# TOWARDS GRADING AUTOMATION OF OPEN QUESTIONS USING MACHINE LEARNING

**A.I. Aldea**[1]
University of Twente, Industrial Engineering and Business Information Systems
Enschede, Netherlands

**S.M. Haller**
University of Twente, Industrial Engineering and Business Information Systems
Enschede, Netherlands

**M.G. Luttikhuis**
University of Twente, Centre of Expertise in Learning and Teaching
Enschede, Netherlands

***Conference Key Areas****: Engineering curriculum design, Use of professional tools*
***Keywords****: Automated grading, Machine learning, Natural language processing, Open questions*

## ABSTRACT

Assessing the academic capabilities of students should play a key role in both stimulating their learning process (formative assessment) and in the accurate evaluation of their knowledge and capabilities in relation to a topic (summative assessment). Therefore, according to the principle of constructive alignment, any form of assessment needs to be carefully designed to match the learning outcomes of a course and needs to be delivered in an appropriate format (paper-based vs. computer-based) and graded in a suitable manner. However, this is a challenging task, due to the substantial amount of time teachers need to spend on grading open questions. From our experience, this results in using less appropriate assessment methods (e.g.: Multiple Choice questions) or in less time spent by teachers on innovating their courses (e.g.: implementation of formative assessment). Inspired by recent developments in academia and practice, we propose to investigate the application of machine learning technology for supporting grading of open questions, with a focus on summative assessment and exploring possibilities for formative assessment. Our expected results include the design of a method for supporting grading of open questions with machine learning, an investigation into the most suitable machine learning algorithms for small samples of tests.

[1] *Corresponding Author*
*A.I. Aldea*
*a.i.aldea@utwente.nl*

## 1 INTRODUCTION

Assessing the academic capabilities of students should play a key role in both stimulating their learning process (formative assessment) and in the accurate evaluation of their knowledge and capabilities in relation to a topic (summative assessment). Therefore, according to the principle of constructive alignment, any form of assessment needs to be carefully designed to match the learning outcomes of a course and needs to be delivered in an appropriate format (paper-based vs. computer-based) and graded in a suitable manner. However, this is a challenging task, due to the substantial amount of time teachers need to spend on grading open questions. From our experience, this results in using less appropriate assessment methods (e.g.: Multiple Choice (MC) questions) or in less time spent by teachers on innovating their courses (e.g.: implementation of formative assessment).

There are reasons why one type is preferred than the other. Teachers use MC rather than open questions as a final assessment because it is easy to score, provides fast grading in large classes, and can fit more questions [1]. Engineering education should engage students' abilities in independent thinking, problem-solving, planning, decision-making, and group discussion [2]. MC tends to have difficulty in examining students critical-thinking skills than open questions [3] because they just have to select the correct answer from the alternatives and do not need to formulate their own answer. Using open questions can help teachers to distinguish the level of understanding for each student from the quality of the answer.

Moreover, in open question exams, students are encouraged to prepare thoroughly and study more efficiently [4] because they are expected to answer in-depth of knowledge and a wider range of thinking [1]. The open question reveals students' ability to integrate, synthesize, design, and communicate their thought [5]. The teachers can observe whether the students achieve the objective of the course or not by inspecting how the students are applying their concepts and comprehension into a real problem.

An automated grading system can assist the teacher by reducing the grading time and enhance the learning process. Spending less time on grading enables the teachers to deliver faster feedback so that both the teacher and the students can discover which aspects the students need to improve. Several studies have been conducted in the field of Machine Learning (ML) and more specifically in Natural Language Processing (NLP) which apply to this context. On the other side, most of the studies explore how to assess short-answer questions, which requires an answer of one phrase to one paragraph and maximally 100 words [6], or an essay. However, the types of methods used in these studies are not suitable for assessing answers to open questions, which can be longer than a paragraph, but shorter than an essay. Furthermore, the analysis that is done for essays is not suitable for understanding the content of the answers since it is focused on sentence and essay structure (e.g.: word choice or grammar usage, and organization [7] rather than analysing the meaning of the answer. The main goal and contribution of this paper is to provide an alternative to existing methods based on supervised learning algorithms. We

consider that such a method would contribute to advancing the state-of-the-art on automated grading for open questions, which is an area of application for ML that is rather lacking by comparison. Furthermore, we consider that this method would help teachers by providing them with a tool which can be used to improve the process of providing feedback and grades for summative and formative assessment. Additionally, by grouping similar answers, any bias that teachers might have would be minimised as similar answers would receive the same grade.

## 2 METHODOLOGY

To guide the design of this research we use the CRISP-DM (CRoss-InduStry Process for Data Mining), a popular 6-step methodology used in the field of data science. We apply these 6 steps to our research as follows.

### 2.1 Business Understanding

The first step of the methodology refers to understanding the objectives and requirements of the organisation. In this research, several interviews were conducted with teachers from a Dutch university to identify the main requirements:

- Suitable for a smaller sample of exams (50-100 students) as most solutions are designed for large samples (500-1000 students)
- Without the need for grading a sample of exams (70-100) to train the algorithm since the exams have different questions every year
- Have control of results and transparency of the process
- Suitable for open questions
- Able to group the results based on the similarity between student answers

Based on this, the main research question was formulated: **How to design a method for assessing a small sample of open question exams with the help of ML algorithms?**

### 2.2 Data Understanding

This step refers to the collection and exploration of the necessary data. Our research is based on a data set which consists of 3480 student answers to 26 different questions. Each answer has information about the maximum score the student can achieve and the actual score they received (manually graded by teachers). The types of questions range from knowledge and understanding to application and analysis – according to Bloom's taxonomy [8].

### 2.3 Data Preparation

In this step, the emphasis is on deciding which data to include in the research and how to properly prepare this data for analysis. In our research, the most crucial step is data cleaning and preparation. Thus, to increase the performance of the algorithms the following preprocessing steps were followed:

- Removing stop words (e.g.: a, and, but, how, or, what, etc.)
- Expanding contractions (e.g.: can't – can not, shouldn't – should not, etc.)
- Removing special characters and punctuation (e.g.: !=§()%)

- Combining words with hyphens (e.g.: infor-mation – information)
- Lemmatizing the words to their base form (e.g.: went – go; going – go)
- Creating n-grams of maximum n = 3 – considering combinations of up to 3 words (e.g.: “enterprise architecture framework”)

### 2.4 Modelling

This step refers to making a selection of methods and algorithms which should be used for the data analysis. In the case of this research, based on the requirements mentioned in Section 2.1, one of the main requirements of teachers is to not have to train an algorithm by grading a sample of exams. The main reason for this is that the sample is not large enough for using such supervised learning algorithms. Thus, for this research, we have chosen to explore the possibility of using unsupervised algorithms. More specifically, clustering-based algorithms, such as k-means, hierarchical clustering and spectral clustering. Additionally, for feature extraction, we chose to use Count Vectorization (dictionary with all the unique words our documents contain), Term Frequency — Inverse Document Frequency Vectorizer (TF-IDF - vector representation of textual data), and Word Embedding (semantic relationships between words by having “similar” vectors for similar words).

### 2.5 Evaluation

In this step, the goal is to assess the performance of the chosen model(s) in relation to the business requirements. In this research, we discuss the results of applying the methods and algorithms described in Section 2.4, from the perspective of the requirements defined in Section 2.1.

### 2.6 Deployment

In the last step of the CRISP-DM, the plan for implementing the results of the research should be detailed. Thus, we explain the limitations and recommendations for future work based on the results we have obtained in this research.

## 3 RESULTS

In this chapter, we conduct an analysis of our dataset and discuss the results we have obtained by using different ML methods.

### 3.1 Clustering with K-means without a reference answer

First, we analyzed the distribution since we want the scores of the answers to have a reliable clustering. The goal is to distinguish between good and bad answers; therefore we need both in a reliable quantity. After choosing a suitable question we vectorized the preprocessed student answer with TF-IDF: for the parameter we chose min_df=.0025, max_df=.1, ngram_range=(1,3). Because it seems to be a good configuration for this task and we want to include n-grams. After that, we applied the k-means cluster implementation from the Python Scikit-Learn library. For the number of clusters, we started with a high number (n=10) because initially there were 10 different scores possible. We applied this further and with each iteration, we

decreased the number of clusters until n=4. Less than 4 would not be reasonable and therefore we defined it as the lower limit of clusters.

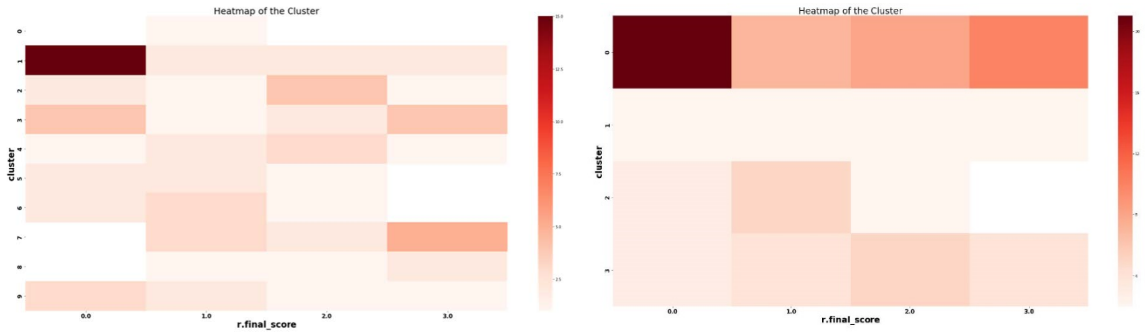



*Fig. 1. Heatmap and Cluster Details for n=10 (left) and n=4 (right)*

The results in Fig. 1 indicate that k-means in this configuration doesn't seem to be beneficial. There is no pattern observable and all scores are distributed over the different clusters. Additionally, regardless of the number of clusters, there is one cluster which contains a large number of answers (depicted in dark red in Fig. 1), while the rest contain much fewer answers. Therefore, this approach doesn't seem promising. One reason could be that it lacks in terms of semantics due to using a vectorizer based on the Bag-of-words approach. Therefore, we decided to experiment with several text similarity methods which could address this issue.

## 3.2 Similarity Analysis

The common approach for finding similar documents or sentences is based on counting the number of occurrences of similar words between sentences. This approach, however, fails to consider the fact that while the number of common words in documents increase the topics can be completely different. Here the cosine similarity helps to fix this flaw. Rather than just counting the words and calculating the Euclidean Distance, the cosine similarity calculates the angle between two vectors in a multi-dimensional space. To facilitate automated grading, not only the similarity between the student answers is important, but also the similarity between a student answer to a reference answer (provided by the teacher) is important. The reason behind this is that we expect to provide a grading system when analyzing how similar the student answer is to the expected reference answer from the professor. Based on the literature review, in the next sub-sections, we present the most promising and well-performing metrics to analyse the similarity between text.

### *3.2.1 Cosine Similarity with TF-IDF Vectorizer with Reference Answer*

In a first step, we implemented a basic similarity analysis. The cosine similarity was simply based on a comparison between the reference answer and each student answer. Each of the answers was vectorized and the cosine similarity was calculated for each of the answers. Based on the given data we calculated the cosine similarity with the function linear_kernal and used the tfidf-vectorizer. The performance is shown in Fig. 2. As can be seen, there seems to be a correlation between the

cosine similarity and the final score. We did this analysis with more than one question and the results show a similar relationship.

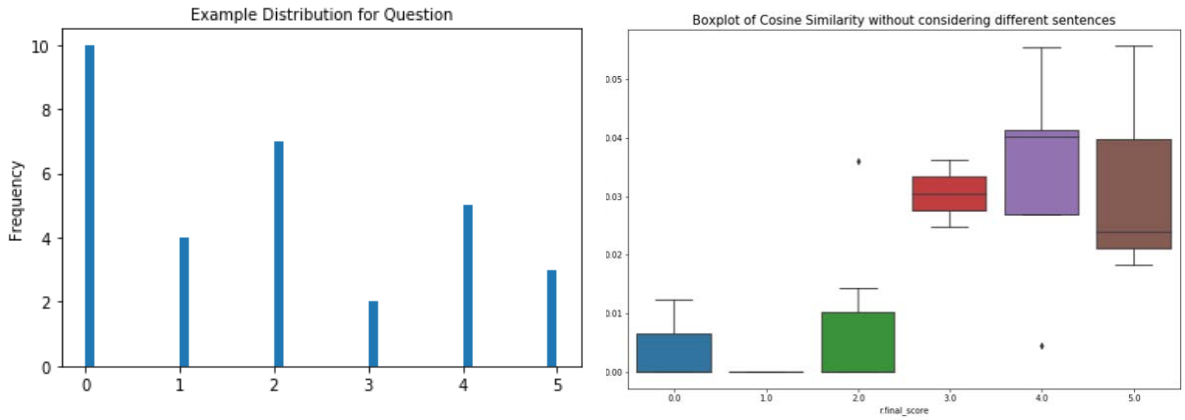
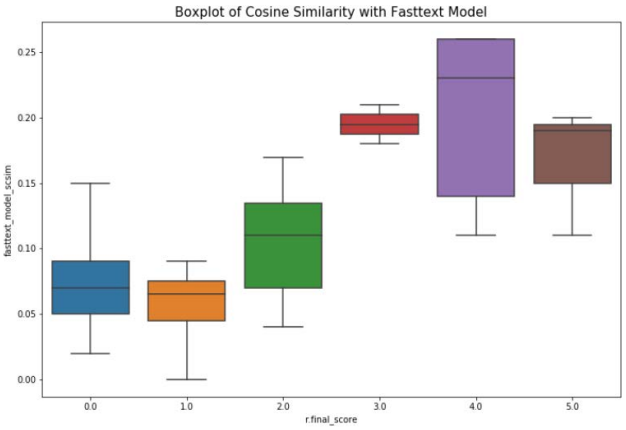



*Fig. 2. Distribution of the actual scores (left) and Boxplot analysis of TF-IDF vectorizer (right)*

This indicates that there is a distinguishable separation between the good (≥3) and the bad (<3) scores. Thus, this can also be used as a way of clustering the answers according to their similarity to the reference answer into 2 groups. However, if we want to further cluster the data, we have to choose a different approach.

### *3.2.2 Cosine Similarity with Word Embeddings with Reference Answer*

Word embedding is important for the similarity measure of soft cosine similarity because student answers can address the same topic in different ways (by using different words, etc). Therefore, it can be advantageous to consider the semantic meaning of an answer as well as word similarities. The soft cosine similarity treats words with similar meaning alike by redirecting the word vectors. For getting the word vectors one needs an embedding model which can be trained on different data sets. In our research, we have used the Fasttext model which seems to be able to distinguish between good and bad answers quite reliably.




| r.final_score | mean_cosine_similarity | number_answers |
|---|---|---|
| 0.0 | 0.075556 | 9 |
| 1.0 | 0.055000 | 4 |
| 2.0 | 0.104286 | 7 |
| 3.0 | 0.195000 | 2 |
| 4.0 | 0.200000 | 5 |
| 5.0 | 0.166667 | 3 |

*Fig. 3. Analysis of the Fasttext Embedding*

The analysis in Fig. 3 shows that in general bad answers have a small similarity with the reference answer, whereas high score answers (4 and 5) have a large similarity, with some exceptions. Therefore, the result indicates that a binary cluster into bad

answers (0-3 points) and good answers (3-5 points) could be beneficial. However, we still have a certain level of inaccuracy because of outliers.

### *3.2.3 Cosine Similarity with Word Embeddings without Reference Answers*

The literature we investigated indicated that hierarchical clustering could be beneficial since it is a bottom-up approach. The goal is to cluster those answers which are most similar. This is done pairwise and after the two most similar are obtained, the next closest answer to the previous answers is found. This is done until all the answers are grouped to one big cluster.
Since we obtained good results using word embeddings we continued using the Fasttext model for this task as well. We implemented the Agglomerative Clustering class from the sklearn.cluster Python library. The number of clusters is initially set to 10 while the parameter affinity is set to "cosine" (distance between the data points). Finally, the linkage parameter is set to "complete", which uses the maximum distances between all observations of the two sets. These default parameters are chosen using a manual parameter search. However, a more expressive study might be beneficial to optimize cluster performance. As part of this research, we want to identify a preferable number of clusters per question. We achieve this by simply decreasing the default number until 4 clusters are reached. This is chosen because fewer clusters seem to have diminishing performance.

| agglo_clustering | final_scores_mean | final_scores_list |
|---|---|---|
| 0 | 1.000000 | (0, [1.0, 1.0, 1.0]) |
| 1 | 1.750000 | (1, [2.0, 2.0, 2.0, 1.0]) |
| 2 | 3.800000 | (2, [3.0, 3.0, 3.0, 5.0, 5.0]) |
| 3 | 2.666667 | (3, [3.0, 2.0, 3.0]) |
| 4 | 5.000000 | (4, [5.0, 5.0, 5.0]) |
| 5 | 2.333333 | (5, [3.0, 2.0, 2.0]) |
| 6 | 2.000000 | (6, [2.0, 2.0]) |
| 7 | 1.666667 | (7, [2.0, 2.0, 1.0]) |
| 8 | 4.200000 | (8, [5.0, 5.0, 5.0, 3.0, 3.0]) |
| 9 | 1.000000 | (9, [1.0]) |

| agglo_clustering | final_scores_mean | final_scores_list |
|---|---|---|
| 0 | 1.307692 | (0, [1.0, 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 2.0, 1.0, 2.0, 1.0]) |
| 1 | 1.600000 | (1, [1.0, 2.0, 2.0, 2.0, 1.0]) |
| 2 | 2.250000 | (2, [1.0, 3.0, 2.0, 3.0]) |
| 3 | 3.000000 | (3, [3.0, 3.0]) |
| 4 | 1.666667 | (4, [1.0, 2.0, 2.0]) |
| 5 | 2.333333 | (5, [3.0, 1.0, 3.0]) |

*Fig. 4. Hierarchical Cluster analysis with 10 clusters (left) and 6 clusters (right)*

It seems that a high number of clusters is good when the answers are quite diverse in terms of content. Furthermore, by decreasing the number of clusters the algorithm still captures the differences well. This can be seen that the contained answers in the cluster are most of the time either 3 and 5 or 1 and 2.To understand the clustering behaviour we have analysed the student answers in greater detail (based on the results with 6 clusters – right side of Fig. 4). As can be seen in Fig. 5, all the answers in Cluster 2 are using similar wordings (e.g. Volvo Car, innovation, orchestration). However, the meaning of the sentences differ which results in different scores (answer 3 - 0.9/3.0, answer 16 & 23 - 3.0/3.0, answer 17 - 1.5/3.0).

```
3     The Volvo Car innovation requires a good focus on as much feedback as possible.
16    For the Volvo case, the Orchestration layer is the most appropriate to manage innovation and new product development
among external innovatiors.
17    Most appropiate one for Volvo Car is the fact that innovation agencies are more distributed with them being formed by
people with different backgrounds with no head person commanding.
23    The most appropiate logics to deal with the digitation of innovation for Volvo would be Orchestration.
```

*Fig. 5. Detailed Analysis of Cluster 2*

A similar situation can be observed in Cluster 5 (Fig. 6) where most of the words contained in the answers are similar, with some exceptions. From these results, we can conclude that the words which are common between the answers have a high impact on how the results are clustered together. Therefore, in certain situations, the algorithm can cluster together answers which do not belong together semantically.

```
6     Orchestration is the most appropriate for Volvo Car.
19    For Volvo Car it is most appropriate to think out of the box
24    Orchestration is the most appropriate for Volvo car
```

*Fig. 6. Detailed Analysis of Cluster 5*

We performed the same analysis by using Spectral clustering, and the results were fairly similar. The main difference is in how the method itself does the clustering. Namely, spectral clustering aims to divide data points into groups, where each point in a group is similar to other points in the group and dissimilar from points in other groups. The intention behind this approach is to try to minimize the distance between data points inside a cluster while “maximizing” the differences between data points of another cluster. This approach is useful when the data has special properties.

## 4 CONCLUSION AND DISCUSSION

In this paper, we aimed to propose a method for assessing open question exams with the help of ML which can be used for a small sample. To demonstrate how we achieved this goal, we refer to the requirements stated in Section 2.1.

To have a solution that is suitable for a smaller sample of exams (50-100 students) and to remove the need for grading exams to train a ML algorithm, we have chosen to explore several unsupervised learning methods (K-means, Hierarchical Clustering and Spectral Clustering). From these three methods, the last two seem to outperform K-means. Additionally, we have tested whether clustering based on similarity to other student answers or the reference answer has an impact on the results. In general, it seems that looking only at the similarity between student answers can yield similar if not better results. Thus, if the method doesn’t need to provide actual grades, but rather just a grouping of student answers that the teacher can manually grade, then not using a reference answer is beneficial. Additionally, from our interviews, it resulted that in many cases, teachers cannot formulate a reference answer which contains all the possible correct answers. This is especially the case at the master level where the questions are predominantly on application and analysis. Thus, in this situation, using reference answers would not be feasible.

Another requirement from teachers was to have control of the results and transparency of the process, while the answers should be grouped based on similarity. With our proposed method, the teachers would be able to see which

answers were grouped and can determine if their similarity is sufficient to provide the same grade and feedback. If this is not the case, then adjustments to the clusters can be made. Further details about this are available in the next section.

### 4.1 Limitations and Future Work

One of the main limitations of using a clustering-based method is that it cannot be fully automated. Human input will always be needed to do the grading. However, this seems to be in line with the expectations of teachers which mentioned that they would not easily trust a solution which does not require their intervention. Second, while the clustering-based methods we used showed that we can provide reasonable performance when grouping similar answers, there are still situations in which answers are included just because they share common words, while semantically they don't belong in that cluster. One solution for future work would be to remove the words that belong to the question (e.g.: Volvo car, innovation) from the answer since students tend to use these words in their answers to be more related to the question. Thus, the same preprocessing steps applied to the answers can be applied to the question and the resulting words can be used as an exclusion filter for the answers. Third, our research has included a relatively small sample of questions and answers. To further test the viability of this method, future work should include a larger sample from several exams, from both the master and the bachelor levels. This will help determine whether certain question types are more suitable for this method and whether certain topics perform better with particular ML algorithms. Additionally, future work can experiment with standardising open questions formats to make them more structured and perhaps more suitable for ML algorithms. Fourth, we have used only a small selection of available methods, algorithms and features with a limited variety of parameter adjustments. In further research, we advise using methods such as Grid search for hyperparameter tuning, and to include more features, such as Word count, Word length average, Token-based similarity, Sequence-based similarity, etc. This would help improve the performance of the algorithms since it improves their capability to capture the semantic meaning of student answers. Additionally, more advanced word embedding models such as Google's BERT and OpenAI's GPT-2 should be tested since they might be more suitable than Fasttext. Alternatively, a custom word embedding model, based on BERT and GPT-2 can be developed to include more domain-specific terminology.

The results of this research are currently used to develop a web-based tool which will include a user-friendly interface. This will allow the teachers to run the analysis of student answers based on the proposed methods, adjust the clusters to make them more uniform, and to provide grades and feedback to a whole cluster of student answers. We intend to test this prototype by using exams from multiple courses and report on the results in a future paper. The teachers involved in these courses will be asked to compare the results of using the tool to manual grading. This would allow us to assess the performance of the tool and make decisions about improving the underlying algorithms. A future goal for this prototype is to use it for supporting teachers with formative assessments by reducing the time spent on this task.

## REFERENCES


[1] Stanger-Hall, K. F. (2012), Multiple-choice exams: An obstacle for higher-level thinking in introductory science classes, *CBE Life Sciences Education,* Article vol. 11, no. 3, pp. 294-306, doi: 10.1187/cbe.11-11-0100.

[2] Wang, H. C., Chang, C. Y. and Li, T. Y. (2008), Assessing creative problem-solving with automated text grading, *Computers and Education,* Article vol. 51, no. 4, pp. 1450-1466, doi: 10.1016/j.compedu.2008.01.006.

[3] Funk, S. C. and Dickson, K. L. (2011), Multiple-Choice and Short-Answer Exam Performance in a College Classroom, *Teaching of Psychology,* Review vol. 38, no. 4, pp. 273-277, doi: 10.1177/0098628311421329.

[4] Pinckard, R. N., McMahan, C. A., Prihoda, T. J., Littlefield, J. H. and Jones, A. C. (2009), Short-answer examinations improve student performance in an oral and maxillofacial pathology course, *Journal of Dental Education,* Article vol. 73, no. 8, pp. 950-961.

[5] Roy, S., Narahari, Y. and Deshmukh, O. D. (2015), A perspective on computer-assisted assessment techniques for short free-text answers, *Communications in Computer and Information Science*, vol. 571, pp. 96-109.

[6] Burrows, S., Gurevych, I. and Stein, B. (2015), The eras and trends of automatic short answer grading, *International Journal of Artificial Intelligence in Education,* vol. 25, no. 1, pp. 60-117, doi: 10.1007/s40593-014-0026-8.

[7] Zupanc, K. and Bosnić, Z. (2015), Advances in the field of automated essay evaluation, *Informatica (Slovenia),* vol. 39, no. 4, pp. 383-395.

[8] Anderson, L. W. and Krathwohl, D. R. (2009), A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives, Longman, New York.