

## Enhanced Peer Assessment in MOOC Evaluation Through Assignment and Review Analysis

<https://doi.org/10.3991/ijet.v13i01.7461>

Ramón Alcarria<sup>(✉)</sup>, Borja Bordel, Diego Martín de Andrés, Tomás Robles  
Universidad Politécnica de Madrid, Spain  
ramon.alcarria@upm.es

**Abstract**—The rapid evolution of MOOCs in recent years has produced a change in the education of students and in the development of professional skills. There is an increasing pressure on universities to establish procedures for the recognition and certification of student participation in MOOCs. In order to guarantee that the evaluation procedures are in line with the quality of the procedures traditionally established in the university, a proposal for an enhanced peer assessment is required to allow a more precise review of the students' tasks and the assessments provided by his colleagues, considering procedures of verification of originality and a complete rubric for the peer review that takes into account reviewer's history for a correct grade calibration. This paper describes the implementation of the evaluation tool, and an experimental validation that indicates that the majority of the students who have used the tool for the revision of assignments have generated grades closer to the revisions generated by the professors in the study.

**Keywords**—P2P activity, peer assessment, semi-automatic evaluation, MOOC, originality check, self-plagiarism, Learning Tools Interoperability

### 1 Introduction

Massive Online Open Courses (MOOCs) have proliferated a lot in recent years. Proof of this is the large number of platforms available for collaboration between participants and the management of evaluations by the professor. Most important MOOC platforms incorporate certification procedures. Certificates usually contain the name of the University and the signature of the professors who teach the course, to give the MOOC an academic relevance. In this way academic recognition takes place beyond virtual badges [1].

The main difficulty encountered by universities in this new form of certification lies in the lack of quality control of student learning assessments [2]. The fulfillment of learning objectives at high levels of comprehension (detailed in Bloom's taxonomy as learning, application, creation, etc.) presents a complex evaluation for environments with high number of students, where the evaluation tools are in general collaborative

For the provision of exercises on free text fields, which require an effort of creativity of the student, two particular problems occur. As the proposed task has a non-unique solution, the correction of the exercise raises a certain subjectivity on the peer's side, who must have a rubric to perform the correction. The evaluation of the work and the corrections are conditioned by the profile of the corrector, who, being not a professional, can be demanding or benevolent or provide useless feedback to other students [3].

In addition, creative creations and compositions are subject to plagiarism, which frequently occurs given the limited experience of reviewers as proofreaders (as opposed to corrections made by the professor). The lack of automated mechanisms to detect the plagiarism of assignments from other peers or resources on the Internet often prevents the student from reaching the expected level of learning.

Thus, the objective of this work is the proposal of an evaluation framework for enhanced peer assessment in MOOCs, considering assignment and review analysis, enabling students to check the originality of their assignments and their reviews and facilitating professors the detection of low-level feedback or biased reviews.

The rest of the paper is organized as follows: Section 2 describes related work in modern evaluation methods in MOOCs, Section 3 includes the proposed evaluation framework and its implementation. Section 4 presents an experimental validation and result discussion. Finally, Section 5 provides some conclusions and future work.

## 2 Related work

This section describes the approaches that explore modern evaluation methods in MOOCs, focusing on those which allow the prevention of biased corrections in academic writings, plagiarism and low-level feedback detection.

A new approach in the curriculum offered by the MOOCs considers the concept of "specialization" [4] as a collection of online courses around a particular subject that allows students to achieve a set of transversal competencies resulting from synergies between courses, hardly achievable through a single educational product.

This training offer often comes from the adaptation of defined subjects with traditional teaching methods, such as master classes supported by slides. This has produced that the content and type of evaluation designed for MOOCs contains mostly video-lessons, small exercises or quizzes [5]. Some authors consider that more learning strategies should be introduced from the field of distance learning [6]. In this field, some authors provide a comparative study to assess e-learning platforms from the distance learning's point of view [1]. Some authors maintain that the use of the technologies allows automatic feedback through feedback generators [7], while others consider that evaluation activities should be analyzed from the Bloom's taxonomy [8], maintaining that evaluations based only on quizzes do not reach deep levels of learning [9], since these types of evaluation strategies only reach the levels of *recall* and *comprehension*, from the point of view of conceptual knowledge. These authors propose a greater participation of the students, through the contribution of educational content [10] (participation in forums or spaces of debate) or evaluation through P2P

techniques, allowing students to develop activities in higher levels of Bloom's taxonomy [9] (such as *creation* or *application*), requiring the feedback phase to provide a deeper insight into metacognitive learning [11].

Evaluations of student behavior in performing P2P tasks describe attitudes to be corrected, such as plagiarism [12], sometimes non-consciously, or poor quality feedback, attributed to a lack of training in evaluation [13] [14]. Some studies consider that peer grades are higher [15], while others do not locate friendship bias in peer assessment [16] but consider a peer assessment rubric to be essential for a more accurate evaluation [17]. All these works are centered on the academic research line and require practical solutions that support improved assessment techniques in learning environments with a high number of students, such as MOOCs.

Our work tries to cover the deficiencies detected in the mentioned studies, proposing an evaluation framework for assignment and review analysis, preventing from plagiarism and facilitating the detection of low-level feedback or biased reviews.

### 3 Enhanced peer assessment in MOOC evaluation

In this section we describe the requirements of the evaluation framework and its design, implementation and integration to other learning tools.

#### 3.1 Problems of peer feedback

We have detected a number of problems in peer feedback that must be taken into account as requirements of our evaluation environment.

There is little training of students in competencies related to work evaluation and the provision of constructive feedback. We include in this category the students who overwork in their revision, providing excessive feedback, mixing it with other questions or with their personal experience in doing the same work and also the so-called "grammar police", usually native students who focus their comments on questions related to the use of English, and students who provide minimal feedback by only accepting or rejecting the assignment, or providing destructive feedback. To address these problems, we establish as a requirement **REQ#1** the need for a conformance review of the submissions and their subsequent revisions, so that they comply with the publication rules established in the course. In order to properly detect the biased reviews and those profiles that are too strict or benevolent with their revisions, we propose to solve **REQ#2**: detection of reviewer's opinion deviation regarding the general opinion.

In the case of platforms that offer tools for the detection of plagiarism, such as Coursera, there is a misinterpretation of the values of plagiarism, detecting accusations without exploring which are the similar references. In many cases the detected fragments belong to the same author, which is known as self-plagiarism, which, for exercises whose response does not have to be original, and assuming granted the corresponding publication rights, the use of these contents for the delivery of exercises could be allowed. The contribution to the resolution of these problems leads us to

the proposal of the requirement **REQ#3**: Accurate detection of plagiarism and self-plagiarism.

There is an absence of tools for feedback evaluation. Feedback evaluation is an important aspect that allows us to know the knowledge and competences of the students related to the evaluation of works. We propose a rating of the feedback following the principles of the social network (likes, dislikes) through **REQ#4**: consider the feedback ratings for the detection of a possible bias of the reviewer.

Rigid revision procedure: The usual platforms do not allow to have several rounds of feedback, as it happens in review platforms for scientific articles, that would allow ask for clarification, gauge student feedback based on student profiles, particular strength, weaknesses, etc. This rigid procedure does not allow a bidirectional contact between student and reviewer, and offers a little sense of reciprocity that causes the student not to become involved in the community promoted by the course, so the term *peer assessment* loses relevance. We propose as **REQ#5** a configurable review process in the number of reviews by submission and the number of review rounds, with anonymous communication between reviewer and reviewed.

### 3.2 Evaluation framework design

Considering the characteristics gathered in the state of the art and the requirements described in the previous section, we propose the design of an evaluation environment enabling enhanced peer assessment in MOOCs. The environment presents an architecture described in Figure 1.

The architecture presents the three roles that we consider in the evaluation process: student, reviewer (usually a student who has to perform a review task) and professor, who we consider as administrator of the course within the framework, although the roles of professor and administrator can also be separated.

The evaluation framework is connected to a content provision platform that collects user submissions. The core of the evaluation framework is the MOOC evaluation workflow engine, which is responsible for the review process logic. In this process, the reviewers access the submissions through the revision panel, and send their revisions as input to the processes of similarity checking and conformance checking.

The conformance checker detects inadequate submissions (contributing to **REQ#1**), considering empty documents, short revisions, incorrect file formats or corrupted documents. If the conformance requirements are not fulfilled, the user will be prompted to resend the assignment.

The similarity checker checks if there is a probability of plagiarism of the document and of the revision and gives a result that indicates the level of copy that has occurred (therefore contributing to **REQ#3**). This similarity checker should compare the work against other works previously delivered by the students in this edition of the MOOC or in previous editions. It should also contrast the contributions of students against other documents on the Internet.

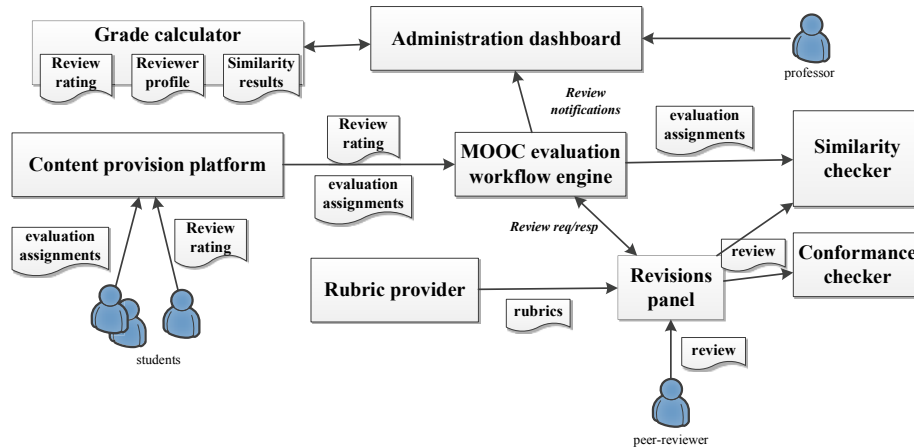


Fig. 1. Evaluation framework architecture

Once the review is done, the set of revisions are published anonymously so that students can do a review rating (as *REQ#4* states). In this way a new measure is obtained that can be used to detect biased reviews and those profiles that are too strict or benevolent with their reviews.

Finally, the review rating, reviewer profile and similarity results are used by the grade calculator module to generate a rating. The administration dashboard allows the professor/course administrator to set the parameters for an optimal calculation of grades, as well as initiate review processes, manually assign revisions or generate more feedback rounds if necessary, complying with *REQ#5*.

### 3.3 Evaluation process illustrated

In this section the developed evaluation process is described to be able to better illustrate the operation of all implemented components.

**Step #1:** Before the start of the submission period of the P2P task, the professor should define the evaluation methodology in the administrator dashboard. One of the necessary configurations is the choice of values to weight the reviewer's opinion deviation from other reviewers' and also to weight revisions according to the students' rating. He/she must also decide on configuration options related to the evaluation methodology, such as the number of review iterations, if the reviews are anonymous, and whether the review ratings process should occur. The proposed methodology highly influences the improvement in student learning. Thus, it must be very carefully designed. All this information is stored in a database.

**Step #2:** Students submit their assignments into the content provision platform through the MOOC platform web interface (it can be a Moodle, Sakai, Open edX LMS platform, etc.). This submission generates a web service request (usually a HTTP POST operation) containing the student identity (typically his name, ID number, email and home country) and the submitted assignment.

**Step #3:** The assignment is sent to be evaluated in the revision panel. In order to protect the student's identity, this operation is performed by means of a unique transaction number, so student's personal information remains hidden in the database. The assignment goes through the similarity check step. The similarity score is offered to the reviewer in the revision panel.

**Step #4:** Results of the formal evaluation are sent to the conformance checker for detection of some inconsistencies (detection of corrupted files, empty documents, etc.) which may produce a revision resubmission. Also, the provided revision goes through the similarity check step. The similarity scores (both of the assignment and revision) are returned to the workflow engine to be stored in the database.

**Step #5:** Reviews are sent to the content provision platform, which lists all reviews so that users can rate them. Reviews rating are then sent to the MOOC evaluation engine to be stored in the database. Reviewers can see their review ratings in order to check if their work corresponds to other student expectations and check if they correctly interpreted the provided rubric.

### 3.4 Scoring algorithm and calculation process design

This section proposes a scoring algorithm that takes into account the user's grading history in the platform, the appreciation of the other users of the revisions made and allows professors to configure a series of parameters for the calculation of the final grade.

We consider an assignment  $S_j$  consisting of a set of  $m$  revision requests  $S_j \equiv \{r_{1n} \dots r_{mn}\}$ , each made by a reviewer. The reviewer  $m$  has so far performed  $n$  revisions, expressed by this set  $R_m \equiv \{r_{m1} \dots r_{mn}\}$ , considering the latter  $r_{mn}$  the revision made or to be made for the assignment  $S_j$ .

We define the average dispersion of a reviewer  $m$  as  $d_m$  according to equation (1):

$$d_m \equiv \frac{\sum_{j=1}^n r_{mj} - \bar{s}_j}{n} \quad (1)$$

That is, the mean of the differences between  $r_{mj}$ , the revision made for the assignment  $S_j$  and  $\bar{s}_j$ , the average score of the assignment  $S_j$  according to all the revisions obtained. This average dispersion allows us to know the deviation of the opinion of the reviewer with respect to the general opinion, and therefore to know if it is a strict or benevolent reviewer, and to take it into account later to weigh its review against the review of the other reviewers (thus, complying with **REQ#2**).

We consider the revision  $r \equiv \{0, 1, 2, 3, 4, 5\}$  that can take values from 0 to 5. We define the rectified grading  $rg$  as the difference between the rating of each reviewer and their average dispersion. We used the quantifier parameter of the mean dispersion  $p$  to give the average dispersion a configurable weight with the professor. By default, we have chosen this value of 0.5.

Finally, we defined a rating mechanism for revisions to comply with **REQ#4**. This mechanism is based on the publication of revisions anonymously and on the provision

of a mechanism so that other students can label revisions with "likes" or "dislikes", following the model of well-known social networks.

For a revision  $r_{1n} \in S_j$ , we calculate the coefficient of appreciation  $a_{1n}$  as  $a_{1n} = q (\#likes - \#dislikes) + 1$ , where  $q$  is a quantifier of appreciation that weighs the revisions according to the students' valuation, by default we have associated  $q = 0.1$ . We limit the coefficient of appreciation between a range to saturate when there is a large number of likes and dislikes. The selected range is [1-2]. Finally we define  $Q_{S_j}$  as the final score of  $S_j$ , by equation (2):

$$Q_{S_j} = \frac{\sum_{i=1}^m a_{1n} * r_{gin}}{\sum_{i=1}^m a_{1n}} \quad (2)$$

With  $Q_{S_j}$  we normalize the weighted values through the quantifier of appreciation.

As can be seen, there are certain parameters that are configurable by the course administrator and that regulate the effects of revision history, reviewers, and student ratings of anonymized revisions. In the next section we describe the interface implemented to offer this customization environment to the course administrator.

### 3.5 Implementation and interoperability with other learning tools

We implemented our evaluation framework as a web application. Server part consist of the evaluation workflow engine, considered the core component in the architecture, rubric provider, grade calculator and conformance checker. These modules are implemented in Node.js, for http libraries and core functionalities, Express.js for routing, sessions, front-end serving, and Sails.js for Model-View controller (MVC) pattern and DB connection.

The database is NoSQL, implemented in MongoDB, for easy integration with server's back-end. It stores evaluation assignment metadata, performed reviews and review ratings, rubrics and, also, similarity results.

We developed two front-ends, the administration dashboard and the revisions panel. Front-ends are programmed in HTML5 with CSS3 styling with the help of Bootstrap and jQuery.

The administration dashboard provides the professor responsible for the course with a list of submissions and revisions to match. The configuration tab is shown in Figure 2. Number of reviews per submission and number of iterations can be configured. Also the usage of the rectified grading instead of the traditional grading, considering the dispersion quantifier measuring the bias in reviews.

The review rating panel allows the configuration of the quantifier of appreciation  $a_{1n}$  mentioned in Section 3.4, and the interval that allows us to limit the value of the quantifier to high number of likes or dislikes.

The *Save configuration* button saves the customization options associated with a specific course or assignment to the database.

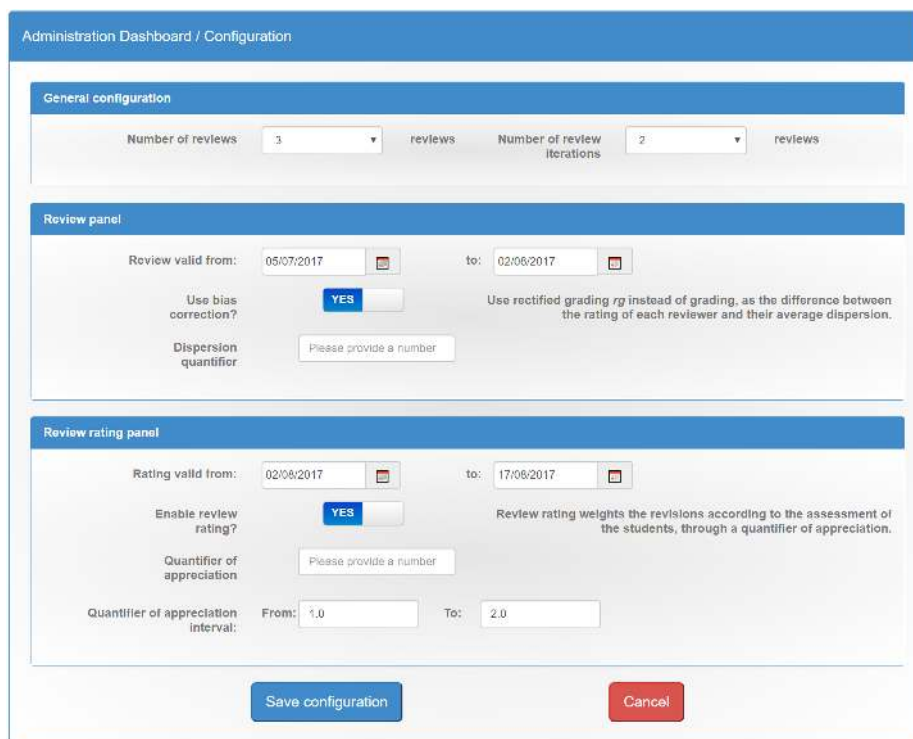


Fig. 2. Screenshot of administration dashboard - configuration

The reviewer's panel is shown in Figure 3, and allows reviewers to view the history of revisions they have made (with no possibility of modification) and select a new assigned review to perform. To do this, they must click on the "Open review form" button where information about the submission is shown, such as the assignment identifier, linked to the *Content provision platform* in this case *Moodle*, where the delivered submission is presented, the similarity value offered by the similarity checker and the link to open the similarity checker tool (in this case *Turnitin*).

If the reviewer wants to make sure that the percentage found corresponds to content of the same author, to other tasks previously delivered during the course, or to other Internet resources, the reviewer can click on the *Link to report* link to open the similarity checker tool.

Regarding the integration features of the proposed solution, the content provision platform requires integration to the MOOC evaluation workflow engine, for user authentication and information exchange related to assignment metadata (information about the delivered assignment such as the delivery *timestamp*, author's *alias*, *assignment id* and *assignment url*, for further retrieval from the revisions panel).

User generated feedback in the form of review ratings (see Figure 1) is also transmitted from content provision platform to the evaluation workflow engine. In the last one we implemented a RESTful interface for easy communication and integration.



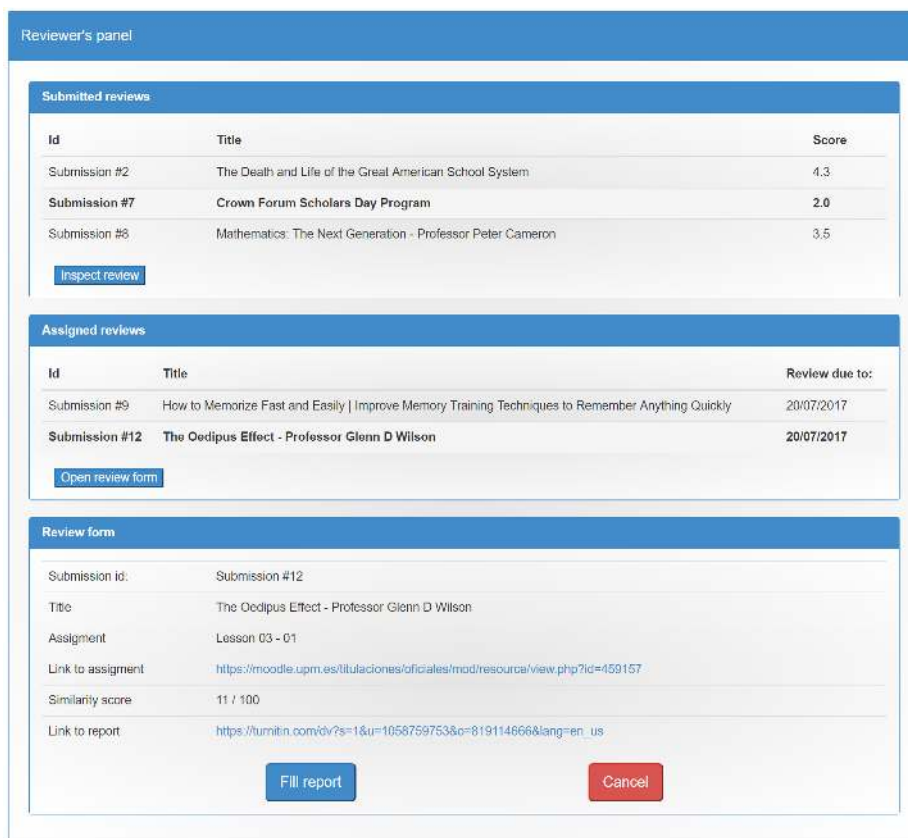


Fig. 3. Screenshot of reviewer's panel

Regarding the communication between the evaluation workflow engine and the similarity checker we used the commercial similarity checker (*Turnitin* [18]) in order to support these functionalities. We implemented an LTI protocol, which is a common solution in educational tools based on the exchange of HTML forms using HTTP operations. Briefly, the workflow engine acts as LTI consumer and the similarity checker as LTI tool provider. Then, a LTI request sends the student's assignment or reviewer's review and their identity to the similarity checker, which returns the results of the revision in a response message. In order to protect the provided information LTI requests are always signed.

#### 4 Evaluation and results

In order to evaluate the performance of the proposed enhanced peer assessment, an experimental validation was planned and conducted. A group of 20 students selected for the experiment were registered as students of a MOOC course, created for validation purposes. Students had prior experience of participating in other MOOC plat-

forms. The defined MOOC contained a selection of 4 videos from the Youtube’s #Education channel. Students were asked to generate a one-page essay on one of the videos, evaluating the current state of higher education and proposing an educational approach. The videos were distributed among the students so that there were 5 students for each video. Once the submissions were made, students were asked to make 5 revisions on the same video they studied, according to a rubric provided, with scores from 0 to 5 points.

Course administrator enabled the tools of detection of biased reviews, the low-level revisions checker and the possibility to rate other student’s revision. The plagiarism detector and the possibility of maintaining an anonymous exchange of messages if the reviewer had any issues to resolve were initially deactivated. For review rating, the students could evaluate the revisions of any other student, excluding those made on their submission.

The reviewer was then given the opportunity to use the initially deactivated tools, and the user was asked if he would like to change his rating.

There have been three results retrievals on this experiment. The first result set responds to submission ratings without considering bias correction or revision ratings, we call it *students#1*. For the second result set we considered bias correction and revisions, but neither the plagiarism detector nor the possibility of maintaining an anonymous exchange of messages, we call it *students#2*. The last result set was produced considering all the functionalities of the correction environment, we call it *students#3*.

Finally, three professors, authors of this work, reviewed and graded the 20 contributions according to the rubric provided, also from 0 to 5 points. The average value of these grades constitutes the *professors#1* result set.

In Table 1 we provide the statistical values of mean and standard deviation of the results obtained. As can be seen the incorporation of assessment functionalities described as contributions in this paper increases the students' overall grade to resemble the professor's overall grade. Related to the standard deviation, the samples collected in the first experiment are more grouped in the mean values, while as the contributions are incorporated, the value of the standard deviation resembles the value calculated by professors, slightly more dispersed.

From the point of view of the significance of the data, we provide a histogram in Figure 4 with the ratings grouped according to grading ranges from 0 to 5, representing the frequency on the ordinate axis.

Figure 4 shows that *students#3* result set is closer to *professors#1* than the rest of the result sets. In addition, the scores between 3 and 4 predominate for all cases.

A t-student test was performed to check if the *students#* datasets were significantly different from the *professors#1* result set. The t-student test was configured with two queues for dependent (paired) samples.

**Table 1.** Statistical values of provided experimentation

| Values             | students#1 | students#2 | students#3 | professors#1 |
|--------------------|------------|------------|------------|--------------|
| Mean               | 3.15       | 3.19       | 3.21       | 3.24         |
| Standard deviation | .847       | .911       | .957       | 1.047        |

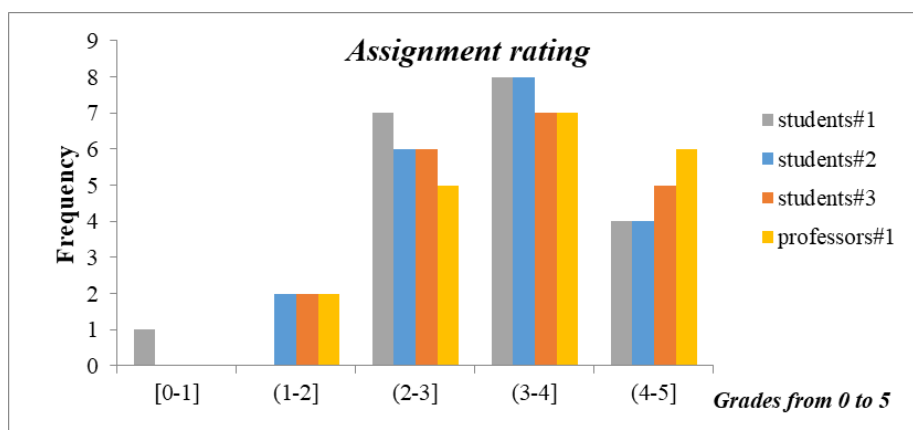


Fig. 4. Histogram of review grades by frequency

Results shown that *students#2* results were significantly closer to *professors#1* than *students#1* (confidence of >99%). Also *students#3* was significantly closer to *professors#1* than *students#1* (confidence of >99%). Finally, *students#3* was significantly closer to *professors#1* than *students#2* (confidence of >95%). These results are expected, and show how the assessments made by the students increase in quality as the contributions proposed in this work are incorporated into their revisions, taking as reference model the revisions made by professors.

A qualitative analysis of the values of the revisions allows us to describe certain phenomena. In the case of the habilitation of the similarity check tool (*Turnitin*), there were 4 occasions in which the reviewers changed their review grade after using the tool, in 50% of cases was to decrease the grade and in the other 50 % to increase it. This indicates that having a tool for detection of similarities does not necessarily produce greater rigor in the reviews, but allows better adjustment of the score if we take as a reference the revisions made by professors.

## 5 Conclusions and future works

In this paper we have proposed an evaluation framework for enhanced peer assessment in MOOCs, considering assignment and review analysis, enabling students to check the originality and their works and their reviews and facilitating professors the detection of low-level feedback and biased reviews.

An analysis of the problems commonly encountered in the evaluation platforms of MOOCs is proposed and requirements that give rise to the design of a system architecture are established. The proposed evaluation framework consists of a set of functional modules that perform the functions of assignment retrieval, review's similarity checking and conformance checking, rubric provision, and review rating provision. It is also described the integration of the environment with the content provision platform and with the professional similarity checking tools, through standard interfaces (LTI).

The scores obtained are calculated according to a scoring algorithm that takes into account the deviation of the opinion of the reviewer regarding the general opinion of the other reviewers. The review grading allows, according to a coefficient of appreciation, to weigh the revisions according to the students' assessment.

Once this environment is implemented, it is evaluated by comparing students' assessments on certain tasks with the assessment of professors of those same tasks, using the same evaluation rubrics. The results indicate that the functions of assignment and review analysis, originality check and biased review correction offer more precise grading of the assignments.

Future work will focus on the profiling of student and reviewer in MOOC environments. This will allow us to more accurately categorize the reviewer's opinion, in order to detect possible deviations that may initially be compensated, but also in order to communicate to the student. In this way, the student will be able to correct these deviations, thus providing an added educational value to the course offered, developing the competency of review and work evaluation, creativity and teamwork, transversal competences defined in the new European Higher Education Area (EHEA).

## 6 Acknowledgment

The research leading to these results has received funding from Universidad Politécnica de Madrid through the 2016-PIE call for “funding educational innovation and improvement of the quality of teaching”. We also thank the Ministry of Economy and Competitiveness for SEMOLA project (TEC2015-68284-R) and the Autonomous Region of Madrid for MOSI-AGIL-CM project (grant P2013/ICE-3019).

## 7 References

- [1] Ouadoud, M., Chkouri, M., Nejari, A., El Kadiri, K. Studying and Analyzing the Evaluation Dimensions of E-learning Platforms Relying on a Software Engineering Approach. *International Journal Of Emerging Technologies In Learning (IJET)*, 2016, vol. 11(01) pp. 11-20.
- [2] Abeer, W., Miri, B. Students' preferences and views about learning in a MOOC. *Procedia-Social and Behavioral Sciences*, 2014, vol. 152, pp. 318-323. <https://doi.org/10.1016/j.sbspro.2014.09.203>
- [3] Dasarathy, B., Sullivan, K., Schmidt, D. C., Fisher, D. H., Porter, A. The past, present, and future of MOOCs and their relevance to software engineering. In *Proceedings of the Future of Software Engineering*, 2014, pp. 212-224 <https://doi.org/10.1145/2593882.2593897>
- [4] Wong, K. Experiences in Constructing a MOOC Specialization. In *Proceedings of the 21st Western Canadian Conference on Computing Education*, ACM, New York, NY, USA, 2016, art. 19, 4 pages. <https://doi.org/10.1145/2910925.2910948>
- [5] Admiraal W., Huisman B., Pilli O. Assessment in Massive Open Online Courses. *The Electronic Journal of e-Learning*, 2015, vol. 13(4), pp. 207-216.
- [6] Anderson, S., Collier, A., Horii, C. V. Designing and implementing MOOCs to maximize student learning. 2013. Online presentation delivered as part of the EDUCAUSE Learning Initiative Spring.

- [7] Wang, K., Lin, B., Rettig, B., Pardi, P., Rishabh Singh. Data-Driven Feedback Generator for Online Programming Courses. In Proceedings of the Fourth ACM Conference on Learning @ Scale (L@S '17). ACM, New York, NY, USA, 2017, pp. 257-260.
- [8] Krathwohl, D. R. A revision of Bloom's taxonomy: An overview. *Theory into Practice*, 2012, vol. 41(4), pp. 212-218. [https://doi.org/10.1207/s15430421tip4104\\_2](https://doi.org/10.1207/s15430421tip4104_2)
- [9] Bali, M. MOOC pedagogy: gleaning good practice from existing MOOCs. *Journal of Online Learning and Teaching*, 2014, vol. 10(1), pp. 44.
- [10] Wong, J. S., Pursel, B., Divinsky, A., Jansen, B. J. Analyzing MOOC discussion forum messages to identify cognitive learning information exchanges. *Proceedings of the Association for Information Science and Technology*, 2015, vol. 52(1), pp. 1-10. <https://doi.org/10.1002/pr2.2015.145052010023>
- [11] Cook, E., Kennedy, E., McGuire, S. Y. Effect of teaching metacognitive learning strategies on performance in general chemistry courses. *Journal of Chemical Education*, 2013, vol. 90(8), pp. 961-967. <https://doi.org/10.1021/ed300686h>
- [12] Griesbaum, J. Students as Teachers in MOOCs? The Double Gain of MOOCs as an in-Class Teaching Method Experiences from a Student-Made MOOC. *International Journal of Information and Education Technology*, 2014, vol. 4(1), pp. 29. <https://doi.org/10.7763/IJJET.2014.V4.363>
- [13] Luo, H., Robinson, A. C., Park, J. Y. Peer Grading in a MOOC: Reliability, Validity, and Perceived Effects. *Journal of Asynchronous Learning Networks*, 2014, vol. 18(2). <https://doi.org/10.24059/olj.v18i2.429>
- [14] Capuano, N., Caballé, S., Miguel, J. Improving Peer Grading Reliability with Graph Mining Techniques. *International Journal Of Emerging Technologies In Learning (IJET)*, 2016, vol. 11(07), pp. 24-33. <https://doi.org/10.3991/ijet.v11i07.5878>
- [15] Sadler, P.M., Good, E. The Impact of Self- and Peer-Grading on Student Learning. *Educational Assessment*, 2006, vol. 11(1), pp. 1-31. [https://doi.org/10.1207/s15326977ea1101\\_1](https://doi.org/10.1207/s15326977ea1101_1)
- [16] Azarnoosh, M. Peer assessment in an EFL context: attitudes and friendship bias. *Language Testing in Asia*, 2013, vol. 3(11).
- [17] Panadero, E., Romero, M., Strijbos, J.W. The impact of a rubric and friendship on peer assessment: Effects on construct validity, performance, and perceptions of fairness and comfort, *Studies in Educational Evaluation*, 2013, vol. 39(4), pp. 195-203. <https://doi.org/10.1016/j.stueduc.2013.10.005>
- [18] Gregory, J., Strukov, A. Ensuring academic integrity in the age of the Internet: evaluating a web-based analytic tool. *International Conference on Computers in Education*, 2002, vol. 2, pp. 1363-1364. <https://doi.org/10.1109/CIE.2002.1186250>

## 8 Authors

**Ramón Alcarria** is an assistant professor at the E.T.S.I Topography of the Technical University of Madrid, UPM. He received his M.S. and Ph.D. degrees in Telecommunication Engineering from the Technical University of Madrid in 2008 and 2013 respectively His research interests are Service Architectures, Sensor Networks, Human-computer interaction and Prosumer Environments (ramon.alcarria@upm.es).

**Borja Bordel** is Ph.D. candidate in telematics engineering at E.T.S.I Telecommunication of the Technical University of Madrid, UPM. He received the B.S. degree in Telecommunication Engineering in 2012 and the M.S. Telecommunication Engineering in 2014, both from Technical University of Madrid. His research interests include

cyber-physical systems, wireless sensor networks, radio access technologies, communication protocols and complex systems (b.bordel@upm.es).

**Diego Martín de Andrés** is an assistant professor at the E.T.S.I Telecommunication of the Technical University of Madrid, UPM. He received his doctoral degree in 2012, and holds a B.Sc in Computer Engineering and an M.S. in Computer Science from the Department of Informatics at the Carlos III University of Madrid. His main research areas are Software Process Improvement, Knowledge Management and Reutilization and Prosumer Environments (diego.martin.de.andres@upm.es).

**Tomás Robles** is a full professor at the E.T.S.I Telecommunication of the Technical University of Madrid, UPM. He received a M.S. and Ph.D. degrees in Telecommunication Engineering from Technical University of Madrid in 1987 and 1991 respectively. His research interests are advanced applications and services for broadband networks (tomas.robles@upm.es).

Article submitted 23 July 2017. Published as resubmitted by the authors 12 December 2017.