IFSE - Personalized Quiz Generator and Intelligent Knowledge Recommendation

X. Wang, H. Li, A. Zimmermann, N. Pinkwart
Educational Technology Lab
German Research Centre for Artificial Intelligence
Berlin, Germany
first.last@dfki.de

S. Werde, L. van Rijn, C. de Witt, B. Baudach
Research Cluster D²L²
FernUniversität in Hagen
Hagen, Germany
first.last@fernuni-hagen.de

***Abstract*—This paper has presented an AI-based quiz subsystem that customizes personalized exercises for individual learners together with accurate instant feedback and knowledge recommendation after taking quizzes, eventually aiming to assist learners with self-regulated learning. A knowledge-based quiz generation algorithm and a set of intelligent feedback recommendation algorithms are proposed, which are designed for a large number of exercise materials from various courses to numerous students' self-learning and, generically, for all kinds of knowledge domains. Intelligent feedback for student exercise (IFSE) application has been developed and integrated into the Moodle learning management system in order to conduct field experiments and to be evaluated by university students.**

***Index Terms*—Personal Quiz, Intelligent Feedback, Ontology, Knowledge Graph, Knowledge Recommendation, Self-regulated Learning**

I. Introduction

Aiming to apply the latest AI technologies in higher education and effectively assist students' quality online learning, we have encountered many difficulties at the beginning of our research project *AI.EDU Lab*. In full compliance with data protection regulations, we found that the learning management system (LMS) in use provides coarse-grained and digitizing learning content on a small scale, only. The collected data sets (for the specified data analysis purposes) are generally partial and fragmented, or even missing. For instance, in the current quiz component, the provided quiz questions come in small amounts, they are the same fixed quizzes and undifferentiated for all students. They are normally created manually by tutors without being linked to or annotated with the tested knowledge concepts. Only the students' answers might be tracked and collected. Thus, it is barely possible to apply machine learning for any profound data-centered analysis and to expect meaningful results.

Based on the prevailing requirements and real use cases, we started to design a new AI-based learning system (named as *iLS*) piece by piece and case by case [2]. The development takes place in three steps, viz., first building our new learning system with AI-enabled and trackable new features, then deploying it and having it tested by university students and, finally, based on the collected anonymous data, conducting various machine learning data analytics to obtain new learning insights and improve students' self-regulated learning (SRL).

This paper mainly focuses on the quiz/exercise subsystem of our *iLS*, i.e., intelligent feedback for student exercise (*IFSE*) (see Section VI). The proposed quiz subsystem changes the ways of teaching and learning due to the use of new designs and algorithms. Basically, two types of algorithms are applied in this paper, one for generating personalized quizzes (denoted as *PQ*, see Section IV), and the other one for learning and knowledge recommendation (denoted as *KR*, see Section V).

The innovation of the IFSE quiz system is to separate quiz questions from their options into a question pool and an option pool, respectively, and to formally model them with ontologies. Each quiz question and its question options are semantically linked and further linked to knowledge concepts or learning objects being tested (defined in a domain knowledge graph). Meanwhile, students' knowledge concepts competence masteries of their personal knowledge graphs are calculated and updated by taking their quiz results as evidence.

With these innovations, the semantic relations among quizzes, domain knowledge, and learning resources are able to be represented and mined in depth. Since our domain knowledge is defined with formal logic with reasoning capability, eventually we migrate the previous quiz system from the static quiz (providing the same to all students) to a dynamic state (providing different quizzes to different students) and, ultimately, to the personalized and adaptive level.

II. Related Work

For more than two decades, self-regulated learning (SRL) is an important research area within education psychology, especially right now in the era of increasingly prevalent online learning and digitalization [12], [13]. SRL is learning guided by metacognition, strategic action (planning, monitoring, and evaluating personal progress against a standard), and motivation to learn [5]. As pointed out by [7], the most effective learners are self-regulating. A self-regulated learner monitors, directs, and regulates actions towards goals of information acquisition, expanding expertise, and self-improvement. *Feedback* is inherent in and a prime determiner of the SRL process and affects students' cognitive engagement with tasks and achievement.

This work is based on our cooperative research project, AI.EDU Lab funded by D^2L^2 at FernUniversität in Hagen.

978-1-6654-3418-8/22/$31.00 ©2022 IEEE
DOI 10.1109/ICSC52841.2022.00041

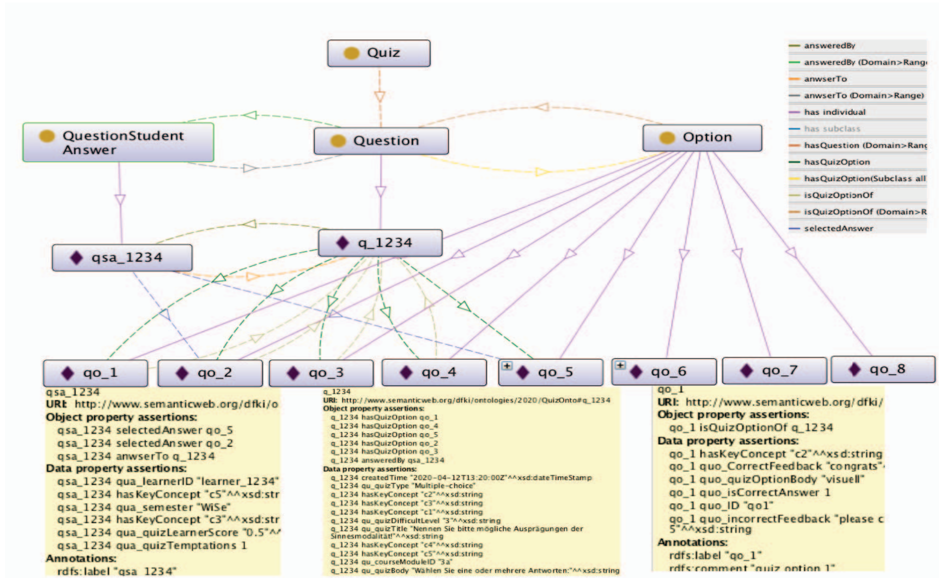

Fig. 1. Quiz Ontology in Protégé v5.5.

Six SRL models have been well-reviewed in depth from a number of aspects by [8]. At this moment, our work is more inclined to refer to *Winne's and Hadwin's* model [9], [10], because it is strongly influenced by the Information Processing Theory [6] and emphasizes domain knowledge, knowledge tasks, and knowledge beliefs. From the implementing learning setting, we currently focus on how to enhance the external feedback of this model with AI, which means more intelligently generating proper tasks, tracking performance, providing feedback, self-adapting tasks with cues, and computing knowledge beliefs. The other parts of Winne's and Hadwin's model, such as profiling a goal, cognitively evaluating the discrepancy between goal and current learning state, and contextual or collaborative learning, we consider as our future work.

With time, quite a number of intelligent tutoring systems (ITS) [11] has emerged to support self-regulated learning [23], [24]. In general, an ITS is an AI-powered computer system that aims to provide customized instruction or feedback to learners without intervention from teachers. For example, in a dialogue-based ITS with almost the same goal as ours, aiming to pinpoint in-/correct concepts in student answers, [22] uses neural discourse segmentation and classification methods to yield a relational graph, then to match student answers with reference solutions, and finally to generate personalized feedback. The difference is that they use a bottom-up approach in the context of discourse analysis.

Moreover, increasingly machine learning, deep neural networks (NNs), and natural language processing (NLP) are used to generate recommendations [14]–[16] in e-learning. Although different methods are employed, there are three main kinds of recommendations, viz., content-based, collaborative filtering-based, and knowledge-based. For knowledge-based recommendation, applying ontologies is a feasible approach [25], [26] in order to formally represent knowledge and pedagogical rules and to provide potential reasoning ability. For exactly this reason, we invited four domain experts to manually build our domain ontologies. Although it is time-consuming and difficult, the domain knowledge built provides solidly structured content for future accurate recommendation.

On the other hand, quizzes are normally generated by teachers or tutors who teach the subjects and know their students' learning states to some extent. Automatically generating quizzes is an attractive research challenge, especially when fusing technologies of neural networks and natural language processing [17]–[21]. For example, [19] uses NLP and optical character recognition (OCR) technologies to extract keywords from uploaded text images (e.g., scanned books) and from the internet to generate facts-based multiple choice questions (MCQ). Moreover, [17], [18] work on generating distractors of MCQ from free text with different methods, such as a point-wise ranking support vector machine, a list-wise ranking neural network and a ranking generative adversarial network (GAN). The results of their experiments are very inspiring. Compared with their work, we are focusing more on forming adaptive and personalized quizzes based on existing quiz material and the domain knowledge linked. Automatically and semantically generating quizzes from raw resources will be our next step.

III. Semantic Foundation

Before presenting any core algorithm, the question of how to model a quiz and how to define the domain knowledge have to be addressed, because they are the semantic foundation of our work. To this end, ontologies and the tool Protégé v5.5 are used.

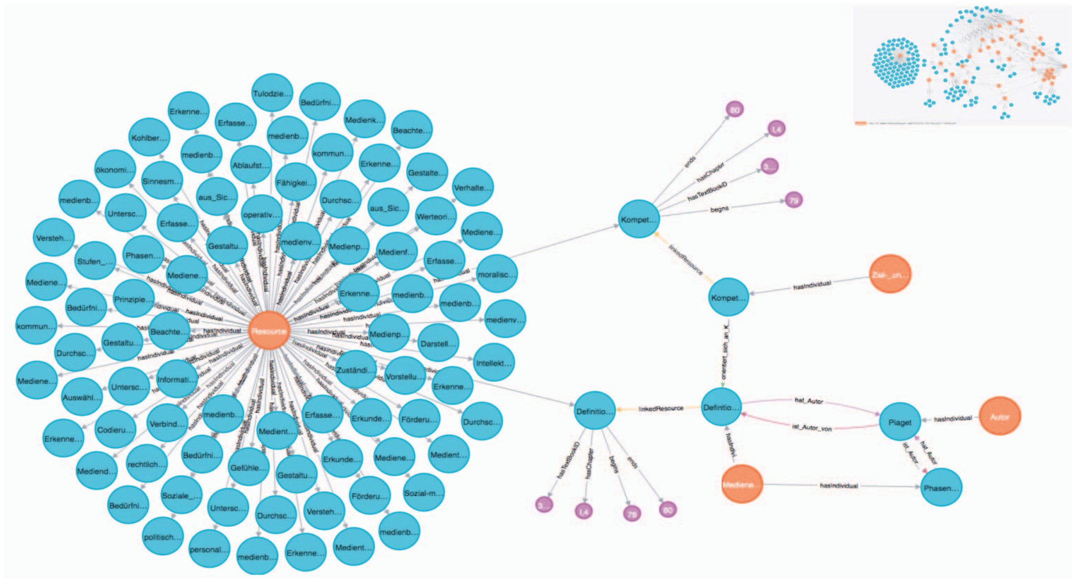

Fig. 2. Knowledge graph of the course module_x.

A. Quiz Ontology

Oriented at massive exercise material from numerous courses to serve a large number of students in real-time, we first separate quiz question bodies from their question options and put them into a question pool and an option pool, respectively. For instance, a Multiple-Choice question has 5 question options, now in IFSE they are split and stored into these two pools. Moreover, this aims to reduce teachers' manual effort and eventually to provide much better, intelligent exercises automatically. Teachers can focus on teaching and need to create/update quiz questions or options inside the two pools once in a while, only.

Fig. 1 presents the Quiz Ontology with some sample instances. Four concepts, viz., *Quiz*, *Question*, *Option*, and *QuestionStudentAnswer*, are formally defined. *Quiz* consists of a number of *Questions*. *Question* has a series of properties, such as *hasTitle*, *hasBody*, *hasType*, *hasKeyKnowledgeConcepts*, *hasCompetenceLevel*, *hasDifficultyLevel*, *hasOptions*, and *answeredBy*. Every instance of *Quiz* or *Question* is only created on the fly for a specific individual student. *Question* is answered by *QuestionStudentAnswer*.

There are numerous question options in the option pool. Each option can semantically link to multiple questions. An *Option* contains an *OptionBody*, some *KeyKnowledgeConcept*, some *isOptionOf* and two optional properties (i.e., *correctFeedback* and *incorrectFeedback*). When appointing an option to a question, whether it is the correct answer to this question or not must be specified at the same time. Therefore, the *isOptionOf* property has a pair of subproperties, *questionID* and *isCorrectAnswer*.

The concept *QuestionStudentAnswer* is designed for tracking a student's accomplishments when answering quiz questions. For instance, the instance *qsa_1234* (see Fig. 1) is the learning record of the *learner_1234*, who selected the options *qo_2* and *qo_5* as answers to the question instance *q_1234* and scored 0.5 (since the correct answers were *qo_2* and *qo_4*). Therefore, this student's knowledge mastery regarding the knowledge concepts *C3* and *C5* are updated subsequently and accordingly (see Section IV).

The separation of question bodies and their options makes it possible to dynamically generate personal quizzes and questions. Since every option specifies the knowledge concepts which it is testing, it is not only easy to accurately locate errors but it also is the way we connect the exercise system with the knowledge domain. Connecting knowledge concepts directly to quiz options (instead of questions) makes it possible to adaptively provide accurate and specific knowledge-based feedback (see Section V). This quiz Ontology is used as the blueprint of the data structure in the implementation of IFSE.

B. Domain Knowledge

Just digitizing learning content and material is not enough to intelligently assist self-learning. Instead, formally and deeply mining the semantic between knowledge concepts and learning material is key. Therefore, we started with an example course in the domain of *Media Education*, namely *Module_x*, and used ontologies to manually model its knowledge base. As Fig. 2 shows, the knowledge graph of *Module_x* currently consists of 41 knowledge concepts, 168 individuals, 14 data properties, 26 object properties, 845 logical axioms, 248 declaration axioms, 715 assertions, and so on.

Several domain experts and tutors of this course carefully defined 41 core knowledge concepts (orange nodes of Fig. 2) based on the textbook named *StudyLetter_xxx*. There are 168 learning objects (blue nodes of Fig. 2) directly connecting to actual learning resources (pink nodes of Fig. 2). The mapping

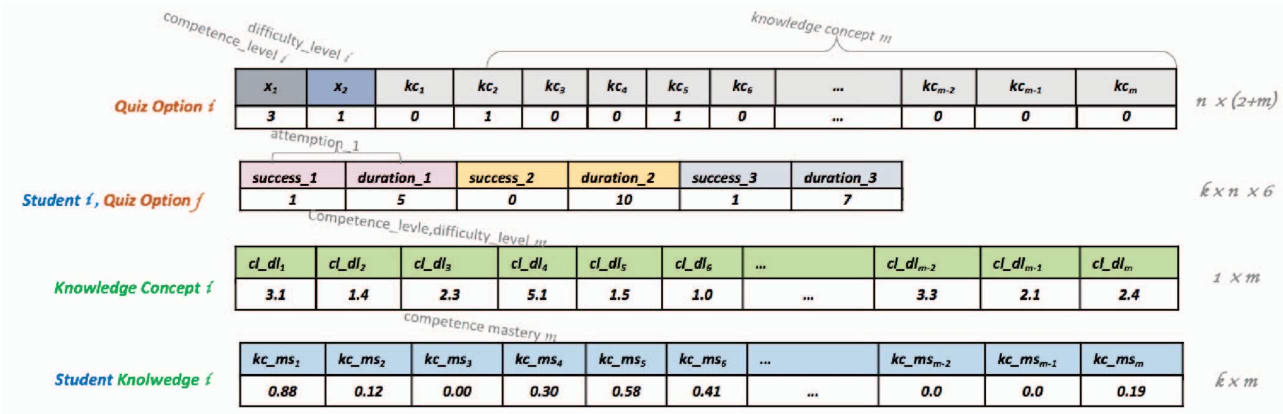

Fig. 3. Illustration of the collected datasets.

relationship between learning objects and learning resources is $n : m$. It means, a learning object may connect to several learning resources. The linked *resources* may have different types, e.g., *textbooks*, *reading articles*, *audios*, *videos*, *slides*, or *images*. The learning resources of course *Module_x* are currently all from its textbook. For example, in Fig. 2 the concept *Ziel-_und_Aufgabenbereiche_xxx* has an individual *Kompetenz* and it links to a resource element *Kompetenz*. The resource element *Kompetenz* specifies the learning content referring to the pages 79–80 in Chapter I.4 of *StudyLetter_xxx*.

Since learning materials are connected to quiz options, it becomes easy to locate where exactly students' knowledge weaknesses or mistakes are when they submit wrong quiz answers. Moreover, it is possible to provide accurate feedback to the wrong answers and recommend further learning content when they got correct answers.

IV. Personalized Quiz Generation

In essence, algorithms heavily depend on the data processed. The following data are collected from the current student learning testbed (see Fig. 3):

- ***Q***: option matrix of quiz questions. It is assumed that there are *n* options in the option pool.
- ***K***: knowledge concept vector with the specified knowledge difficulties. It is assumed that in the domain ontology *m* knowledge concepts and learning objects are defined.
- ***SK***: mastery matrix of students' knowledge competence. It is assumed that *k* students are involved in testing.
- ***SQ***: student learning cube of quizzes. For instance, entry SQ_{ij} tracks the learning records of an individual student *i* on the quiz option *j*, where $i \leq k$ and $j \leq n$.

Based on the matrices *Q*, *K* and the data cube *SQ*, the Item Response Theory (IRT) [3] and the Transferable Belief Model (TBM) [4] from our previous work are reused to generate, update, and propagate the personal competence mastery values of students [1], denoted as *SK*. Basically, every time a student took some quizzes, his/her results are taken as evidence to update his/her own knowledge mastery values, either increasing or decreasing them. These values are used to measure the degree of knowledge mastery of students, which lie in [0, 1].

Fig. 3 is an illustration of the collected datasets with some demonstrative data. For instance, the shape of the option matrix *Q* is *(n, (2+m))* and the i_{th} row of *Q* represents the option *i* with its features, i.e., the *competence_level* with value 3, the *difficulty_level* with value 1, the connected knowledge concepts as kc_2 and kc_5. Similarly, the vector *K* lists the combined *competence_level* and *difficulty_level* of all defined concepts, e.g., saying *CD_level* of the knowledge concept kc_5 is 1.5.

Following the same setting as Moodle, students are allowed to have 3 attempts on a quiz option. For example, the student *i* was successful on his/her first attempt on option *j* and took 5 seconds to answer this question. As students continue to take exercises, their mastery of knowledge constantly changes and is updated in real time. For instance, after a while, the mastery value of the student *i* on the knowledge concept kc_5 becomes 0.58.

The sizes of the data and data features could expand over time or grow on demand. Due to the popularity of online learning and the currently ongoing pandemic, the numbers of students and learning materials could easily surge by many or hundreds of thousands. Hence, we are systematically preparing for a massive computational workload. At this moment, our testbed is dimensioned for about 600 students, 360 quiz question options, and 1059 learning objects.

A. Personalized Quiz Algorithm (PQ)

In order to generate personalized quiz questions in a dynamic format, students can simply specify their own selecion criteria (further explained in Fig. 6), e.g., explicitly indicating a certain knowledge concepts and a competence level. Besides, a selector matrix of weights, *W*, is applied to tune the results.

Fig. 4 gives an example of the combination of multiple criteria for generating a personalized quiz. These criteria are, respectively,

- the quiz option competence level and the difficulty level,

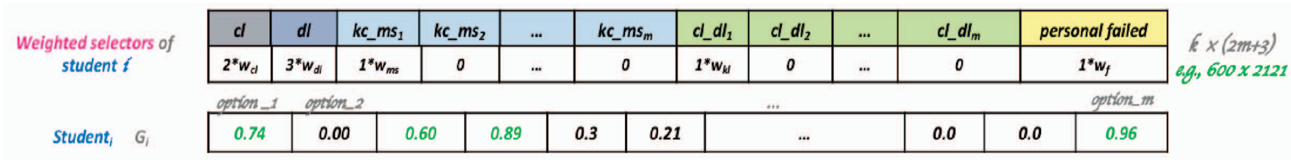

Fig. 4. Weighed selector matrix W and an example of the combined criteria (*above*); Example of winning quiz options (*below*)

- the specified knowledge concepts,
- the mastery value of knowledge concepts,
- the *CD_level* of knowledge concepts,
- the personal failed quiz options, e.g., whether considering the options for which a student failed or not.

Therefore, with the selectors given, a series of data transformations and normalizations are applied to Q and W. Taking the example of Fig. 4, one of the transformations observes two rules, viz., (1) selecting all quiz options with a list of knowledge concepts, such as kc_1, kc_3, and kc_{12}); and (2) selecting quiz options with competence level not less than 2.

Eventually, the current rule-based *PQ* algorithm is executed resulting in a matrix G,

$$G_{i,j} = F_j \times {R_j}^T, i < k, j < n \tag{1}$$

where F and R are the new matrices into which W and Q were transformed, respectively, and where the vector G_i represents the current states of all quiz options for the individual student i. Fig. 4 also gives an example of 4 final quiz options winners with a threshold of 0.6. The algorithmic pseudocode is presented in the Fig. 5.

Algorithm 1 Selects question options by filter

$results$ = empty set
for Every filter parameter p **do**
  set_p = select options from database by p
  Calculate $score_p$ for every element in set_p
  Remove the long tail of set_p
  Add all elements from set_p into $results$
end for
while $size(results) <$ required size **do**
  Add more related options into $results$
end while
for Every option $op \in results$ **do**
  $score(op) = \sum_{\text{parameter } p} score_p(op) * weight_p$
end for
Sort options in $results$ by $score$
Remove the long tail of $results$
return $results$

Fig. 5. Pseudocode of the Personalized Quiz Algorithm.

Based on the semantic relationships of quiz questions and options (defined by the quiz ontology), quiz questions can easily and automatically be formed, and the top s of them are returned as personalized quiz to student i.

B. Personal Quiz Filter

During self-regulated learning, students may want to test their knowledge from time to time and expect to get useful feedbacks instantly. Fig. 6 is the filter being used to capture students' requirements in order to generate proper quizzes adaptively. Students can set the knowledge coverage by selecting multiple courses or all of them in their learning programme, specify the number of questions, the competence and difficulty levels of question options. As we have a well defined knowledge domain, students can even select knowledge concepts from a dropdown menu.

Since we are recording students' learning, it is possible for students to set the percentage of the completely new and already learned knowledge in their quizzes to test or review their knowledge. There are also three smart switches. Switching on *adapted* means to automatically fill the whole filter with system-recommended values, which the students can still change by sliding bars. If *myFailed* is on, then some previously failed questions may appear again. And if *dynamic* is on, then quiz questions are generated one by one on the fly. The next question is adaptively created based on the previous question's result.

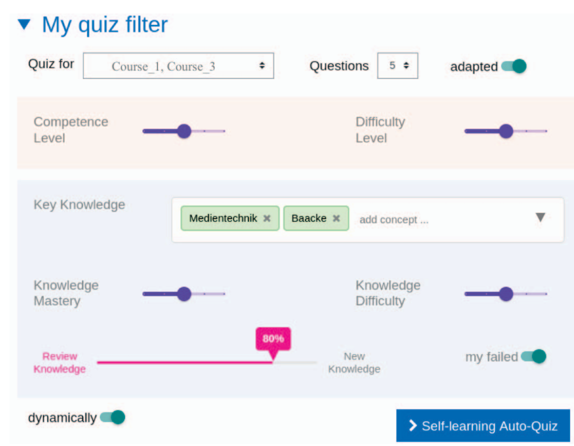

Fig. 6. The filter for generating personal quizzes.

V. Knowledge Recommendation Algorithms

Every quiz option connects to the knowledge concepts being tested. Thence, it is possible to accurately distinguish and exactly position the errors when students submit wrong answers. Currently, the knowledge recommendation algorithm (*KR*) consists of a set of query rules. Basically, IFSE provides individual instant feedback to both the selected incorrect question options and unselected correct options (see *rule1*), and also gives recommendations (instead of feedbacks) to correct answers (see *rule4*). Feedback is formed directly based on the linked resources. Students can visit the given resources immediately to either correct their knowledge or continue their study from here.

For generating accurate feedbacks, the complex relations among all kinds of elements have to be specified at first. Suppose a question option o connects to a set of knowledge concepts *oc*, where $oc = \{c_1, c_2, ..., c_i\}$; and each concept links to multiple resources re, where $c_i = \{re_1, re_2, ..., re_j\}, i, j \in \mathbb{N}$. Further suppose Re_{ALL} to be the resource set of all corrected selected options and Re_X the resource set of a wrongly selected or a missed option *X*. Now the following rules are defined:

- *rule1:* Generate feedback for either a wrong or missed option one by one, only.
- *rule2:* If $Re_X \neq \varnothing$ and $Re_X \cap Re_{ALL} = \varnothing$, then return Re_X.
- *rule3:* If $Re_X \neq \varnothing$ and $Re_X \cap Re_{ALL} \neq \varnothing$, then return $(Re_X - Re_{ALL})$.
- *rule4:* Generate recommendations for a fully correct answer, only.
- *rule5:* First, a recommendation is derived from the siblings of the existing knowledge concepts and, then, from their superclass concepts. Moreover, the knowledge concepts with low mastery value siblings or superclasses are set to be recommended first.

Taking the case presented in Fig. 7 as an example, supposing a student's answers to a question are the option *A)* and *B)*, unfortunately the correct answers are *A)* and *E)*, then following *rule1*, the feedback should be generated only on the options *B)* and *E)*. Further, supposing that $oc_A = \{c_5\}$, $oc_B = \{c_6\}$, and $oc_E = \{c_7\}$, where c_5 and c_6 share the same resource re_b. Then, the final feedback is generated for option $B)$ based on the resource re_a, and for option $E)$ with the resource re_c (which is quite similar to the case in Fig. 8).

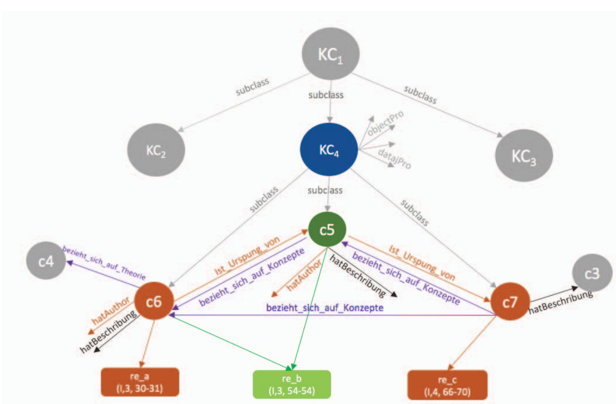

Fig. 7. Example case for demonstrating query rules.

VI. Intelligent Feedback to Student Exercises

The two algorithms, *PQ* and *KR*, have been implemented in our Intelligent Feedback to Student Exercise (IFSE) application, which is gradually being tested and released in the winter semester 2021/22 at FernUniversität in Hagen.

IFSE is an AI-based quiz subsystem designed for both teachers and students. Its objective is to effectively provide learners/students with adaptive quizzes/exercises and precise personal feedback according to their particular levels of knowledge during their independent and self-regulated learning. For this purpose, the levels of knowledge are first determined using various quiz formats based on a knowledge-based expert system. On this basis, the students receive learning recommendations for both content and cognitive learning strategies.

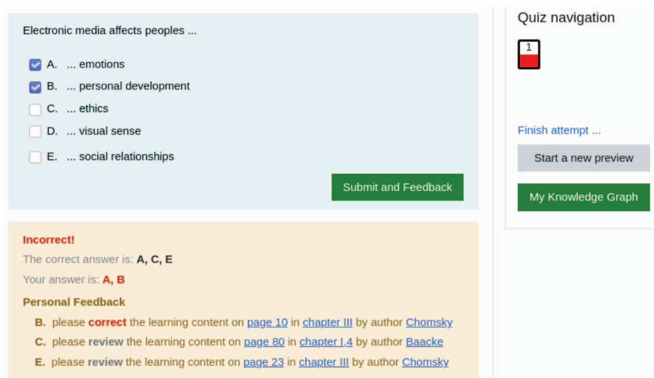

Fig. 8. Screenshot of IFSE application.

Up to the current stage, three major features were developed (see Fig. 8). First, personal quiz questions are generated for students and, based on the performance for the respective previously answered question, the next question is dynamically and intelligently adjusted in their difficulty levels and adapted to the students' knowledge competence levels detected. Secondly, instant individual feedback and knowledge recommendations to quiz question answers are automatically generated and delivered. The third function is that students can graphically overview their knowledge mastery at any time via a knowledge visualization (similar to the one used in [2]).

A. IFSE Implementation as a Moodle Plug-in

IFSE is planned to be a standalone application for generic purposes and versatile domains, which means that its core algorithms can be applied to any pluggable specific knowledge domain and student learning data. Since our current testbed is a Moodle learning system, we also implemented it as a Moodle plug-in.

Although Moodle itself has already a quiz module, IFSE's core (i.e., features, workflow, and user navigation) is totally different, and most of the existing functionalities cannot be reused directly. Therefore, we followed a standard-compliant approach to implement IFSE as a type of *Moodle Activity Module*, which can seamlessly and without any problem be integrated into any Moodle topic.

Thence, IFSE includes an entry point view for quiz creation, quiz navigation and attempts overview. These generated views closely resemble the Moodle quiz activity to reduce user confusion. The views' behaviour is strongly reinforced by asynchronous Javascript to reduce page reloads. This plug-in also includes an administration view for configuring backend connectivity parameters.

IFSE totally inherits all the standard Moodle question types, such as *True/False*, *Multi-choice*, *Single-choice*, *Gap Select*, *Matching*, *Image drag/drop*, and *Short Answer*. Moreover, IFSE's *immediate intelligent feedback* solution replaces the existing preset and fixed feedback.

Fig. 8 is a screenshot from the IFSE plug-in, supposing a student is answering a multiple-choice question. Unfortunately, this student fails. The correct answers are *A*, *C*, and *E*, but the student went for *A* and *B*. Therefore, the instant feedback suggests him/her to correct his/her knowledge on *B* and review again the knowledge on *C* and *E*. Moreover, if the student clicks the provided linked resource, the student can directly start his/her learning from there.

When this student chooses to take his/her quiz dynamically, then every time only one quiz question is generated for him/her on the fly. If the student answered wrongly, then his/her next question might be an easier one or a similar question (according to the system setting). This student also gets a chance to review his/her own knowledge graph to have an overview of the learning progress and how good he/she is mastering the knowledge.

B. Simulation and Evaluation

Since IFSE will be put into use in the winter semester 2021/22, some simulation experiments were designed to conduct a preliminary evaluation on the feasibility and efficiency of the quiz generation algorithm. Additionally, we try to gain as much insight as possible into the key parameters, and how they affect the efficiency of the algorithm.

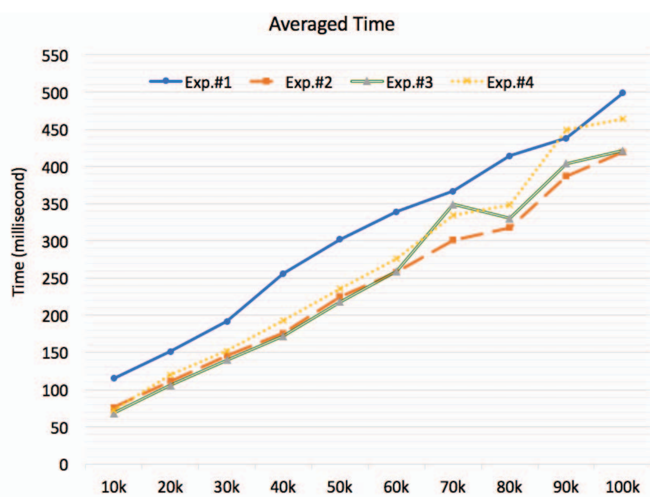

Fig. 9. Average computing time of the four experiments.

First, we assume that each question contains 6$\sim$10 options, and the number of one option to belong to multiple questions is temporarily kept around 10$\sim$15%. Our current domain knowledge for testing contains 305 key knowledge concepts (KCs). We then created two datasets, the small dataset with 1,000 students and 10K question options, and the big dataset with 10K students and 100K options. We carried out four experiments on the different combinations of four features, i.e., competence levels (*cLevel* combining difficult levels (*dLevel*)), coverage of knowledge concepts, mastery value, and *myFailed* (see Table I). We also simulated two different distributions (i.e., *Uniform* and *Normal* distribution) regarding the 25 categories of competencies (resulting from 5 competence levels with 5 difficult levels). When the number of question options is large enough, intuitively a tendency towards a uniform distribution is expected, but a normal distribution turned out to be much closer to the actual situation (with $\mu = 0$, $\sigma = 5$) .

Moreover, for example, we ran the first experiment altogether 2250 times (randomly selecting 3 students on 25 C/D levels and repeating about 30 times) and recorded the minimum, maximum, and average times of quiz generation. The results shown in Table I give rise to the following findings:

- It takes less than 2 sec to generate a quiz with 30 questions for a student both on the small dataset and the big one.
- When a certain knowledge concepts is specified, the time (of the query contained query and the generation) drops by nearly 40% (i.e., from 400+ to 200+ msec).
- When more parameters are considered, less calculation time is required (see Fig. 9). The two features *mastery value* and *myFailed* have no significant effect on the results.
- Since mainly querying a MongoDB, basically the time complexity of this algorithm is *O(log(n))*. Comparing the two datasets reveals that the more question options there are the more calculation time is needed.

So far we are quite satisfied with the results of our simulation experiments. They demonstrate our algorithm to work quite efficiently, and to operate well with even relatively large datasets. For deeper functional tests and students' satisfaction with the generated quizzes, feedbacks, and knowledge recommendations, we will conduct detailed evaluations after collecting a large amount of usage data after the semester.

VII. Conclusion and Future Work

An AI-based quiz subsystem with a completely new design for quizzes was proposed in this paper, which separates question bodies and their options according to the semantics formally defined by a quiz ontology. Two novel algorithms for quiz generation and knowledge recommendation were applied in order to adaptively generate personalized quizzes for individual learners. The intelligent instant feedback to students' quiz results aims to provide accurate knowledge feedback after locating students' incorrect knowledge concepts. This IFSE quiz application was developed as a Moodle plug-in and is ready for testing. The simulation results so far are quite promising.

TABLE I
GENERATING A QUIZ WITH 30 QUESTIONS.

Parameters	**Averaged Time (ms)**				**Min Time (ms)**				**Max Time(ms)**				**#Exp.**	**Exp. Description**
	small data		**big data**		**small data**		**big data**		**small data**		**big data**			
	Uni.	Nor.	Uni.	Nor.	Uni.	Nor.	Uni.	Nor.	Uni.	Nor.	Uni.	Nor.		
C/D Levels	115	**118**	499	**447**	64	62	349	319	568	674	1111	922	2250	**25 combination of c/d levels** * 3 random learners * 30 repeats
Key KCs, C/D Levels	76	**71**	419	**281**	62	56	337	225	301	228	**1735**	1369	2745	**305 knowledge concepts** * 3 c/d levels (i.e., (1,1),(3,3),(5,5)) * 3 random learners
Key KCs, C/D Levels, Mastery Value	68	77	421	395	58	56	292	288	113	146	708	659	2400	**20 mastery value** (.05, .10, ...,1.0) * 3 random learners * 40 random knowledge concepts * cLevel = 3, dLevel = 1
key KCs, C/D Levels, Mastery Value, MyFailed	**72**	**79**	**464**	**409**	62	62	374	253	103	128	825	1695	2400	***myFailed* = true** * 40 random knowledge concepts * 3 random learners * 20 mastery value (0.1, 0.4, 0.6, 0.8) * cLevel = 4, dLevel = 2

In the future, IFSE is to incorporate more natural language processing (NLP) featured functions, for example, that students will be allowed to input some plain text to describe their wishes on what knowledge to be tested. On the other side, using NLP to process the imported learning materials, teachers will be able to be assisted with automatically generated questions or options as suggestions, which are all fact-/knowledge-based aiming for a given domain's basic knowledge.

Furthermore, as a new learning environment, IFSE is able to collect the new types of students' learning information for more profound machine learning-based analysis, such as the time needed to answer questions requiring specific knowledge, the duration of checking feedback, the number of hits of recommended resource links and so on; besides, the weights W will be tuned and optimized. Pattern and sequence of learning certain content are also our major objectives in the future.

REFERENCES

[1] A. Faulhaber and E. Melis, "An Efficient Student Model Based on Student Performance and Metadata". the 18th European Conference on Artificial Intelligence (ECAI), vol. 178, pp. 276–280, 2008.

[2] X. Wang, T. Gülenman, N. Pinkwart, C. de Witt, C. Gloerfeld and S. E. Wrede, "Automatic Assessment of Student Homework and Personalized Recommendation". IEEE 20th International Conference on Advanced Learning Technologies (ICALT), pp. 150–54, July, Estonia, 2020.

[3] F. M. Lord, "Applications of item response theory to practical testing problems". Hillsdale, NJ: Erlbaum, 1980.

[4] P. Smets and R. Kennes, "The transferable belief model", Artificial Intelligence, vol. 66, pp.191–234, 1994.

[5] J. T. Burman, C. D. Green and S. Shanker, "On the Meanings of Self-Regulation: Digital Humanities in Service of Conceptual Clarity". Child Development 86 (5): 1507–1521, 2015.

[6] J. G. Greeno and C. van de Sande, "Perspectival understanding of conceptions and conceptual growth in interaction". Educ. Psychol. 42, 9–23, 2007.

[7] D. L. Butler and P. H. Winne, "Feedback and Self-Regulated Learning: A Theoretical Synthesis". Review of Educational Research 65, 245–281, 1995.

[8] E. Panadero, "A review of self-regulated learning: Six models and four directions for research". Frontiers in Psychology 8, Article 422, 2017.

[9] P. H. Winne, "A metacognitive view of individual differences in self-regulated learning". Learn. Individ. Differ. 8, pp. 327–353,1996.

[10] P. H. Winne and A. F. Hadwin, "Studying as self-regulated engagement in learning". in Metacognition in Educational Theory and Practice, eds. D. Hacker, J. Dunlosky and A. Graesser. Hillsdale, NJ: Erlbaum, pp. 277–304, 1998.

[11] A. Alkhatlan and J. Kalita, "Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments". International Journal of Computer Applications 181 (43): 1–20, 2019.

[12] R. S. Jansen, A. van Leeuwen, J. Janssen, R. Conijn and L. Kester, "Supporting learners' self-regulated learning in Massive Open Online Courses". Computers & Education vol. 146, 2020.

[13] O. Viberg, M. Khalil and M. Baars, "Self-Regulated Learning and Learning Analytics in Online Learning Environments: A Review of Empirical Research". the 10th International Learning Analytics and Knowledge Conference (LAK), 2020.

[14] Q. Zhang, J. Lu and G. Q. Zhang, "Recommender Systems in E-learning". J. Smart Environ Green Comput vol. 1, pp. 76–89, 2020.

[15] S. S. Khanal, P. W. Prasad, A. Alsadoon and A. Maag, "A systematic review: machine learning based recommendation systems for e-learning". Education and Information Technologies 25 (4), 2635–2664, 2020.

[16] S. Zhang, L. Yao, A. Sun and Y. Tay. "Deep learning based recommender system: A survey and new perspectives". ACM Computing Surveys 52 (1), 1–38, 2019.

[17] G. Kumar and A. Wang, "Distractor Re-ranking for Automatic Quiz Generation". Preprint, Stanford report, 2019.

[18] Z. P. Qiu, X. Wu and W. Fan, "Automatic Distractor Generation for Multiple Choice Questions in Standard Tests". Proc. 28th International Conference on Computational Linguistics, pp. 2096–2106, 2020.

[19] V. Gupta, H. Pant, A. Chaurasia, A. Dwivedi, S. Singh and P. Singh, "Automatic Quiz Generator". International Journal of Research in Engineering, Science and Management, 4 (7), 54–56, 2021.

[20] A. Bongir, V. Attar and R. Janardhanan, "Automated quiz generator". International Symposium on Intelligent Systems Technologies and Applications, pp. 174–188, Cham: Springer 2017.

[21] M. E. Venitha, "Automatic Quiz Generator Using Deep Learning". International Journal of Advanced Engineering Science and Information Technology, 4 (4), 2021.

[22] M. Grenander, R. Belfer, E. Kochmar, I.V. Serban, F. St-Hilaire, and J. C. K. Cheung, "Deep Discourse Analysis for Generating Personalized Feedback in Intelligent Tutor Systems", The 11th Symposium on Educational Advances in Artificial Intelligence (EAAI-21), Feb 2021.

[23] P. Dasic, J. Dasic, B. Crvenkovic, and V. Serifi, "A review of intelligent tutoring systems in e-learning". Annals of the Oradea University – Fascicle of Management and Technological Engineering 15, 85–90, 2016.

[24] J. Paladines and J. Ramirez, "A Systematic Literature Review of Intelligent Tutoring Systems With Dialogue in Natural Language". IEEE Access 8, 164246–164267, 2020.

[25] D. Dermeval, J. Albuquerque, I. Bittencourt, S. Isotani, A. Silva and J. Vassileva, GaTO: An Ontological Model to Apply Gamification in Intelligent Tutoring Systems, Frontiers in Artificial Intelligence 2, 10.3389/frai.2019.00013, 2019.

[26] M. Chang, G. D'Aniello, M. Gaeta, F. Orciuoli, D. Samspon and C. Simonelli, "Building Ontology-Driven Tutoring Models for Intelligent Tutoring Systems Using Data Mining". IEEE Access 8, 48151–48162, 2020.