# Segmenting Student Answers to Textual Exercises Based on Topic Modeling


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

Anna Kovaleva
*Department of Informatics*
*Technical University of Munich*
Munich, Germany
anna.kovaleva@tum.de

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





***Abstract*—Giving feedback when grading textual exercises in very large courses is a challenge, especially when instructors want to provide consistent feedback to each student in real-time already during the lecture.**

**This paper outlines a real-time assessment approach based on topic modeling and reuse. Segmenting student answers fosters a structured form of feedback, improving the feedbacks' reusability. We present the design of an answer segmentation system, to be integrated with an assessment system for textual exercises. The resulting system aims at quicker and more consistent feedback for textual exercises and an improved learning experience for students.**


## I. Introduction

With a growing number of students enrolled at universities worldwide,[1] large courses have thousands of students participating. Large courses pose a problem for instructors when grading textual exercises. The main problem is the asynchronous assessment, which usually requires a week of time, or even longer. To reduce this delay, we teach interactive lectures where we combine theory and exercises live during the lectures, grade them immediately and provide quick feedback to students [1]. This increases student comprehension and deepens understanding.

Technology to foster interaction and discussion within large courses does exist [2, 3], as well as scalable exercise systems for programming and modeling exercises with automatic assessment [4, 5]. Textual exercises are commonly used in examination, but no automatic assessment solution is available on the market for this exercise type.

Conducting open answer questions requires time-consuming activities from instructors, including designing exercises and manual assessment, due to the high variability in student answers. To reduce efforts, instructors tend to reuse exercises from previous years. Grading is a repeatable process, instructors look for common mistakes or predefined solution patterns. The students' learning success benefits from detailed and personalized feedback [6]. To enable large scale courses, the need to reuse feedback comments arises. Individual feedback can still rely on the domain expertise of the teacher. Multiple graders require means to create consistent feedback for learners.

[1]United Nations, "UN Global Assessment on Higher Education Reveals Broad Socio-Economic, Gender Disparities," https://news.un.org/en/story/2017/04/555642-un-global-assessment-higher-education-reveals-broad-socio-economic-gender, 2017.

This paper outlines a segmentation algorithm to be applied to student answers to textual exercises. It is intended to be used as part of an assessment system for textual exercises, fostering reuse of feedback between students and increasing consistency between assessments [7].

## II. Segmenting Student Answer

We abstracted the topic modeling approach and preserve the idea that every answer is a collection of topics, and many topics are distributed among different answers [8]. We compensate for the scarcity of the words in the answers by reducing topics to keywords. Another strategy adapted from other works is "vocabulary introduction" [9]. As soon as new keywords are introduced, a new segment begins. The presented approach differs from thesaurus or ontology in a way that we do not know what the keywords are going to be, and they are calculated for every problem separately.

The algorithm can be separated into three phases: Text Preprocessing, Keyword Extraction and Segmentation. Figure 1 depicts the algorithm's flow of events, which is described in detail in the following sections. Segments can be used as a baseline for providing manual structured instructor feedback, or as a unit for assessment systems to generate feedback automatically [7].

### A. *Text Preprocessing*

Student answers are of inconsistent quality in regards to spelling, formatting and use of punctuation. Poor data quality impacts the segmentation quality negatively. Due to the nature of the system, manual preprocessing is not practical. Student submissions must not be modified, as feedback should be based on the original answer only. We correct common irregularities to an intermediate format suitable for further calculations.

Removing stop words from text is a very common way to clean textual data for Natural Language Processing (NLP) tasks [9, 10]. Words like "I", "the", "what" and "did" do not contain much lexical content and can be removed.

Lemmatization is the process of reducing a word to its meaningful root. Naturally, students use different forms of a

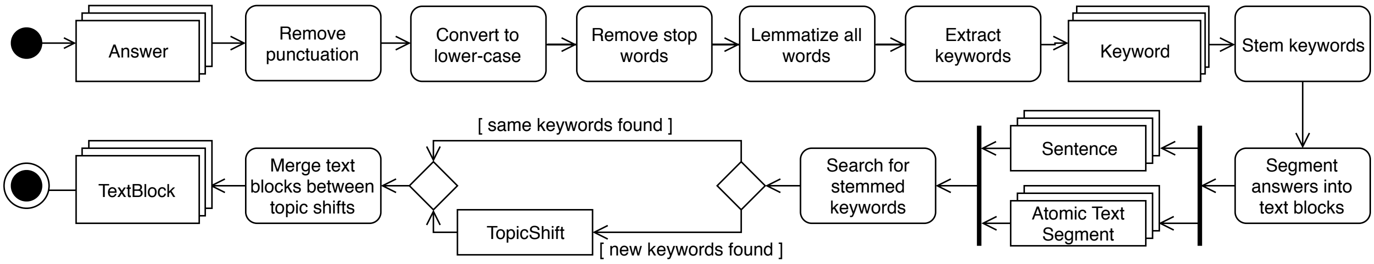


Fig. 1. The segmentation algorithms flow of events depicted using a UML activity diagram.

word: either singular or plural, different tenses, degrees of comparison, etc. The result of the text preprocessing is a set of lemmatized lower-case words without any punctuation or stop words.

### *B. Keyword Extraction*

We generalize the idea of topic modeling that claims that every student's submission is a collection of topics that are common among different answers. Compensating for data scarcity, we reduce each topic to a single keyword.

The resulting keywords are the ten most frequently used words in the texts. The number was chosen empirically based on our data.

### *C. Segmentation*

The segmentation of the texts is split up into two steps: First, the answers are split up into initial text blocks. Second, adjacent text blocks are considered and merged if there are no new keywords introduced. The result of this is a set of segments for each answer.

For identifying sentences we use a pre-trained model of the "punkt tokenizer" [11, 12] and a custom implementation for bulleted lists. To identify clauses we rely on conjunctions. This is an incomplete clause identification approach, however sufficient for this use case. We consider that subordinating conjunctions indicate a new clause, only considering sentences that are longer than 20 words to reduce false positives.

We use a stemmer to unify different forms of a word in the text. Based on lexical cohesion and vocabulary introduction [9, 13], we define segments. Within each student answer, the extracted keywords are compared for adjacent segments. A change in keywords signals a topic shift. For equal keywords, segments are merged into a single text block.

## III. Summary

In this paper we have presented a high level overview of a new algorithm based on topic modeling and text segmentation to segment student answers into topically coherent text blocks. Following a "divide & conquer" approach, we first divide student answers into initial, small segments and then merge them according to topic boundaries to larger text blocks.

The algorithm produces topically coherent segments. Segments allow for more structured assessment approaches, similar to how modeling exercises can be assessed today. This enables use of semi-automated assessment systems to be used in the assessment process, reducing the delay between exercise and feedback. Further, tools can help to keep feedback consistent between students, as comparisons can be made between segments.

The result of the algorithm's application can be improved in two areas: (1) deriving keywords and text blocks using statistical models, topic models, or decision trees. (2) Additionally, a thesaurus could be used to recognize synonyms. Future work is needed to evaluate this algorithm in a lecture setting.

## References


[1] S. Krusche, A. Seitz, J. Börstler, and B. Bruegge, "Interactive learning: Increasing student participation through shorter exercise cycles," in *19th Australasian Computing Education Conference*. ACM, 2017, pp. 17–26.

[2] J. Knobloch and E. Gigantiello, "AMATI: Another massive audience teaching instrument," in *15. Workshop für Software Engineering im Unterricht der Hochschulen*, 2017, pp. 63–68.

[3] R. Mayer, A. Stull, K. DeLeeuw, K. Almeroth, B. Bimber, D. Chun, M. Bulger, J. Campbell, A. Knight, and H. Zhang, "Clickers in college classrooms: Fostering learning with questioning methods in large lecture classes," *Contemporary Educational Psychology*, vol. 34, pp. 51–57, 2009.

[4] S. Krusche and A. Seitz, "Artemis: An automatic assessment management system for interactive learning," in *49th ACM Technical Symposium on Computer Science Education*. ACM, 2018, pp. 284–289.

[5] S. Krusche and A. Seitz, "Increasing the Interactivity in Software Engineering MOOCs - A Case Study," in *52nd Hawaii International Conference on System Sciences*, 2019, pp. 1–10.

[6] A. Poulos and M. J. Mahony, "Effectiveness of feedback: the students' perspective," *Assessment & Evaluation in Higher Education*, vol. 33, no. 2, pp. 143–154, 2008.

[7] J. P. Bernius and B. Bruegge, "Towards the Automatic Assessment of Text Exercises," in *2nd Workshop on Innovative Software Engineering Education*, 2019, pp. 19–22.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[9] M. A. Hearst, "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Computational Linguistics*, vol. 23, pp. 33–64, 1997.

[10] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Conference on Empirical Methods in Natural Language Processing*. ACL, 2003, pp. 216–223.

[11] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed. O'Reilly Media, Inc., 2009.

[12] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," *Comput. Linguist.*, vol. 32, no. 4, pp. 485–525, 2006.

[13] M. A. K. Halliday and R. Hasan., *Cohesion in English*, ser. English Language Series. London: Longman, 1976.