Automatic Assessment of Student Homework and Personalized Recommendation

Xia Wang, Tom Gülenman, Niels Pinkwart
Education Technology Lab
German Research Centre for Artificial Intelligence
Berlin, Germany
first. last@dfki.de

Claudia de Witt, Christina Gloerfeld, Silke Wrede
Institute of Educational Science and Media Research
FernUniversität in Hagen
Hagen, Germany
first.last@fernuni-hagen.de

***Abstract*—AI technologies are applied to automatically assess students' homework texts and to provide intelligent recommendations based on both students' current learning knowledge and whole domain knowledge. Several traditional machine learning methods are evaluated and compared in order to find a suitable method for customizing personalized assessment results.**

Keywords: data analysis, machine learning, intelligent auto-assessment, personalized recommendation, education

I. INTRODUCTION

Artificial Intelligence (AI) technologies are dramatically changing our daily lives in many ways, and there is no exception in the area of education. The education industry as a whole is being transformed by AI, and AI systems are being used to tailor and personalize learning for each individual student [1]. This Forbes article has also predicted that "by 2024 upwards of 47% of learning management tools will be enabled by AI capabilities". Moreover, the New Media Consortium's Horizon Report 2017 listed AI as an important trend in higher education and reaffirmed it in 2018 and 2019 with an adoption timeframe of two to three years [2].

Although originating in the 1950s, AI is reaching another peak in the hype cycle for quite a few years, which is fully benefited from the capabilities of processing big data [3]. Thanks to the World Wide Web and overall digitalization, we are not short of big amounts of data for decades, but the challenge was how to deal with it in real time. Since technologies of large storages became feasible and mature for storing fast-growing volumes of diverse data with high velocity [4]; and fast, distributed and parallel processing systems based on GPUs [5] became able to process native SQL queries across billions of records in milliseconds, AI is having its new era.

We are closely working with several universities running popular online learning systems (e.g., Moodle, Opal[1], ONYX[2], and MOOCs) to provide courses, exercises, assignments, and fora to students for their studies. Lots of student data and learning data have been accumulated by these universities. Therefore, our work focuses on applying big-data-driven AI technologies to these online learning systems in the education industry to provide intelligent learning services.

Our vision is to systematically construct a comprehensive AI learning system for universities or educational companies, not only focusing on several specific challenges. With this perspective, we fundamentally build our AI applications on top of the three models presented in [6,7], which will be reused and redefined with our data insights, as follows:

- *Domain model* captures numerous and complicated domain knowledge in a fine structural and semantic way in order to specify what is to be adapted.
- *Learner model* depicts learner features (e.g., personal information, motivations, interests, goals, preference, plans, and so on). It is used to track all kinds of learning behaviors, activities, and processes, then to tell according to what parameters it can be adapted.
- *Pedagogical model* expresses how the adaptation should be performed, considering a selection of didactical methodologies for the current purpose, e.g., individual learning objective and learning context.

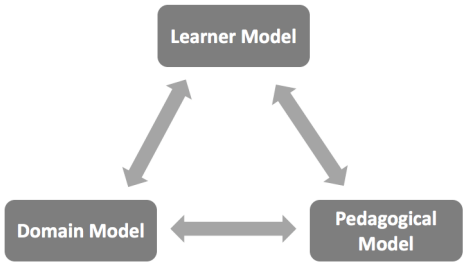

Figure 1. The Three Models of an Intelligent Learning System

Currently, we are at the first stage of our projects[3,4] which is adopting the above described three models (see Fig. 1). Based on the current states of our learner model (LM) and domain model (DM), this paper focuses on one specific challenge: *How to grade student homework texts and, then, provide some personal informative recommendation based on individual learning knowledge?* This is a real learning scenario occurring at FernUniversität in Hagen.

Our solution proposed in this paper is based on machine learning. It consists of two parts: (1) automatically reviewing the students' answers to the required knowledge questions, and (2) providing a real-time adaptive knowledge recommendation according to the student's performance.

[1] https://bildungsportal.sachsen.de/opal/
[2] https://www.bps-system.de/cms/produkte/onyx-testsuite/
[3] AI.EDU Research Lab, fernuni-hagen.de/forschung/schwerpunkte/ddll_projekte_al-edu.shtml
[4] tech4comp Project, https://tech4comp.de/

II. RELATED WORK

As addressed in the web post [9], researchers at Stanford University recently combed through over one million homework submissions from a large MOOC class offered in 2011. Of over 120,000 enrolled students only about 10,000 completed all homework assignments, and some 25,000 submitted at least one assignment. This may not be a typical example, but is not rare. Automatically grading homework assignments is a significant challenge in many online learning systems.

To have automatic grading systems for short answers has become a widely stated demand during the past decade [10]. Similar to our work, [11] aimed to develop an effective and impartial grading system for short answers in educational measurement for a MOOC system. Their proposed approach was based on non-negative semi-supervised document clustering technologies [12].

Not surprisingly, many studies were done on programming assignment assessment [13,14,15,16], which are quite different from the free style of plain text homework that we deal with. Since programming homework results in executable code, it suggests itself to use the test-driven software development method [13]. Alhami et al. [12] also focused on software code assignments, and used a code similarity as the grading measure. They first parsed key abstractions or concepts from students' answers, and gave weights to the code keywords. Then, the similarities were measured between students' assignments by a Euclidean distance and calculated between each assignment and all other assignments. Differently, the approach in [15] relied more on the formal semantics of a program, tried to capture the semantics of execution paths as its grading measure.

Back to our challenge, it is essentially a multi-class text classification problem [17]. The traditional machine learning methods, i.e., supervised classification (e.g., decision trees, k-nearest neighbor and Naive Bayes classifiers) and unsupervised clustering algorithms (e.g., k-means and hierarchical clustering), can all achieve good results. The success of these learning algorithms relies, however, on their capacity to understand complex models and non-linear relationships within data. As presented here, we have tried out four supervised classification algorithms in order to select a suitable learning model from our labeled dataset.

Another popular, yet noteworthy approach of carrying out the text classification tasks is to use convolutional neural network (CNN) transformers [21]. Brownlee [19] and Goldberg [20] both agree that deep learning for natural language processing generally offers better performance than classical linear classifiers, especially when used with pre-trained word embedding, which often leads to superior classification accuracy. Goldberg [20] comments that CNNs with pooling layers are effective at document classification, as they are able to pick out strong local clues regarding class membership as features (e.g., tokens or sequences of tokens), regardless of their positions within the input sequences.

Bidirectional Encoder Representations from Transformers (BERT) is the well-used language model published by Google. It is designed to pre-train deep bidirectional representations from an unlabeled text by jointly conditioning on both left and right context in all layers [18]. As a result, a pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

At the moment, our current dataset is not yet big enough to apply deep learning methods, but it will be considered in our future work, since we are continuously collecting learning data from a large number of students.

III. DATA AND DATA COLLECTION

A. Course Modules of Moodle System

Moodle is an open-source learning management system widely trusted by many schools, universities, institutions, and organizations. Based on the work of [22], Moodle is developed on pedagogical principles and used for blended learning, distance education, flipped classroom, and further e-learning projects. Also, as was declared [5], "*Moodle's worldwide numbers of more than 90 million users across both academic and enterprise-level usage makes it the world's most widely used learning platform*".

Our current testbed is built on top of two course modules, which are offered to the students of the *Educational Science and Media Education* department at FernUniversität in Hagen. The targeted testers are the registered students who study with Moodle, accessed the learning resources or participated in various learning activities. Our data collection and usage are strictly following the General Data Protection Regulation (GDPR[6]).

B. Selected Dataset

Data were collected regarding a course module in the bachelor study program, which constitute our test data here. In total, 251 students had registered to the *module,* which was taught in the *Summer_18* and the *Winter_18/19* semesters. To pass this course module, students need to take an examination comprising six knowledge questions and two reflection questions, each one to be answered by a short text in German. Students are required to complete at least three knowledge questions plus one reflection task. Based on evaluating the answers given, students are offered subsequent modules best suited for them.

TABLE I. DATASET DESCRIPTION

Question	*Answer/Stud*	*Best/Ave. Grade*	*0.0*	*0.5*	*1.0*
Question1	104/131	1.0/0.5	12	16	76
Question2	98/131	1.0/1.0	1	8	89
Question3	**116/131**	**1.0/0.5**	**20**	**40**	**56**
Question4	30/120	1.0/1.0	-	-	-
Question5	38/120	1.0/0.5	-	-	-
Question6	92/120	1.0/1.0	1	12	79
Reflection1	20/120	3.0/2.25	-	-	-
Reflection2	19/120	3.0/2.25	-	-	-

[5] https://docs.moodle.org/38/en/About_Moodle
[6] https://gdpr-info.eu/

Tab. 1 presents the details of our dataset. It contains the following information:

- 251 students submitted 517 short answers distributed over 8 questions.
- All answers are in German and the average length is around 50 words.
- Each question has a sample solution created by the tutors.
- To each answer a score was given. The maximum score of Questions 1 to 6 is 1.0.
- To the answers of Question 1 to 6 one of three possible grades was given: 0.0, 0.5, and 1.0.
- Additionally, all answers were manually annotated by the tutors. These annotations are taken as positive or negative indicators for future recommendations.

As we see in Tab. 1, we eventually selected Question 3 as our test data for this practice, only. The reason is that just 30 and 38 valid answers were given to Question 4 and Question 5, respectively, which is too little for testing. The same applies to the two reflection questions. Although Question 2 has quite some valid answers, they are not very evenly distributed, similarly to the Question 1 and 6.

IV. Data Preparation

A. Data Preprocessing

Referring to our task, we applied the following cleaning techniques to the selected dataset: tokenizing, removing punctuation and stop words, stemming and dealing with the encoding errors. But we did not check or correct common typographical errors nor misspellings, because the misspelled words should be taken into consideration for the students' assessment. We also kept the numbers and acronyms.

B. Word Embeddings and TF-IDF

Raw text cannot directly be fed into machine learning models. Text data must be encoded in a certain way as numbers in order to be used as an input or output for machine learning algorithms. The measure Term Frequency/Inverse Document Frequency (TF-IDF) is a popular way to represent text in a vector space, by reflecting how important a word is to a document in a collection or corpus. In recent years, word embedding [18] has become the most popular technique in the area of natural language processing (NLP) for representing words and documents by using a dense vector representation, which allows words with similar meanings to have similar representations.

When it comes to the question of how to apply word embedding for our task, we have considered two different approaches:

a) *Train our own word vector*: several textbooks had been used for the course module, which is possible to be processed as the text corpus. For example, the textbook *Studienbrief_***61* has 152 pages of material. *Gensim* is an open-source Python library for NLP, which could be considered as the tool for handling large text collections to transform the text into a word vector.

b) *Use pre-trained word embedding*: a pre-trained model is nothing else than a file containing tokens and their associated word vectors [19]. Based on results for word representation, at the moment NLP practitioners seem to generally prefer Stanford's *GloVe* over *Word2Vec* developed by Google. We have downloaded the smallest *GloVe* pre-trained model. It is an 822 Megabyte zip file with 4 different models (50-, 100-, 200- and 300-dimensional vectors), which were trained on Wikipedia data with 6 billion tokens and a 400,000 words vocabulary.

Owing to time limitation, in the future we will apply the above approaches for improvement. So far, a straightforward TF-IDF transformation is used in the current implementation.

V. Text Classification by ML Methods

After preparing the data, some machine learning methods could be tried out to select the best one for our task. Four models from the *scikit-learn* library were tested here, namely *Multinomial NB*, *Logistic Regression*, *Random Forest Classifier,* and *Linear SVC*.

First, by using *TfidfVectorizer*, each of the 116 answers was represented by 103 features, which are the TF-IDF scores of unigrams and bigrams. It is obvious that a by-product provided by the *TfidfVectorizer* is the list of the most correlated terms in each category. For example, for the group of answers having the score 1.0, the three most correlated unigrams terms are "*wechselbezug, entwicklung, handeln*" and the correlated bigrams are "*mediatisierung metatheorie, beharrenden innovativen, soziokulturellen wandel*".

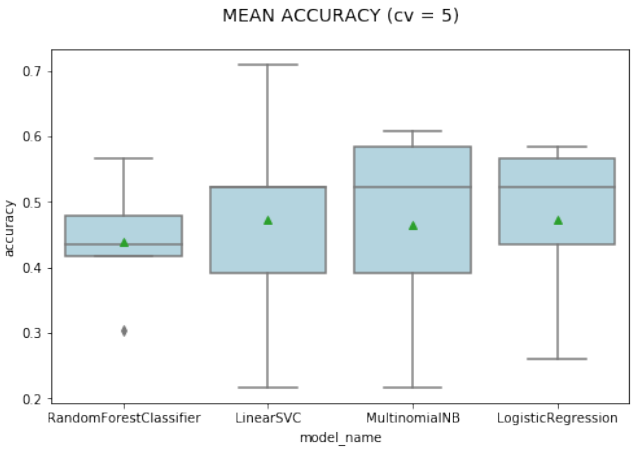

Figure 2. Mean Accuracy of Four Machine Learning Methods

After training the four models, we obtained the mean accuracy and standard deviation of each model. As shown by Fig. 2, their mean accuracies from left to right are 0.439855, 0.472101, 0.464493 and 0.473188, i.e., they are quite close. It means there is no big difference in our current dataset.

Since the Multinomial Naive Bayes classifier (MultinomialNB) does a slightly better job than the other three methods, we then looked into its confusion matrix for the details of its performance, see Fig. 3. The size ratio of the test set is 0.25, which means that it randomly selects 29 out of 116 answers as the test data. Fig. 3 showed us that 9 answers of category 2 are wrongly predicted as category 3.

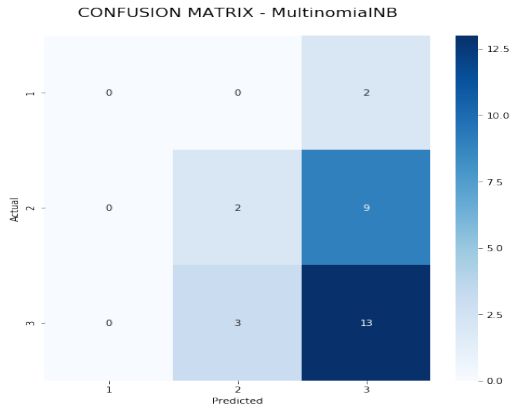

Figure 3. Confusion Matrix of *MultinomialNB*

VI. KNOWLEDGE RECOMMENDATION

The second contribution of this work is, after auto-assessing a student's homework, that it goes further to provide personalized recommendations based on the student's performance and the domain knowledge of the course module. Since the domain model is under development, here we just simulate it with a knowledge graph in an abstract way.

Suppose the domain knowledge of a course module is represented as DK, i.e., a tuple of (C, R, ∂_R), where C is a set of concepts c or concept units C_i, R is a set of concept relations, and ∂_R: $R \rightarrow C \times C$. Similarly, the knowledge of the student m is represented as SK_m, where $SK_m = (C_m, R_m, \partial_{Rm})$ and $SK_m \subseteq DK$. As defined by [23], a conceptual unit C_i is a group of related concepts belonging to the domain knowledge of an instructional system. The concept unit is a very practical device: for instance, using the partial order over a set of concept units allows to represent the learning path (LP) of a student as $LP = (C_i, \leq)$.

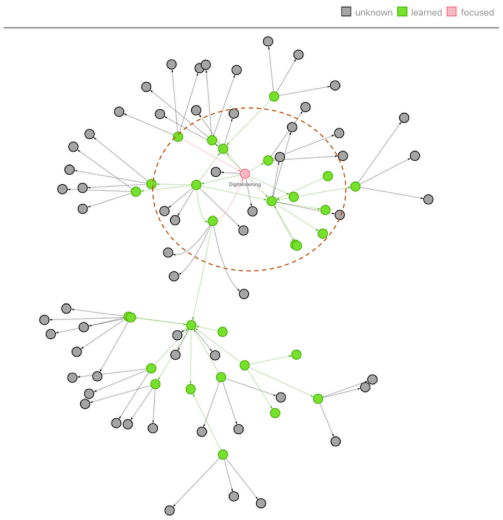

Figure 4. Student Knowldege vs. Domain Knowledge

Since all of the students' learning activities and learning results that happened in Moodle are being recorded, we can track the students' knowledge. We generalize and reuse the learner model of [24], which enables us to evaluate how well a student mastered the knowledge he/she learnt and diagnose the competencies of the student.

Fig. 4 partially visualizes the coverage of a student's knowledge over the domain knowledge during his/her learning process. The knowledge nodes in green were studied by the student and approved by the student's exercise results. The nodes in gray represent the knowledge that is not yet learnt by the student. To customize an individual recommendation, our system focuses on a chosen node to select all connected nodes within a certain number of lengths, e.g., in two lengths (see the red circle).

VII. IMPLEMENTATION

Before applying our work to a real learning management system, we created a local prototype to evaluate the results. Fig. 5 has illustrated that a student entered his/her answer to the question "*Definieren Sie mit wenigen Worten den Begriff Mediatisierung*". After submission, he/she got an automatic score of 0.5, which is the second-best grade.

Since there is room for improvement, and given the student wants to further deepen his/her knowledge on this topic, he/she is given the possibility of looking into personalized feedback and a sample solution. By clicking the "recommendation" button, an analysis of his/her answer and an individual recommendation are generated. As we can see, in the student's answer, the positive indicators contributing to his/her answer are highlighted in green and the negative ones in red.

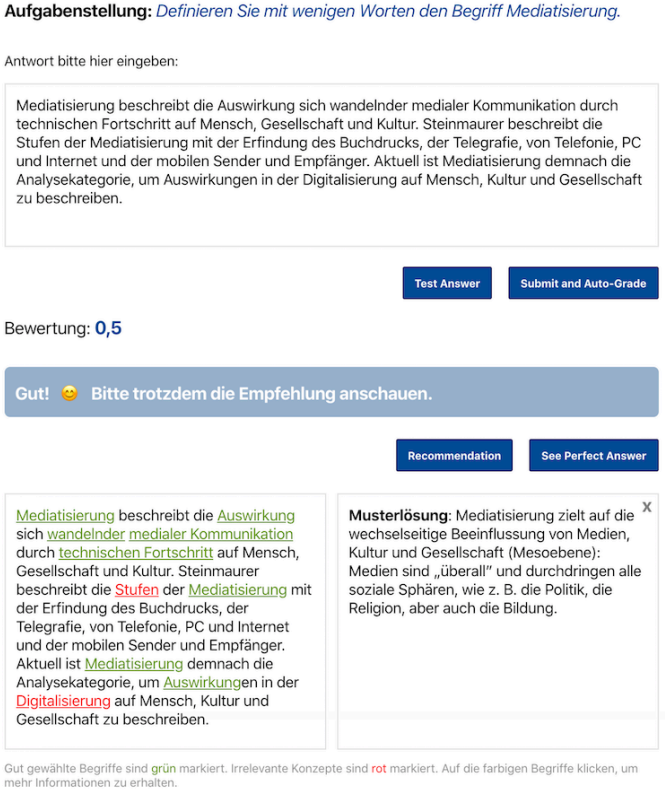

Figure 5. Auto-assessment of Question 3

Furthermore, if the student clicks on a concept, e.g., "*Digitalisierung*" in red, in order to correct his/her knowledge on this concept, a customized SPARQL query is generated for the domain knowledge and the query result is visualized in a graph (see Fig. 6). The resulting graph is the response to the query for the concept "*Digitalisierung*" from domain knowledge, which is also filtered by the student's current knowledge competence. That is, returned is only the necessary information, e.g., the unlearnt, missing or misunderstood knowledge concepts and so on.

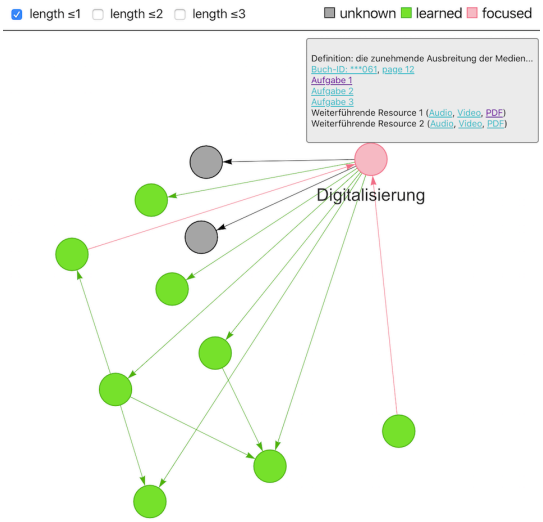

Figure 6. A Student's Knowledge Review on "Digitalisierung"

Normally, when a knowledge node is clicked, its sub-domain knowledge is presented together with different types of relations. When placing the mouse over a concept node, a list of information regarding the concept knowledge is displayed as well, e.g., its definition, source link, competence information, attached exercises, reading materials and so on. If the student is interested in the recommendation, he/she can click on the extra exercise to correct or deepen his/her knowledge.

VIII. Conclusion

Intending to assist students in their usual exercises by offering intelligent knowledge recommendations through their online learning system, notably Moodle, the challenge of classifying student homework by using machine learning methods was addressed. The approach turned out to be feasible. So far, the classification accuracy is considerably above chance, which we consider a promising result given the small size of the sample investigated. Once we will have a big enough dataset, we will improve the results by considering some deep learning transformer. After auto-assessment, we went further to provide some personal recommendations on the students' current knowledge.

References

[1] R. Schmelzer, "AI Applications In Education", Forbes article, Jul 2019, https://www.forbes.com/sites/cognitiveworld/2019/07/12/ai-applications-in-education/

[2] A. Becker, S. Cummins, M. Davis, A. Freeman, A. Giesinger et al., NMC Horizon Report: 2017 Library Edition. Austin, TX: The New Media Consortium.

[3] I. Yaqoob, I. Hashem, A. Gani et al., "Big data: From beginning to future", Int. J. of Information Management, vol. 36, pp. 1231–1247, Dec 2016.

[4] A. Siddiqa, A. Karim and A. Gani, "Big data storage technologies: a survey", Frontiers Int. Technol Electronic Eng. 18, pp. 1040–1070, Sep 2017.

[5] E. Sitaridi, "GPU-Acceleration of In-Memory Data Analytics", Ph.D thesis, Columbia University, 2016.

[6] M. Elson-Cook, "Student modeling in intelligent tutoring systems", Artificial Intelligence Review, 7, 227–240, 1993.

[7] P. Pavlik, K. Brawner, A. Olney and A. Memphis, "A Review of Learner Models Used in Intelligent Tutoring Systems", published by Army Research Labs/ University of Memphis, pp 39–68, Jan 2013.

[8] M. Dunleavy and C. Dede, "Augmented Reality Teaching and Learning", Handbook of Research on Educational Communications and Technology. Springer, New York, NY, pp. 735–745, 2014.

[9] B. Lorica, "Semi-automatic method for grading a million homework assignments", http://radar.oreilly.com/2013/10/semi-automatic-method-for-grading-a-million-homework-assignments.html, Oct 2013.

[10] C. Brew and C. Leacock, "Automated short answer scoring", Handbook of automated essay evaluation: Current applications and new directions, pp. 136, 2013.

[11] S. Jing, "Automatic Grading of Short Answers for MOOC via Semi-supervised Document Clustering", Proc. Int. Conf. On Educational Data Mining, pp 554–555, 2015.

[12] L. Lelis and J. Sander, "Semi-Supervised Density-Based Clustering", Proc. IEEE Int. Conf. on Data Mining, Miami, Florida, USA, pp. 842–847, Dec 2009.

[13] Ö. Demir, A. Soysal, A. Arslan, B. Yurekli, O. Yilmazel, "Automatic Grading System for Programming Homework", AutomaticGS, 2010.

[14] I. Alhami and I. Alsmadi, "Automatic Code Homework Grading Based on Concept Extraction", Int. J. of Software Engineering and its Applications, vol. 5, no. 4, pp. 77–84, 2011.

[15] X. Liu, S. Wang, P. Wang and D. Wu, "Automatic grading of programming assignments: an approach based on formal semantics", 41st Int. Conf. on Software Engineering: Software Engineering Education and Training, IEEE Press, pp. 126–137, 2019.

[16] D. Morris, "Automatic grading of student's programming assignments: an interactive process and suite of programs," 33rd Annual Frontiers in Education, pp. S3F-1, 2003.

[17] K. Kowsari, K. J. Meimandi et al., "Text Classification Algorithms: A Survey", Information 2019, vol. 10(4), pp. 150, 2019.

[18] J. Devlin, M. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint, 2018.

[19] J. Brownlee, "Deep Learning for Natural Language Processing", Machine Learning Mastery, pp. 414 , Nov 2017.

[20] Y. Goldberg, "A Primer on Neural Network Models for Natural Language Processing", CoRR abs/1510.00726, 2015.

[21] Y. Kim, "Convolutional Neural Networks for Sentence Classification", Proc. EMNLP , Oct 2014.

[22] M. Dougiamas and P. Taylor, "Moodle: Using Learning Communities to Create an Open Source Course Management System", EDMEDIA, 2003.

[23] E. Pecheanu, L. Dumitriu and C. Segal, "Domain Knowledge Modelling for Intelligent Instructional Systems", Int. Conf. on Computational Science, pp. 497–504, 2004.

[24] G. Goguadze, S. Sosnovsky, S. Isotani and B. McLaren, "Evaluating a Bayesian Student Model of Decimal Misconceptions", Proc. 4th Int. Conf. on Educational Data Mining. pp. 301–06, 2011.