future we will refine and realize our concepts. We will also consider designing our approach as generic as possible to be able to support other educational domains.

5. Acknowledgements

This research is supported by Bundesministerium für Bildung und Forschung under grant no. 01PL12022A.

6. References

- [1] Abd-Elrahman, A., Andreu, M., Abbott, T. 2010. Using text data mining techniques for understanding free-style question answers in course evaluation forms. In *Research in Higher Education Journal*, 9, 12-23
- [2] Alhija, F. N.-A., Fresko, B. 2009. Student evaluation of instruction: What can be learned from students' written comments? In *Studies in Educational Evaluation*, 35(1), 37-44.
- [3] Bourque, P., Fairley, R. E. (Eds.) 2014. *Guide to the Software Engineering Body of Knowledge (SWEBOK): Version 3.0*, IEEE Computer Society Press, New York
- [4] Gantayat, N. 2010. Building domain ontologies from lecture notes. Master Thesis. Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, India
- [5] Hung, J.-L., Hsu, Y.-C., Rice, K. 2012. Integrating Data Mining in Program Evaluation of K-12 Online Education. In *Educational Technology & Society*, 13(3), 27-41
- [6] Institute of Electrical and Electronics Engineers 1990. *610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology*.
- [7] Jordan, D. W. 2011. *Re-thinking student written comments in course evaluations: Text mining unstructured data for program and institutional assessment*. Doctoral Thesis. California State University
- [8] Mardikyan, S., Badur, B. 2011. Analyzing Teaching Performance of Instructors Using Data Mining Techniques. In *Informatics in Education*, 10(2), 245-257
- [9] Sliusarenko, T., Clemmensen, L. K. H., Ersbøll, B.K. 2013. Text mining in students' course evaluations: Relationships between open-ended comments and quantitative scores. In *CSEdu 2013 - Proceedings of the 5th International Conference on Computer Supported Education*, 564-573