CAERS: A conversational agent for intervention in MOOCs' learning processes

Diego Rossi[1(✉)], Victor Ströele[1(✉)], Regina Braga[1], Santi Caballé[2], Nicola Capuano[3], Fernanda Campos[1], Mário Dantas[1], Luigi Lomasto[4], and Daniele Toti[5]

[1] Federal University of Juiz de Fora, Juiz de Fora, Brazil
{diego.rossi, victor.stroele, mario.dantas}@ice.ufjf.br
{regina.braga, fernanda.campos}@ufjf.edu.br

[2] Open University of Catalonia, Barcelona, Spain
scaballe@uoc.edu

[3] University of Basilicata, Potenza, Italy
nicola.capuano@unibas.it

[4] University of Salerno, Fisciano, Italy
llomasto@unisa.it

[5] Catholic University of the Sacred Heart, Brescia, Italy
daniele.toti@unicatt.it

Abstract. Massive Open Online Courses (MOOCs) are a teaching modality that aims to reach a large number of students using Virtual Learning Environments. In these courses, the intervention of tutors and teachers is essential to support students in the teaching-learning process, answer questions about their content, and provide engagement for students. However, as these courses have a vast and diverse audience, tutors and teachers find it difficult to monitor them closely and efficiently with prompt interventions. This work proposes an architecture to favor the construction of knowledge for students, tutors, and teachers through autonomous interference and recommendations of educational resources. The architecture is based on a conversational agent and an educational recommendation system. For the training of predictive models and extraction of semantic information, ontology and logical rules were used, together with inference algorithms and machine learning techniques, which act on a dataset with messages exchanged between course forum participants in the humanities, medicine, and education fields. The messages are classified according to the type (question, answer, and opinion) and parameters about feeling, confusion, and urgency. The architecture can infer the moment in which a student needs help and, through a Conversational Recommendation System, provides the student with the opportunity to revise his or her knowledge on the subject. To help in this task, the architecture can provide educational resources via an autonomous agent, contributing to reducing the degree of confusion and urgency identified in the posts. Initial results indicate that integrating technologies and resources, complementing each other, can support the students and help them succeed in their educational training.

Keywords: Massive Open Online Courses, Recommender System, Conversational Agent.

1. Introduction

With the relevance and prominence of massive open online courses (MOOCs), the debate on new ways to improve this teaching modality's efficiency has gained relevance, as some problems are recurring. MOOCs have expanded and democratized education, since anyone with a computer or smartphone and an Internet connection can access a course in this modality. In most cases, the participant chooses a course free of charge, and, in some cases, there is a small fee for obtaining the certificate [10]. Several major universities produce MOOCs as well, including Harvard, Stanford, and the Massachusetts Institute of Technology, just to name a few. Several platforms also produce courses of this type, such as Coursera, edX, and Future Learn [9].

Although the expansion of MOOCs is somewhat noticeable, the number of evasions is also very high [11]. School dropout is a problem that can be triggered by several factors, one of which is the difficulty of understanding the content and the lack of support to clarify some questions. However, due to a large number of students for a low number of tutors in this teaching modality [14], it is challenging to be able to carry out a pedagogical intervention on time. Therefore, the use of technology would make it possible to accomplish the task.

In this work, CAERS (Conversational Agent in an Educational Recommender System) is presented. CAERS is a software architecture composed of a conversational agent. It works as follows: when a post in a discussion forum indicates that there is a need for pedagogical intervention, the subject in the message is recognized, then the educational content is recommended for clarification of questions.

The study followed four main steps: (i) defining an architecture for the conversational recommender system; (ii) using predictive models to automatically detect the need for student intervention; (iii) building an ontology to extract subjects from the student's question in the MOOC forums; (iv) implementing the system prototype to filter contents corresponding to the student's need.

This work relies upon the recommendation system's overall concepts [12] to design our architecture. The approach proposed includes an infrastructure that automatically recognizes the student intervention necessity and, from a conversational agent, interacts with them to recommend educational resources. The proposed architecture has been evaluated through a usage scenario where messages from a discussion forum were sent, and by using machine learning methods, the system prototype identified the corresponding intervention moment. An ontology extracted the message concepts providing rich data to build the student profile, enabling the recommendation of educational resources to be made.

This paper is structured as follows. In Section 2, the related work is briefly presented. In Section 3, the CAERS architecture is presented, by detailing each layer with its own features. An evaluation of the proposed architecture and prototype is presented in Section 4, where the corresponding experiments are detailed. Finally, Section 5 reports the concluding remarks and mentions ideas for future work.

2. Related work

Several papers discuss the topic of recommending educational resources for students of online courses.

In [8], the authors use a very rich data source to tackle the problem of discussion forums in a massive open course environment, which is often not given the desired attention, as the number of students is often higher concerning the number of instructors. This problem can be very limiting since guided and closely monitored activities by a responsible person seek a lot of effort. A tool has been proposed to label the characteristics of the messages posted on massive course forums to assist this type of interaction. The knowledge is automatically discovered, guiding the instructor in his operations. The information can also be helpful for conversation agents.

The colMOOC platform is proposed in [4], where a conversational agent is modeled in a specific domain to mediate between dyads by triggering appropriate interventions in order to facilitate productive dialogue during chat-based collaborative learning activities in massive education, such as MOOCs. The agent's interaction in the chat is based on the Academically Productive Talk, which is a framework for modeling experienced teacher's interventions during students' dialogues to make them elaborate in the knowledge domain [18].

According to [5], students' dependence on poor assistance and adaptation in massive courses leads them to lose motivation and, consequently, dropout the learning process. As a solution, the authors developed a strategy for adapting activities through a recommendation system. The proposal uses a hybrid recommendation system based on knowledge, supported by ontology, to recommend activities for students in the context of MOOCs.

Considering many questions posted on the forums in massive courses, YouEDU, which helps students, detecting all messages that express content confusion is proposed in [6]. The authors trained a set of classifiers to categorize the forum posts in several aspects: feeling, urgency, and other descriptive variables that guide a classifier to detect confusion. Then, the confusing posts are directed to video excerpts from the course.

The authors in [7] proposed an agent-based recommendation system that aims to help students overcome their disabilities. The system recommends relevant learning resources to provide support to improve the learning experience. An agent-based cooperative system was designed, where agents act independently and update recommendation data, making the recommender more efficient and enhancing experiences on the learning platform.

The work developed in [16] aims to detect and analyze the involvement of course participants in the context of online education, obtaining relevant information related to aspects that indicate student involvement, such as feeling, urgency, confusion, and the probability of evasion of each student is also informed. To accomplish this task, students' posts and comments are considered, using classification algorithms based on machine learning.

In this work, the system architecture proposed uses a combination of techniques to improve the intervention in the students' learning process, detecting their need for help via machine learning techniques and producing the needed help by recommending educational resources through a semantics-based conversational agent.

3. CAERS - Conversational Agent in an Educational Recommender System

As shown in Fig. 1, the CAERS architecture was designed as an architectural structure with seven components: data, extraction, filtering or filter, knowledge model, recommendation or recommender, conversational, and application.

In the **data component**, the information is obtained from external sources that feed the system. This layer has a logical integration between the databases, making it possible to trace the student's profile, details of the student's learning process, objectives, and the possibilities that this training provides.

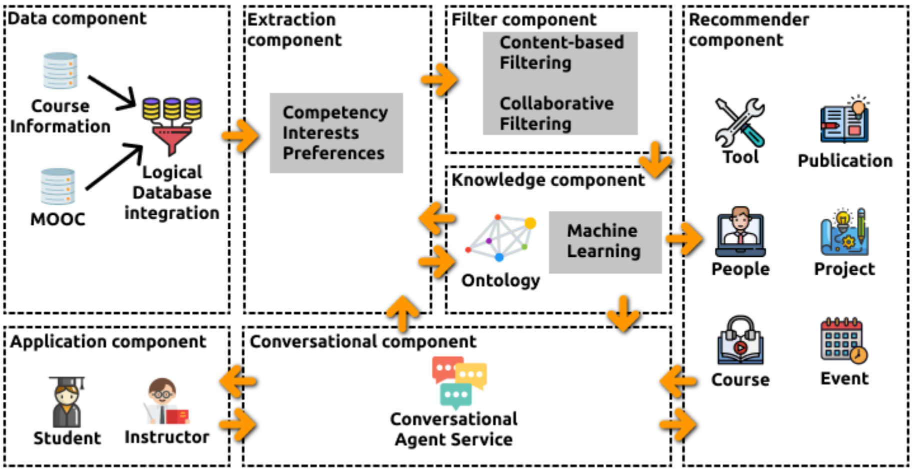

Fig. 1. CAERS Architecture

The **extraction component** is responsible for capturing data from different sources and providing information for both the filter and knowledge models components. The extraction process can be Explicit and/or Implicit.

The explicit extraction occurs when the system asks the user to fill in their data, promoting their initial profile, which can be updated over time. This filing is done through forms, surveys, and even evaluations of the information presented to him.

The implicit extraction occurs when the information to define the profile is obtained without the user's actions, i.e., it occurs in a passive way, reflecting his behavior in an environment. This layer aims to provide information on skills, preferences, and interests.

The **filtering component** is responsible for implementing two types of filters, which serve to recommend the resources. The first type is content-based filtering, which has as a principle the similarity among the recommended items. The basic idea is that if a user likes a particular item, he may also like a similar one. The second type is collaborative filtering, where the principle is to recognize users with similar interests based on positive reviews made by similar users.

In collaborative filtering, after consuming the resources, the users collaborate with evaluations about these resources, indicating whether they are relevant or irrelevant to them. In this way, the system can make new recommendations considering the users' evaluations. Thus, it is possible to recommend resources more adherent to users' preferences considering similar previous evaluations.

The **knowledge component** can bring understanding of the context, providing information for both the conversational agent and the recommendation algorithms. In this component, ontology is used with its rules and knowledge representation. Machine learning techniques are used to create a predictive model [19], where it is possible to recognize the right moment to intervene with recommendations to enrich the student's learning moment.

This component can also adopt techniques responsible for improving the student's profile through ontologies and machine learning. The component techniques are used in conjunction with the ones mentioned above, as they can be applied to enrich user data or objects to be recommended [12].

The **recommendation component** is responsible for selecting open educational resources (OER), which can be obtained from different repositories, eg. learning objects, linked data, and videos. In an attempt to obtain more relevant resources, a relation-ship is made between the user's profile and educational resources. This component defines what will be recommended and the priority that each item will have. Tutors and teachers will be able to analyse the available resources, thus indicating which ones are more adherent to the educational context of the course.

The conversational agent is responsible for obtaining information according to the students' forum messages in the **conversational component**. The agent interacts with the student when the machine learning model identifies that it is necessary. In this component, intervention and recommendation needs are detected based on the data extracted from the conversation between the student and the agent. The agent is connected with the course instructor, providing pre-selected educational resources. Hence, the instructor can analyze and determine the best content for the student.

The **application component** is the interface of interaction with the user, responsible for sending and receiving messages through the conversational agent's API, being responsible for improving the usability experience by presenting the agent's responses.

Once the architecture is defined, it is essential to analyze its feasibility to assist students in a virtual learning environment. In the next section, the architecture is evaluated considering the technological components adopted in the development and the behavior of the architecture when using messages posted by students in a MOOC.

4. Evaluation

An experiment was conducted using data from messages posted by students in MOOC discussion forums. A prototype was developed to verify the proposal's feasibility concerning the components proposed in the architecture, the usage of the predictive model, and the ontology. The underlying question of this experiment is: *can* *the conversational agent identify the student's need for intervention from the forum's posts and carry out an intervention, recommending an educational resource*?

4.1. Environment setup

A Stanford MOOCPosts dataset[1] with 29.604 messages was used to carry out the experiments, captured from forums of three courses from different areas of knowledge, during 14 months. The data were anonymized to avoid identifying the participants, so it is not possible to tell the number of students who collaborated with these interactions.

Each post has the following attributes: text, opinion (0 or 1), question (0 or 1), answer (0 or 1), feeling (from 1 to 7), confusion (from 1 to 7), urgency (from 1 to 7), course type (education, humanities, and medicine), forum_post_id (unique ID of the respective post), course_display_name (course name), forum_uid (student's unique identifier), created_at (post date), post_type (comment or comment read, the last one is attributed to the post that originated a topic), anonymous (true, if the poster appears to everyone as the anonymous name), anonymous_to_peers (if true, the author of the post will appear as 'anonymous' to everyone except the moderator and the instructor), up_count (number of positive votes for the post), comment_thread_id (topic object ID), readings (the total number of readings recorded in the topic).

Another characteristic is that the phrases were labeled with some attributes that add more knowledge to the text fragments. This work's essential property was the confusion measure, which has a scale from 1 to 7, with level 1 for the least confused and 7 for the most confused. The course that the student was enrolled in also helped in carrying out the work [15].

Based on the confusion attribute, it was possible to train the predictive model using a supervised learning algorithm. Based on other studies [1] [2], logistic regression was used, as it presents good results in the classification of texts.

To use logistic regression, the data set was divided into two parts, in the proportion of 80/20, 80% for training, and 20% for testing the model. The model was trained using k-fold validation, with k=10, a cross-validation technique to avoid the model overfitting. For the model not to become biased, it was necessary to balance classes, the technique used to equalize was the oversample, which consists of generating new examples, because applying the undersample could reduce the number of examples, causing the loss of important information for the model training.

Total accuracy of 84.3% was obtained. Therefore, based on the trained model, if the need for intervention is detected, the conversational agent will send a message directly to the student.

The conversational agent is fed with terms extracted from the messages, added to facilitate the identification of the topic present in the text. Fig. 2 shows all the steps taken to add knowledge to the agent. The process starts with obtaining the dataset with the terms, ending with the term set, called entities.

---

[1] https://datastage.stanford.edu/StanfordMoocPosts/

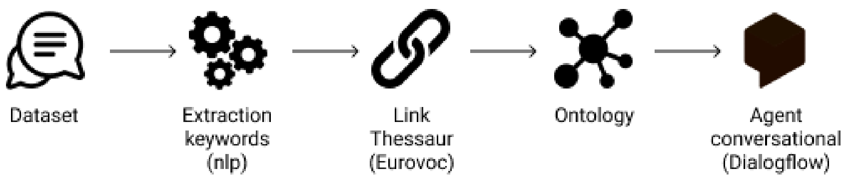

Fig. 2. Workflow to feed the agent.

The first action is to extract keywords from forum messages, and it was necessary to use natural language processing techniques (NLP) to do that. This step was performed using the Python language and the Spacy library to enable keyword identification.

Later, only words that had a match in the Eurovoc thesaurus were considered to avoid identifying terms that did not add relevant information about the domain. In this step, a data set consisting of 5933 descriptive terms was built, organized by areas of knowledge [3].

Once the extraction of the relevant terms has been completed, they are loaded in the ontology, shown in Fig. 3. This step was implemented via the Java OWL APIs, meant to create, manipulate, and serialize OWL ontologies. Using inference machine and ontology rules, instances from the ontology were extracted and then loaded in the Dialogflow tool [13].

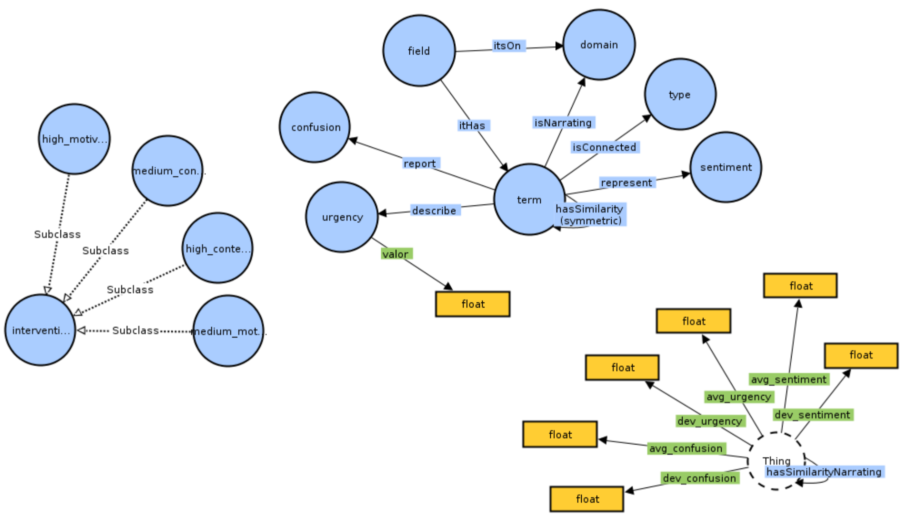

Fig. 3. Visualization of the classes that make up the ontology.

The classes of the ontology shown in Fig. 3 are term, domain, field, type, sentiment, confusion, urgency, intervention, high_motivation, medium_motivation, high_intervention, medium_motivation. The urgency class has an attribute that represents the urgency value. The ontology has some attributes, which are, avg_confusion, dev_confusion, avg_urgency, dev_urgency, avg_sentiment, dev_sentiment.

Dialogflow[2] is a tool that helps in the creation and integration of conversation interfaces. It is possible to integrate Dialogflow with cognitive services through its APIs, such as sentiment analysis and knowledge base [17].

The prototype was developed following these steps and using Dialogflow as an initial user interface. The conversational agent waits for the student message and, based on the predictive model decision, the agent can intervene or not. The agent working is described in the next section.

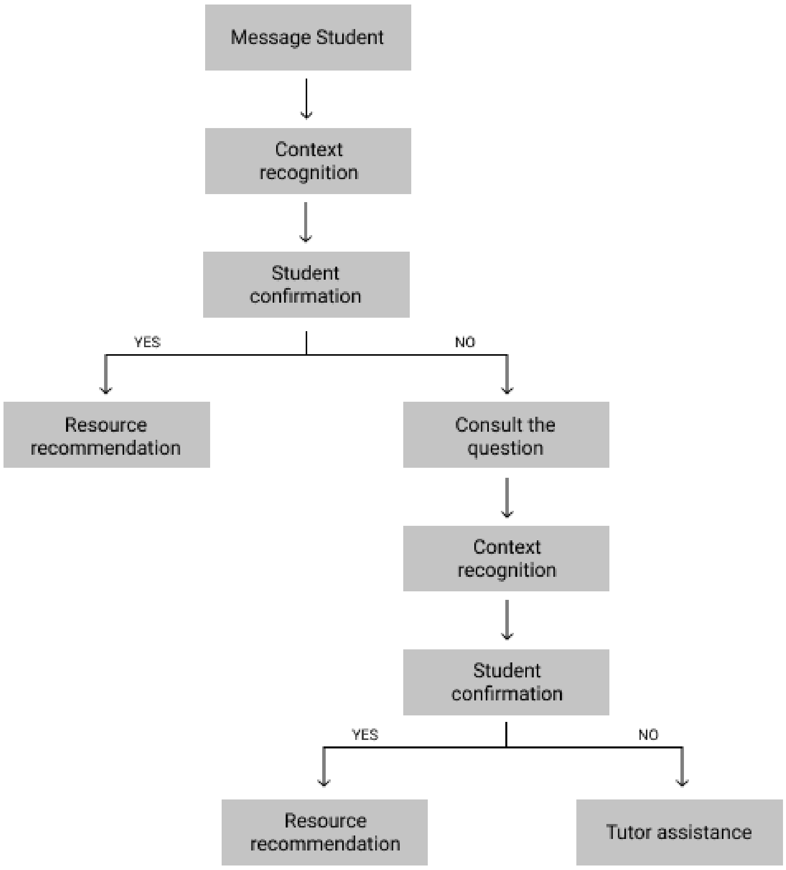

Fig. 4. Diagram of the agent's operation

[2] https://dialogflow.cloud.google.com/#/login

4.2. Conversational Agent in Action

In order for the agent to perform the intervention with the user, it was necessary to create a functioning flow represented in Fig. 4. The first step requires the agent to identify the student message's context, which is enabled by the ontology. Later, the student is asked a question on whether he wants to get more content on the identified topic, and if the suggestion is accepted, a YouTube link is reported. If the student refuses the agent's content suggestion, the user will contribute with more details regarding the message's content. It gives the agent another chance to recognize the message subject and provide the student with useful educational recommendations. However, if the agent fails to identify the student's necessity the second time, the message history is forwarded to the tutor, asking him for a more specific intervention.

In this way, after performing the predictive model's training, the new message sent for the student is classified according to the model. Ten recent messages were selected from the forum and presented to the conversational agent to check its operation, being three of them classified as requiring intervention.

The ontology processed these three messages, and the agent gave an answer for each of them. The students interacted with the agent, saying that they wanted to receive some complimentary educational content. In this first moment, in order to verify the solution feasibility, the agent selected videos from YouTube using the concepts identified by the Ontology in the search string. Once the search was done, the agent provided a link where the student could access the complementary content. Fig. 5 shows one of the chats between the conversational agent and the student.

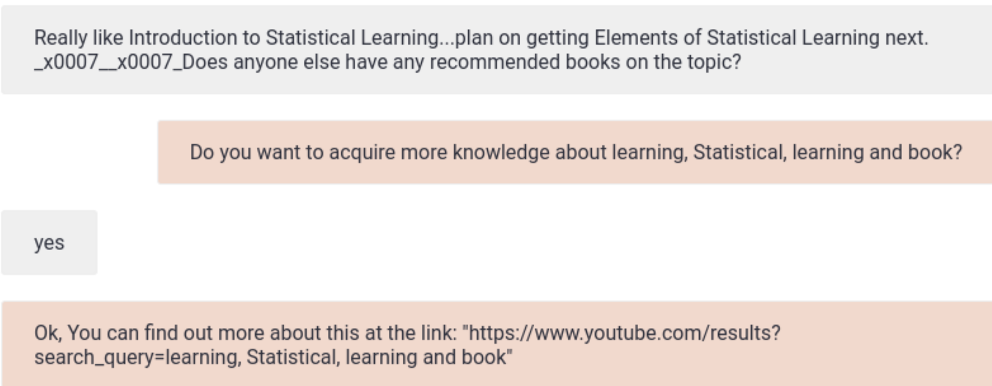

Fig. 5. Operating agent.

If the subject identified by the agent was not relevant to the student, the agent would ask for more details, allowing a new content identification from the ontology. If the student deems the recommendations not interesting, this conversation's history is forwarded to the tutor responsible for the course, requesting help for this intervention.

4.3. Observed Evidence

The experiment demonstrated the feasibility of the proposed architecture, the concepts and technologies involved, the use of the ontology, and the predictive model. It was also feasible to integrate a recommendation system with a conversational agent, covering a complete cycle of questions in a discussion forum within a MOOC.

Concerning the ontology, the conversational agent was able to identify the content of messages posted by students on the forum. On the other hand, the predictive model achieved an accuracy of 84.3%, allowing the agent to identify the messages in which immediate intervention was needed.

The conversational agent's action flow allows educational resources to be recommended directly to the student who presented the question. However, more complex cases, where the agent cannot identify the content of the question correctly, are forwarded to the instructors of the course.

5. Final Remarks and Feature Works

This paper presented CAERS, an architecture for a recommendation system with conversational agent interventions, which acts according to the students' messages posted in MOOC forums. Several works in scientific literature seek to solve similar problems for massive courses; the novelty of this work lies in the union of two powerful techniques to assist in the pedagogical intervention, i.e., recommendation systems and conversational agents.

The works [4] [8] were used as a basis for this paper for recommending educational content with a conversational agent. Given that these works had explored forum and chat posts in MOOCs, they contributed to the development of this research proposal.

In this study, the recommendation system was not widely explored, limiting the recommendation to video resources only, on the YouTube platform. The contents recommended were not evaluated by students, even though it is necessary to check their appropriateness with respect to the student's needs. As a future development, there is a plan to expand the prototype and turn it into a more powerful recommendation system, which may consider the user's profile and recommend open educational resources, the latter even more fitting to the students' needs. We intend to use the teacher's didactic plan to improve resource adherence to the students' needs and avoid bringing inappropriate content concerning the class program.

Another point to be addressed in future work is the delay that must be applied to the agent response. The learning process needs time for reflection and time to allow other students to answer the doubts, providing the generation of a community of interest in the course. Responding to students' questions too early may negatively affect their learning or the student's community.

Regardless, it is the author's belief that this work may contribute to the solution of problems related to the intervention in the learning process, providing clarification of questions in MOOC virtual environments. This contribution may positively influence the students' experience of massive courses and avoid drop-out.

Acknowledgments

This work was carried out with the support of the Coordination of Improvement of Higher Education Personnel - Brazil (CAPES) - Financing Code 001, Federal University of Juiz de Fora (UFJF), CNPq, and FAPEMIG.

References

1. Pranckevičius, Tomas; Marcinkevičius, Virginijus. Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. Baltic Journal of Modern Computing, v. 5, n. 2, p. 221, (2017).
2. Shah, Kanish et al. A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augmented Human Research, v. 5, n. 1, p. 1-16, (2020).
3. Steinberger, R., Pouliquen, B., & Hagman, J. Cross-Lingual Document Similarity Calculation Using the Multilingual Thesaurus EUROVOC. Lecture Notes in Computer Science, 415–424, (2002).
4. Demetriadis, Stavros et al. Conversational agents as group-teacher interaction mediators in MOOCs. In: 2018 Learning With MOOCS (LWMOOCS). p. 43-46. IEEE, (2018).
5. Harrathi, Marwa; Touzani, Narjess; Braham, Rafik. A hybrid knowlegde-based approach for recommending massive learning activities. In: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA). p. 49-54. IEEE, (2017).
6. Agrawal, Akshay et al. YouEDU: addressing confusion in MOOC discussion forums by recommending instructional video clips. (2015).
7. Brigui-chtioul, Imène; Caillou, Philippe; Negre, Elsa. Intelligent digital learning: Agent-based recommender system. p. 71-76. In: Proceedings of the 9th International Conference on Machine Learning and Computing. (2017).
8. Capuano, Nicola; Caballé, Santi. Multi-attribute categorization of MOOC forum posts and applications to conversational agents. In: International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. p. 505-514. Springer, Cham, (2019).
9. Foley, Kimberley et al. Massive open online courses (MOOC) evaluation methods: protocol for a systematic review. JMIR research protocols, v. 8, n. 3, p. e12087, (2019).
10. Hew, K. F., & Cheung, W. S. Students' andtutor' use of massive open online courses (MOOCs): Motivations and challenges. 45–58. Educational Research Review, 12, (2014).
11. Dalipi, Fisnik; Imran, Ali Shariq; Kastrati, Zenun. MOOC dropout prediction using machine learning techniques: Review and research challenges. p. 1007-1014. In: 2018 IEEE Global Engineering Education Conference (EDUCON). IEEE, (2018).
12. Pereira, C.K., Campos, F., Ströele, V. et al. BROAD-RSI – educational recommender system using social networks interactions and linked data. J Internet Serv Appl 9, 7 (2018).
13. Santoso, Heru Agus et al. Dinus Intelligent Assistance (DINA) chatbot for university admission services. In: 2018 International Seminar on Application for Technology of Information and Communication. p. 417-423. IEEE, (2018).
14. Capuano, N.; Caballe, S. Towards adaptive peer assessment for MOOCs. Proceedings of the 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2015), Krakow, Poland, pp. 64-69, (2015).
15. Capuano, N.; Caballé, S.; Conesa, J.; Greco, A. Attention-based Hierarchical Recurrent Neural Networks for MOOC Forum Posts Analysis. Journal of Ambient Intelligence and Humanized Computing, (2020).

16. Toti, Daniele et al. Detection of student engagement in e-learning systems based on semantic analysis and machine learning. In: International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. Springer, Cham, 2020. p. 211-223.
17. SABHARWAL, Navin; AGRAWAL, Amit. Introduction to Google dialogflow. In: Cognitive virtual assistants using Google Dialogflow. p. 13-54. Apress, Berkeley, CA, (2020).
18. Tegos, S., Demetriadis, S., Karakostas, A. (2015). Promoting academically productive talk with conversational agent interventions in collaborative learning settings. Computers and Education (87), 309 – 325.
19. F. Neves et al., "Heath-PRIOR: An Intelligent Ensemble Architecture to Identify Risk Cases in Healthcare," in IEEE Access, vol. 8, pp. 217150-217168, 2020, doi: 10.1109/ACCESS.2020.3042342.