applied sciences

MDPI

Article

An Ontology-Driven Learning Assessment Using the Script Concordance Test

Maja Radovic [1,*], Nenad Petrovic [2] and Milorad Tosic [2]

[1] Faculty of Technical Sciences Cacak, University of Kragujevac, Svetog Save 65, 32000 Cacak, Serbia
[2] Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, 18000 Nis, Serbia; nenad.petrovic@elfak.ni.ac.rs (N.P.); milorad.tosic@elfak.ni.ac.rs (M.T.)
* Correspondence: maja.radovic@ftn.kg.ac.rs; Tel.: +381-648526196

Abstract: Assessing the level of domain-specific reasoning acquired by students is one of the major challenges in education particularly in medical education. Considering the importance of clinical reasoning in preclinical and clinical practice, it is necessary to evaluate students' learning achievements accordingly. The traditional way of assessing clinical reasoning includes long-case exams, oral exams, and objective structured clinical examinations. However, the traditional assessment techniques are not enough to answer emerging requirements in the new reality due to limited scalability and difficulty for adoption in online education. In recent decades, the script concordance test (SCT) has emerged as a promising tool for assessment, particularly in medical education. The question is whether the usability of SCT could be raised to a level high enough to match the current education requirements by exploiting opportunities that new technologies provide, particularly semantic knowledge graphs (SCGs) and ontologies. In this paper, an ontology-driven learning assessment is proposed using a novel automated SCT generation platform. SCTonto ontology is adopted for knowledge representation in SCT question generation with the focus on using electronic health records data for medical education. Direct and indirect strategies for generating Likert-type scores of SCT are described in detail as well. The proposed automatic question generation was evaluated against the traditional manually created SCT, and the results showed that the time required for tests creation significantly reduced, which confirms significant scalability improvements with respect to traditional approaches.

Keywords: ontology; learning assessment platform; script concordance test

Citation: Radovic, M.; Petrovic, N.; Tosic, M. An Ontology-Driven Learning Assessment Using the Script Concordance Test. *Appl. Sci.* **2022**, *12*, 1472. https://doi.org/10.3390/app12031472

Academic Editor: Arcangelo Castiglione

Received: 30 December 2021
Accepted: 26 January 2022
Published: 29 January 2022

1. Introduction

The main aim of medical education is to prepare future health professionals for making an effective diagnostic and therapeutic decision in critical situations under time pressure while under the condition of uncertain information [1]. This complex process is known as "clinical reasoning", widely recognized as the essential element in physician practice [2]. It is much more than a simple application of knowledge, rules, and principles. In each individual case, physicians use clinical reasoning skills to gather patient data, after which a small set of pertinent illness scripts are activated [3]. Illness scripts are bounded networks of medical knowledge that allow physicians to integrate new incoming information with existing ones, recognize patterns in symptom complexes, identify similarities or differences between diseases, and make predictions about how presented diseases are likely to unfold [4].

In addition to evaluating theoretical knowledge, assessment in medical education is accounted to evaluate clinical reasoning [5]. In medical schools, clinical reasoning competency is assessed with few traditional standardized tools, such as long-case oral exams and objective structured clinical examinations (OSCEs). However, these tools are often resource intensive, time consuming, cumbersome to administer or score, or difficult to standardize [6].

On the other hand, other assessment tools, such as written examinations, i.e., multiple-choice questions (MCQs) and extended matching questions (EMQs) repeatedly show the fact that experienced physicians had scores hardly better than less experienced physicians or students [7]. This could be explained by the fact that MCQ and EMQ are tools more appropriate for assessing factual knowledge, which is less complex for evaluation than clinical reasoning [8].

The script concordance test (SCT) was introduced by Charlin et al. [9] in 2000, in an attempt to address formerly described problems. Its aim is to assess students' ability to interpret clinical data under the conditions of vagueness that correspond to reasoning in a realistic clinical setting. The SCT introduces students to patient vignettes that lack some information, after which three independent pieces of additional clinical information are given. Students are expected to make decisions on the diagnosis, investigation, or treatment for each of the three pieces of information offered, including answering three questions on a five-point Likert scale [10].

Reliability and validity of SCT has been confirmed in many medical disciplines such as general practice [10], urology [7], plastic surgery [11], neurology [12], acute abdomen [1], pediatric [13], palliative care [14], ear, nose, and throat (ENT) discipline [15], etc. Nevertheless, there are authors who argue that SCT cannot be an alternative to long case exams but should rather be concerned as an additional assessment tool [2,16]. Using a combination of real patients' data in real clinical settings and computer-based case scenarios would present a more valid and reliable way of assessing clinical reasoning and clinical competence. However, in light of the new events concerning the COVID-19 pandemic, this strategy would be difficult to apply [17]. Minimization of the level of personal interactions between students and patients has urged medical schools to speed up the transition from "real-life" clinical experiences to online learning [18].

Although online tools facilitate automation of the evaluation and administration process in SCT, constructing a patient vignette is still a time-consuming task that requires the engagement of domain experts. On the other hand, a large amount of clinical knowledge and electronic health record (EHR) data have been accumulated in medical institutions. Physicians' experiences in diagnosis, management, and treatment are also hidden in the medical records data [19]. Furthermore, IoT devices that monitor patients' body conditions, collect data, which can be used to make EHRs more detailed [20]. Therefore, there is a significant need to incorporate EHR in academic training settings [21].

The adoption of ontologies and semantic knowledge graphs (SKGs) for knowledge representation has become the cornerstone technology [22] that could enable SCT to advance further. Ontologies have been used for over a decade, for automatic question generation, and its successful application is presented in a number of research papers [23–26], etc.

On another front, automated question generation (AQG) plays a major role in educational assessment nowadays. It reduces the time and expenses needed for the manual construction of questions and also produces a continuous supply of new questions [27]. Therefore, building systems for generating questions has become imperative for researchers all over the world [28]. Although there are a plethora of research papers regarding AQG, we focus our literary review on ontology-based systems. In what follows, we give an overview of relevant ontology-based question generation solutions from the available literature.

Litherland et al. developed the Ontology eLearning (OeLe) E-Assessment platform, which generates open questions for free text answers. In the developed OeLe platform, authors used ontologies, semantic annotations, and similarity functions, as well as natural language processing techniques. However, ontology is created externally, and this process may seem discouraging for non-IT-savvy teachers [29]. OntoQue is an ontology-based system for MCQ item generation presented by Al-Yahya [27]. Although it includes true/false, fill-in, and multiple-choice items, the study was focused on MCQs solely. The proposed system generates items by iterating over all entities in the particular ontology. Although MCQs are satisfactory to a certain extent, the evaluation shows that the majority of MCQs also concentrate only on factual knowledge [30]. Fattoh developed an automatic question

generation (AQG) system that selects an informative sentence and keywords for a question based on the semantic labels and names of entities in the sentence [31]. The system chooses distractors through the application of string similarity measured between sentences in a dataset. This research was also limited to MCQ questions.

In one of their latest studies, Vinu and Kumar [32] elaborated on the details of their prototype system called extended automatic test generation (E-ATG) used for MCQs generation. E-ATG can generate MCQ sets of particular sizes and find the difficulty values. It also controls the overall difficulty level of MCQ sets. The evaluation that the authors conducted shows that the system proposed can generate domain-specific MCQ sets, which are close to the one generated by domain experts regarding semantic similarity [3]. Ontology-based personalized feedback generator (OntoPeFeGe) framework was proposed by Demaidi et al. OntoPeFeGe consists of two components that generate true/false, multiple-choice, and short-answer questions, with five different types of feedback and the personalized feedback algorithm that provide students with appropriate feedback after answering the questions. However, teachers are limited to the above-mentioned type of questions [25]. OntoQuest, a framework for the generation of multiple-choice questions, was presented by Deepak et al. [33]. In order to determine relevant sub-topics and auxiliary topics, domain and granular ontologies were used. OntoQuest uses a strategy for e-assessment by generating MCQ from various crawled web corpora. Research has confirmed the reliability of the OntoQuest framework and states that its accuracy in key and distractor generation is higher than the existing models [33].

Santhanavijayan and Balasundaram proposed fuzzy-MCS-algorithm-based ontology generation for e-assessment, in which MCQs are generated from a given ontology [34]. Java was used as a working platform for the implementation of the proposed ontology for e-assessment. MCQs are generated using ontologies, and the assessment is made based on the answers obtained from the attending candidates. Their results show that, although the system they proposed is very simple, it provides a better percentage of correct answers than the existing optimization algorithms [34].

A bilingual ontology-based automatic question generation system was proposed by Diatta, Basse, and Ouya [35]. It is designed to help learners to generate questions for self-evaluation on laboratory materials concerning product and security rules. MCQ and true/false questions are generated on the fly and distracters change in each execution, providing for the same question a different content. Classes, properties, and individuals are used as inputs to generate questions. The authors also developed a web application that has a user-friendly interface to generate questions on products and materials used in lab works. In the backend is an information querying module that uses SPARQL to query data from ontology [35].

A modular system called the EMMeT multiple-choice question generator (EMCQG) was introduced by Leo et al. [36]. EMCQG is based on The Elsevier Merged Medical Taxonomy (EMMeT) database. It generates medical MCQ questions whose stem is in the form of patient vignettes. This type of question is standard in medical education because of its ability to evaluate clinical reasoning. However, EMCQG is not open source and, thus, not available for public review [36].

In our previous study [37], we developed an ontology called SCTonto, with the goal of automated question generation for the SCT assessment method. SCTonto proved suitable for the purpose. However, a methodology for the development of an ontology-based platform for learning assessment based on SCT was not considered.

The main contributions of this paper are the following:

- A methodology for an ontology-driven learning assessment is proposed and proven in the case of script concordance tests;
- SCTonto ontology, developed in our previous study, is enhanced and confirmed usable in the context of the presented methodology;
- A novel ontology-driven automated script-concordance-test-based assessment platform is proposed;

- The proposed platform is evaluated against the traditional manually created SCT;
- Presented experimental results indicate the significant reduction in tests creation time, confirming significant improvements in scalability.

The remainder of the paper is structured as follows: Section 2 presents the core of this paper with a detailed structure of ontology-driven framework for automated SCT question generation. Section 3 gives results of the evaluation of the traditional construction of SCT against the ontology-based question generation. Discussion about obtained results is given in Section 4. Finally, Section 5 summarizes the main ideas of this paper and outlines the following technical steps in the evaluation of our approach.

2. Methods and Materials

2.1. Script Concordance Test

Script Concordance test is an assessment tool used in measuring assessing reasoning under conditions of uncertainty [8]. The construction of SCT is based on the principles and characteristics of script theory, which states that networks of knowledge, called "illness scripts", begin to form during the physician's first encounter with the patient and become refined with experience. In other words, each time a physician meets a new patient with incoming data (symptoms, signs, laboratory data, etc.), illness scripts enable the selection and interpretation of these data. Through time, evolved illness scripts allow medical experts to make accurate decisions promptly, efficiently, and often with minimal conscious effort [6].

In its traditional written form, the construction of SCT (Table 1) involves two or more experienced physicians who write patient vignettes. These vignettes or clinical scenarios contain a certain amount of uncertainty, in order to simulate the ambiguous conditions that often occur in real life. Vignette is then followed by three mutually independent hypotheses in the form of a diagnostic possibility, an investigative option, or a therapeutic alternative [6]. It is important to note that the hypotheses must all be plausible (i.e., students should feel that the hypotheses are, indeed, reasonable considerations in the context of the given patient vignette) [6]. Each hypothesis is further followed by new information, such as a physical examination sign, an imaging study, laboratory test result, etc. This new information may or may not be relevant for the given hypothesis. The impact of new information on a given hypothesis is captured through a five-point Likert-type scale.

Table 1. Example of SCT.

Case Description: You Are Evaluating a 35-Year-Old Woman with a Sore Throat, Difficulty Swallowing, and Coughing for 10 Days.		
If you were thinking . . . :	Additionally, then you find . . . :	Your hypothesis becomes . . . :
Q1. Acute bronchitis	Bronchovesicular breathing without accompanying sounds; lymph nodes of the neck—not palpably enlarged; throat- hyperemia, enlarged tonsils	−2 −1 0 +1 +2
Q2. Acute pharyngitis	lymph nodes of the neck—not palpably enlarged; throat- hyperemia, enlarged tonsils	−2 −1 0 +1 +2
Q3. Acute rhinopharyngitis	lymph nodes of the neck—not palpably enlarged; throat- mild hyperemia of the throat	−2 −1 0 +1 +2

−2: Ruled out or almost ruled out; −1: Less likely; 0: Neither more nor less likely; +1: More likely; +2: Certain or almost certain.

When the SCT is complete, it is presented to the reference panel of experienced practitioners. Research study shows that the optimal number of experts is 15 [5]. After the reference panel, SCT is presented to students who also make judgments about the impact of new information on a given hypothesis. Each answer can be further measured and compared to those of a reference panel. There are several scoring methods and the aggregation method seems to be mostly used [1]. Here, the credits for each question are

derived from the answers given by the panel of experts and divided by the number of panel members. Scores for each question are added up and divided by the total number of questions and divided by 100 to give a percentage score [2]. Several research studies across different medical disciplines support the SCT's construct validity, reliability, and feasibility across a variety of health science disciplines [6]. To achieve the best score reliability, SCT should include about 25 cases, with 3 hypotheses nested within each question, and testing time should be 60–90 min [4].

2.2. SCTonto

For the development of SCTonto ontology, we adopted the SABiO 2.0 process [38]. Figure 1 illustrates the methodology we followed and the workflow adopted for the development of the SCTonto. The main purpose of SCTonto ontology is the ability to support semantic annotations of the script concordance test assessment method. In other words, it will serve as a framework for an ontology-based e-assessment platform for automatic SCT question generation. The main groups that benefit from the proposed ontology are course administrators and teachers, who will be able to quickly and conveniently generate appropriate questions.

Functional and non-functional requirements were important in the first phase of the development. Functional requirements were stated in the form of competency questions that help developers to determine what is relevant and what is not, thus defining the scope of the ontology [39]. Some of the competency questions regarding SCTonto are "Can each question have more than one case description?", "How many hypotheses can each question have?", "Does every new information item describe exactly one hypothesis item?", "How many possible effects can one new information have on the hypothesis?", etc. Aside from functional parameters, the non-functional requirements were defined as well. They state that an ontology-based system should generate SCT type of questions for student assessments, and a SPARQL reasoner should be used. In our case, we wanted to keep the ontology simple, so we constrained the ontology to be implemented in the RDF language.

As an application ontology, SCTonto should be complemented with a domain ontology. It could be medical ontology or electronic health record (EHR) ontology since they define foundations for most of the main concepts in SCTonto ontology, such as symptoms and signs (case description in SCT), diagnosis (hypothesis in SCT), laboratory and other analysis (new information in SCT), etc. Due to patient privacy issues, populated EHR ontologies are difficult to obtain in the public domain. Hence, for the purpose of this research, we decided to map the medical records database [40] into SCT ontology instead. The detailed process of the mapping is described in the next section.

After ontology type definition, concepts and relations between them were identified as well. A detailed description of conducted analysis and defining of SCTonto concepts and their properties are described in our previous study [27]. Here, we give a brief illustration of the main classes and properties. Sct:Question, sct:CaseDescription, sct:Hypothesis, sct:NewInformation, and sct:Response are the main classes. Since each SCT question consists of one case description, several hypotheses, several new information, and several responses, this was modeled with properties sct:hasCaseDescription, sct:hasHypothesis, sct:hasNewInformation, and sct:hasResponse. The relationship between the instance of sct:CaseDescription class and the sct:NewInformation class was modeled with property sct:hasRelevant since it emphasized that case description is an ill-defined patient vignette in which some part of the information is missing. The sct:hasPossibleEffect and sct:isPossibleEffectedBy are properties that represent the fact that new information may or may not have the effect on the proposed hypothesis and vice versa. Graphical representation of main SCT concepts and relationships, performed in the Graffoo tool [41], is presented in Figure 2. The fact that students grade each hypothesis by selecting an appropriate number on the Likert scale is modeled through two properties: sct:isGradedBy and sct: grades.

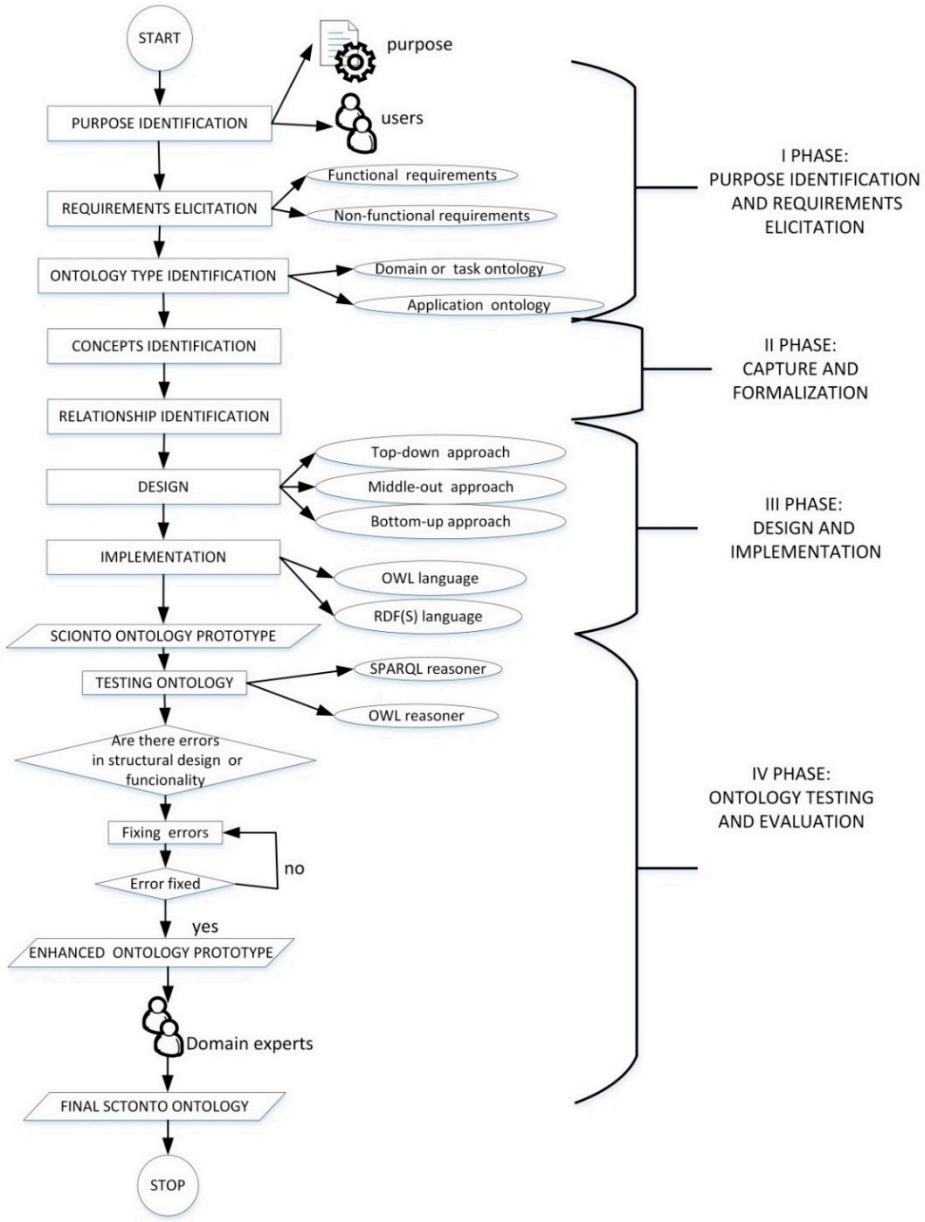

Figure 1. Workflow for the development of the SCTonto ontology.

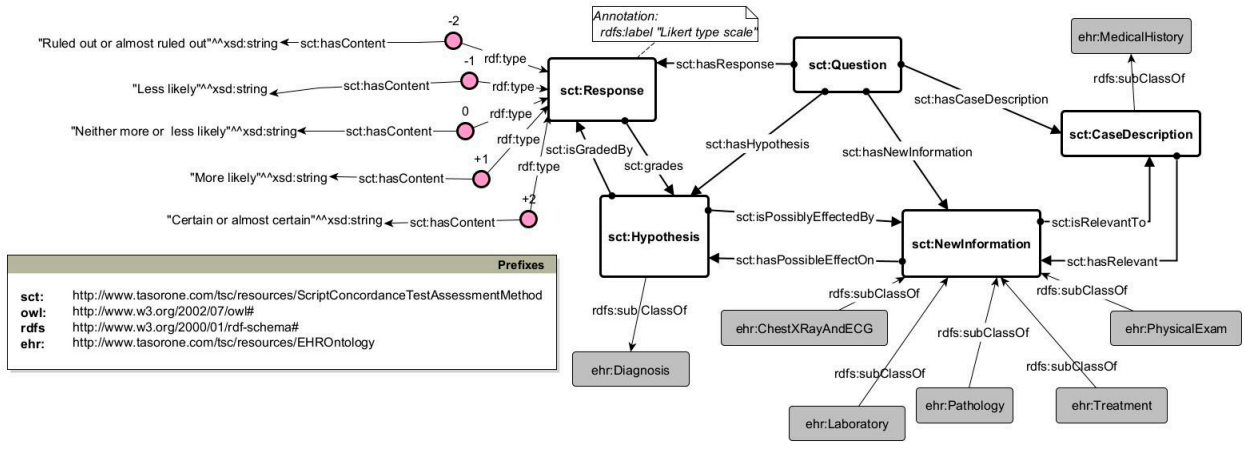

Figure 2. SCTonto ontology.

During the design phase, a middle-out approach was applied since it strikes a balance between levels of details [42]. The most important concepts were defined first and then followed by the higher-level concepts, thus creating them to be presumably stable. TasorOne online editor [43] was chosen for ontology implementation, and the full description of the class sct:Question and the related entities was provided in the RDF implementation file [44].

The final phase of ontology testing and evaluation was conducted through two phases. In the first phase, SCTonto ontology was tested through several SPARQL queries. They were used in order to check ontology behavior on a finite set of test cases, against the expected behavior regarding the competency questions. Listing 1 is the example of testing regarding competency question" Does every new information item describe exactly one hypothesis item?" This query checks if the single hypothesis rule is broken. An ASK type of SPARQL query was used that returns TRUE if the query body returns a result. In the second phase, the ontology was evaluated against the traditional manually created SCT. Section 5 presents obtained results that confirm significant scalability improvements with respect to traditional approaches.

Listing 1. SPARQL query for ontology testing.

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntaxns#>
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sct:
 <http://www.tasorone.com/tsc/resources/ScriptCorcondanceTestAssessmentMethod/>
# single_hypothesis_rule:  returns true if broken
ASK {
    FILTER (?number_of_hypotheses>1)
           {
            SELECT ?q (count(?s) as ?number_of_hypotheses)
                 WHERE {
                        ?s rdf:type sct:Hypothesis.
                        ?q rdf:type sct:NewInformation.
                        ?q sct:hasPossibleEffectOn ?s.
                        }
                 GROUP BY ?q
             }
       }
```

2.3. Proposed Framework

The architecture of the proposed ontology-driven automated script concordance test generation framework is presented in Figure 3. The framework relies on ontology mapping and code generation algorithms that leverage semantic annotations based on ontologies.

First, the mapping between the medical records database and question ontology was performed. In the data preparation phase, a query that retrieves only information relevant to the procedure of SCT question generation was executed (Listing 2). Hypotheses corresponding to diagnostic possibility were selected, as well as information part of the question that corresponds to laboratory results descriptions from the health records database. The average execution time of this query was around 1 s using the data.world [45] online service. Finally, the query results were downloaded from the data.world cloud data catalog and stored in .CSV format on our server.

Listing 2. Query for retrieving information relevant for SCT question generation.

```
SELECT encounter.soap_note, encounter_dx.description,
lab_results.result_description FROM encounter, encounter_dx,
lab_results WHERE
encounter.encounter_id=encounter_dx.encounter_id AND
lab_results.encounter_id=encounter.encounter_id
```

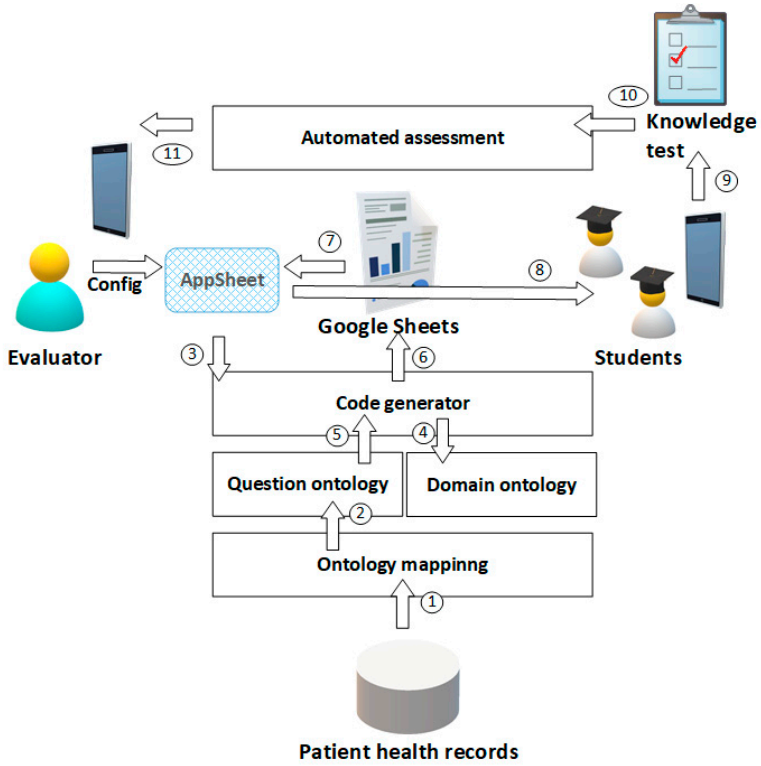

Figure 3. Ontology-driven framework for automated SCT question generation: 1, 2—mapping of patient health records DB to SCT question ontology; 3—question topic and difficulty settings; 4—SPARQL queries; 5—query results; 6—code generation results; 7—rendering mobile app starting from Google Sheets data; 8—quizzes within mobile app; 9—answering the questions; 10—question responses; 11—student assessment results.

The former query returned 833 results that were then parsed and semantically annotated with respect to mapping given in Table 2.

Table 2. Mapping between medical records database and SCT ontology.

Medical Record Database Column	Ontology Class
encounter.soap_note	CaseDescription
encounter_dx.description	Hypothesis
lab_results.result_description	NewInformation

However, due to the way the SCT questions were constructed [6], it was not possible to retrieve Likert-scale response scores directly from the database. Generating Likert scale scores for answers to SCT questions is a knowledge-intensive task that is traditionally performed by domain experts. In this paper, two possible strategies to cover this aspect are proposed: (1) Direct strategy is based on a direct selection of healthcare expert-approved lab results for a given hypothesis. The number of cases for each of the results is summarized, and Likert scores are assigned with respect to the number of experts that agree on the hypothesis used for obtaining these results. This type of question is simpler for code generation but is considered quite difficult for students, due to fact that precise knowledge is needed to select the most appropriate answer. (2) Indirect strategy, on the other hand, selects the lab results for a disease that is closely or distantly related to the one provided by an evaluator. The difficulty level is considered higher when the new information is derived from lab results of a closely related disease (diabetes type 1 and type 2, for example), while it is considered easier when the diseases are not much related (e.g., infraction and diabetes type 1). For this

purpose, a domain ontology about the hierarchy and relations between diseases, such as Disease Ontology (DO) [46], should be adopted to provide the necessary knowledge.

In the direct strategy, processing and calculation of the extracted data are performed for each hypothesis that is identified among the results. First, all of the possible result descriptions (NewInformation) for a given disease description (Hypothesis) are identified. After that, the number of medical records that exist for each of the possible result descriptions for the given disease is determined by simple counting. The probability of each particular result is then determined by dividing the number of records by the total number of cases for that hypothesis. In order to adapt it to the Likert scale, the resulting probabilities are classified into five ranges of 0.0–0.2, 0.2–0.4, 0.4–0.6, 0.6–0.8, and 0.8–1.0, where each range corresponds to one of the scores of -2, -1, 0, 1, and 2, accordingly. Finally, the Likert-scale response score is calculated based on the probability in the following way. If the probability is between 80 and 100%, answer 2 is assigned a maximum score, while the other options are assigned lesser scores:—1 will be assigned a score of $4/5$, 0 will be assigned a score of $3/5$, -1 is assigned a score of $2/5$, and -2 is assigned a score of $1/5$. If the probability is between 60% and 80%, then 1 is assigned the maximum score, 2 is assigned a score of $4/5$, 0 is assigned a score of $3/5$, etc. The pseudo-code of the Likert-scale-score assignment algorithm for the direct strategy is given in Listing 3.

Listing 3. Pseudo-code of Likert-sale-score assignment algorithm for direct strategy.

```
Input: health_records_db, disease_hypothesis
Output: Likert scale scores
Steps:

1.  Result_descriptions: = GetAllResultDescriptionsForGivenHypothesis(disease_hypothesis,
    health_records_db);
2.  total_records_num: = CountRecords(disease_hypothesis);
3.  For each description in result_descriptions
4.      new_information_records_num: = CountRecords(disease_hypothesis, description);
5.      probability: = new_information_records_num/total_records_num;
6.      If(probability > 0.8)
7.          LikertPlus2: = probability
8.          LikertPlus1: = probability * 0.8;
9.          Likert0: = probability * 0.6;
10.         LikertMinus1: = probability * 0.4;
11.         LikertMinus2: = probability * 0.2;
12.   else if(probability > 0.6)
13.         LikertPlus2: = probability * 0.8;
14.         LikertPlus1: = probability;
15.         Likert0: = probability * 0.6;
16.         LikertMinus1: = probability * 0.4;
17.         LikertMinus2: = probability * 0.2;
18.  . . . .
19.      Endif;
20. Endfor;
21. End.
```

In the indirect strategy, the user first defines desired difficulty level that is used to select another disease hypothesis. If a "hard" difficulty level is selected, then the candidate hypothesis is taken from the same disease category. Otherwise, the candidate hypothesis is taken from a disease class at a higher semantic distance. In both cases, the Likert scale -2 gives the full score value, while the other answers are multiplied by 0.75, 0.50, and 0.25, accordingly. However, for the Likert scale answer +2, the obtained score is 0, as it is considered an entirely wrong answer in that case. The full score value is determined with respect to the number of diseases that belong to the same class. With the higher number of diseases belonging to the selected class, the question is considered more difficult. The

impact of this number is corrected by a factor of 0.5, to avoid zero scores in the case of 1 class. The previous score calculation criteria are implemented by the following equation:

$$FullScore := 1 - 0.5 * 1/NumberOfDiseasesFromSameClass(HypothesysDisease) \quad (1)$$

Notably, *FullScore* increases as the number of classes increases while decreasing if the number of classes from the same disease category decreases. In the case of "easy" questions, the full score is further corrected by a multiplicative factor of 0.5. The pseudo-code of the Likert-scale-score assignment algorithm for the indirect strategy is given in Listing 4.

Listing 4. Pseudo-code of Likert-sale-score assignment algorithm for indirect strategy.

```
Input: health_records_db, disease_hypothesis, difficulty level
Output: Likert scale scores
Steps:

1.  If(difficulty level is hard)
2.      new_hypothesis: = SelectRelatedDiseaseFromSameClass(disease_hypothesis);
3.      full_score: = (1 − 1/NumberOfDiseasesFromSameClass(new_hypothesis));
4.  Else
5.      new_hypothesis: = SelectedDiseaseFromAnotherClass(disease_hypothesis);
6.      full_score: = 0.5 * (1 − 1/NumberOfDiseasesFromSameClass(new_hypothesis));
7.  Endif;
8.  LikertPlus2: = 0;
9.  LikertPlus1: = full_score * 0.25;
10. Likert0: = full_score * 0.5;
11. LikertMinus1: = full_score * 0.75;
12. LikertMinus2: = full_score;
13. End.
```

The evaluator sets up the desired number of questions and strategy (difficulty level) using AppSheet [47]-based mobile application (config step in Figure 3). After that, the question generation leveraging ontology-driven code algorithm is executed. During the code generation process, the algorithm executes SPARQL queries against the semantic knowledge base containing the knowledge stored with respect to the form of the desired question type ontology (SCTonto in this case). Optionally, additional domain-specific ontologies might be included (such as Disease Ontology) to support the code generation process. The domain ontologies provide necessary knowledge used for automated question generation mechanisms related to the targeted difficulty level of the question. Results of the SPARQL queries are used to fill in the parameters relevant to the desired type of questions. Moreover, results of the code generation are inserted into the Google Sheets document using the Google Sheets API client in Java [48]. The Google Sheets document is used by AppSheet to generate a mobile application. AppSheet is quite effective when it comes to the rapid creation and distribution of multiplatform web-based mobile applications. Finally, the code generation results are visualized back to the evaluator, so they can be further distributed to the target audience (such as students) via an AppSheet-based mobile app. Apart from AppSheet, the mobile app relies on Google Apps Script triggers for backend capabilities related to testing evaluation and score calculation.

In Listing 5, the pseudo-code summarizing the overall question generation procedure for a single question of SCT-based assessment is given. During this procedure, the values of question properties defined by SCT question ontology are populated. The corresponding Likert-scale-score calculation procedure is executed, depending on the selected strategy.

In Listing 6, an example of a SPARQL query is used to retrieve all possible new information (lab results) for a given hypothesis (disease is given).

The screenshots of the AppSheet-based mobile app for SCT-based assessment from students' perspectives are given in Figure 4. There are three main views. The first one (4a) shows the list of questions that are part of a test. The second (4b) provides the interface to

answer the selected question by setting the value "Y" to one of the values from the Likert scale. Finally, the third screen (4c) shows an overview of scores obtained on previous tests by questions.

Listing 5. Summarization of SCT question generation procedure.

Input: disease name, selection strategy, difficulty level
Output: SCT question
Steps:

```
1.  Config(disease name, selection strategy, difficulty level);
2.  QuestionGeneration(disease name, selection strategy) {
3.     CaseDescription: = GetSoapNoteForDisease(disease name);
4.     NewInformation: = SelectLabResults(disease name, selection strategy);
5.  }
6.  LikertScaleScoreCalculation(disease name, new information, selection strategy, difficulty
    level);
7.  For each property in SCT ontology
8.        InsertTriplet(property, value);
9.  Endfor;
10. End.
```

Listing 6. SPARQL query for new information retrieval based on disease hypothesis.

```
PREFIX sct:  <http://www.example.com/sct/>
         SELECT DISTINCT ?ni
            WHERE {
                GRAPH <http://www.example.com/sct1> {
              ?h sct:hasValue ?disease.
              ?h sct:isPossibleEffectedBy ?ni.
            FILTER(regex(STR(?disease),
                "Chronic Obstructive Pulmonary Disease"))
         }
}
```

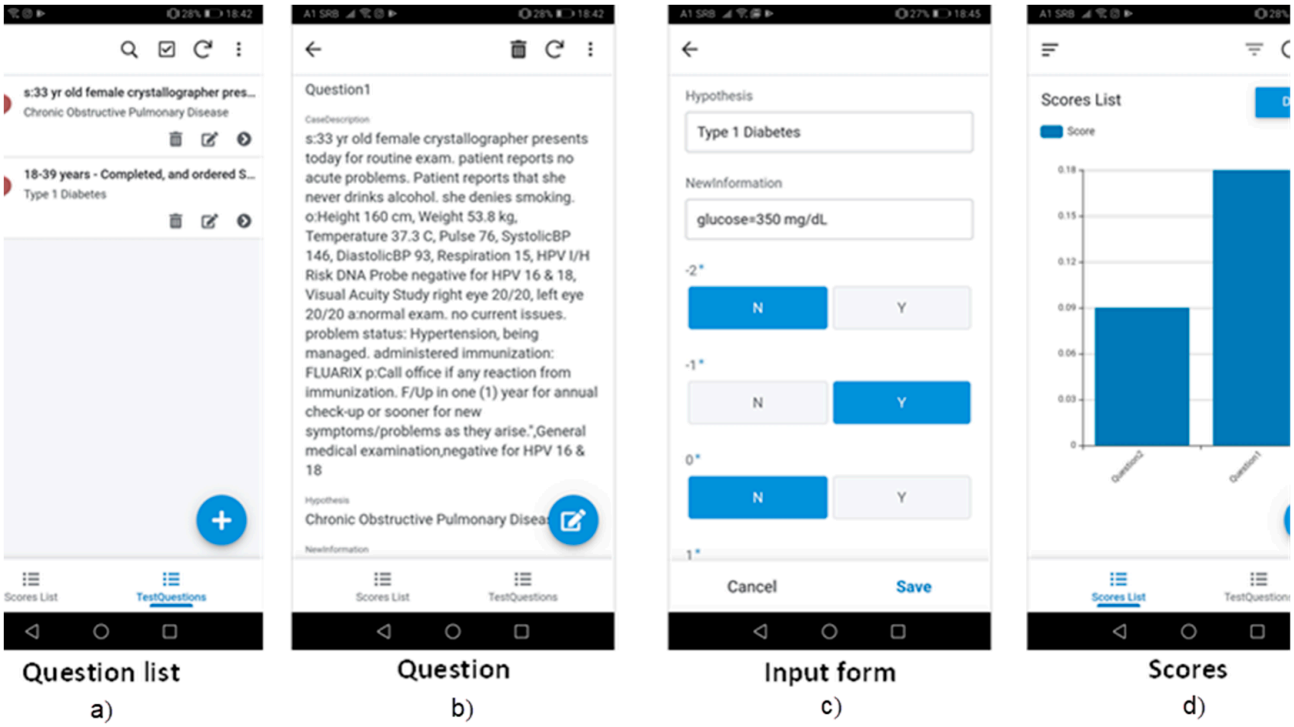

Figure 4. AppSheet-based SCT mobile app for students.

3. Results

The time-based performance evaluation of the proposed framework for the automated, ontology-based generation of questions for SCT-based assessment in medical education is given and compared with the manual creation of the same questions. The evaluation was performed on a laptop equipped with Intel i7 7700HQ CPU, 16GB DDR4 RAM, and 1TB HDD, running on Windows 10. The platform for automated question generation was entirely written in Java, relying on the ontology management, triplet insertion, and querying capabilities of the online TasorONE service and its Java client. Moreover, the backend of the mobile app was written in Apps Script running on Google's cloud infrastructure.

In Table 3, the achieved results for code generation of a single question are given for different disease hypotheses. The first column denotes the disease case selected by an evaluator. The second column denotes the strategy used for the selection of NewInformation corresponding to the targeted difficulty level. The third column shows the number of possible lab results involved (corresponds to NewInformation) for the selected strategy. The fourth column presents the time needed for parsing and triplet insertion with respect to SCT question ontology. The fifth column is the time needed to calculate Likert scale scores for possible answers. The sixth column gives the time needed for semantic query execution and retrieval of parameters, while the seventh column gives the time needed for insertion of the retrieved results into Google Sheets document necessary for mobile application aiming students. Finally, the last column is the total time needed for single question generation, which is a sum of previously mentioned time parameters. All of the time values are given in seconds.

Table 3. Evaluation results for automated generation of questions relying on SCT ontology.

Disease Case (Hypothesis)	Strategy	No. of Candidates	Parsing and Triplet Insertion [s]	Score Calculation [s]	Query Execution [s]	Google Sheets Insertion [s]	Total Time [s]
Chronic Obstructive Pulmonary Disease	Direct	9	1.673	0.004	1.014	0.913	3.603
Diabetes Type 1	Indirect—Hard—Diabetes Type 2	2	1.132	0.002	1.126	1.017	3.277
Diabetes Type 2	Indirect—Easy—Stroke	2	1.432	0.002	1.32	0.931	3.685

According to the obtained results, it can be inferred that question generation time slightly varies but does not exceed the order of magnitude of a second. However, as it can be seen, the indirect strategy has a longer query execution time. This is due to fact that it relies on the retrieval of information about related diseases from the semantic knowledge base, which is not needed in the direct approach. On the other hand, the duration of the Likert scale calculation is longer in the case of the direct approach. Compared with human-based manual construction of such exercises, a panel of experts is needed, while the estimated time for test construction is around 1.5 h. Therefore, the proposed approach for automated question generation significantly reduces the time needed for the construction of SCT-based assessment.

4. Discussion

Clinical reasoning can be defined as the mental process that occurs when a physician meets a patient and has to make a decision on gathered diagnostic information and recommends or initiates treatment. Since clinical reasoning plays a major part in every physician's education, teachers in medical schools need to assess whether students satisfactorily meet this objective. It is mostly accomplished through oral bedside examination and written progress tests [49]. However, when it comes to courses with a large number of students in preclinical practice, the use of web-based tests seems to be a better option. Although arguments are made that a combination of real patients' data in real clinical

settings and computer-based case scenarios would provide a more valid and reliable way of assessing clinical reasoning and clinical competence [2,16], the COVID-19 pandemic forced medical schools to quickly find solutions in offering best digital teaching and assessing for medical students with various online possibilities [50,51]. This issue brought new challenges for research society in terms of how to further improve web-based assessment for clinical easoning.

Following the recent involvement of intelligent systems in all spheres of life, researchers are increasingly focusing on the inclusion of ontologies in the learning assessment area, particularly in terms of automatically generating various types of questions. A literature review revealed that MCQ is the dominant type of question for which different e-assessment platforms are made, while other types of questions such as the script concordance test are less represented, if at all.

On the other hand, the script concordance test emerges as one of the most promising and widely used assessment methods in medical education. It uses a short patient vignette (that is weakly defined) with a diagnostic hypothesis. After being presented a new piece of information relevant to each given hypothesis, students should assign a relevance score to the hypothesis—new information pairs on a scale from −2 to +2, with a score of 0 considered as "no change". Students' judgment is then compared with those of experts (who also score the test). The SCT has been validated in several medical disciplines with satisfactory results [10–15]. In this way, SCT seeks to provide a practical, objective method for evaluating clinical reasoning that is currently assessed subjectively and rather informally in most training programs [6].

The main question addressed in this paper is whether the usability of SCT could be raised to a level high enough to match the current education requirements by exploiting opportunities that new technologies provide, particularly semantic knowledge graphs (SCGs) and ontologies. In other words, could SCT overcome the main drawbacks of traditional standardized tools, such as resource intensiveness, time consumption, and cumbersome administration and scoring?

In order to answer this question, we developed an ontology-driven automated script-concordance-test generation platform. Resource intensiveness was resolved through ontology mapping from medical records stored in a database to previously created SCTonto ontology. Since patient data are populated in EHR's on daily basis, our platform could constantly generate new questions.

In order to obtain Likert scale scores, direct and indirect strategies were proposed and explained in detail. Question generation algorithms ran SPARQL queries against the SCTonto ontology, and the results were used to generate questions presented to users by means of a mobile application created using AppSheet. A performance evaluation for both strategies was conducted, and the results confirm that the proposed approach for automated question generation significantly reduces the time needed for the construction of SCT-based assessment. Thus, the time consumption was resolved as well. Based on the aforementioned contributions, we could state that SCG and ontologies can raise the usability of SCT in order to match the current educational requirements.

5. Conclusions

The following aspects are the main contributions of this paper:

- A methodology for an ontology-driven learning assessment that was proved in the case of SCT;
- A proposal of an ontology-driven automated script concordance test generation platform;
- Direct and indirect strategies for Likert-type scale scoring and the detailed explanation of both approaches;
- Proved usability of SCTonto ontology in the contest of the presented methodology;
- Evaluation of proposed platform against traditional manually created SCT;
- Presentation of experimental results that indicate the significant reduction in the test creation time.

The benefits of the proposed e-assessment platform are manifold. In addition to the obvious benefit for educators whose time for question construction would be greatly reduced, using this platform would facilitate the assessment of a large number of students. Since patient data are populated in EHR's on daily basis, our platform could constantly generate new questions, and the physicians who want to improve their knowledge can use this platform for self-assessment.

In our future research, we plan to conduct an evaluation of generated SCT questions from the viewpoints of both experts and students. Analysis of the evaluation results will help us in further improvement of the proposed platform.

Author Contributions: Conceptualization, M.R. and M.T.; data curation, N.P.; investigation, M.R.; methodology, M.R.; resources, N.P. and M.T.; software, N.P.; supervision, M.T.; validation, M.R., N.P. and M.T.; visualization, M.R. and N.P.; writing—original draft preparation, M.R.; writing—review and editing, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The RDF implementation file of SCTonto is publicly available at:http://www.tasorone.com/tsc/resources/ScriptConcordanceTestAssessmentMethod, accessed on 29 December 2021.

Acknowledgments: The research leading to these results was supported by the Serbian Ministry of Education, Science and Technological Development projects no. 451-03-9/2021-14/200132, 32051 and 47003.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Goos, M.; Schubach, F.; Seifert, G.; Boeker, M. Validation of undergraduate medical student script concordance test (SCT) scores on the clinical assessment of the acute abdomen. *BMC Surg.* **2016**, *16*, 1–9. [CrossRef] [PubMed]
2. Esteves, J.E.; Bennison, M.; Thomson, O.P. Script concordance test: Insights from the literature and early stages of its implementation in osteopathy. *Int. J. Osteopath. Med.* **2013**, *16*, 231–239. [CrossRef]
3. Moghadami, M.; Amini, M.; Moghadami, M.; Dalal, B.; Charlin, B. Teaching clinical reasoning to undergraduate medical students by illness script method: A randomized controlled trial. *BMC Med. Educ.* **2021**, *21*, 1–7. [CrossRef] [PubMed]
4. Lubarsky, S.; Dory, V.; Meterissian, S.; Lambert, C.; Gagnon, R. Examining the effects of gaming and guessing on script concordance test scores. *Perspect. Med. Educ.* **2018**, *7*, 174–181. [CrossRef]
5. Peyrony, O.; Hutin, A.; Truchot, J.; Borie, R.; Calvet, D.; Albaladejo, A.; Baadj, Y.; Cailleaux, P.E.; Flamant, M.; Martin, C.; et al. Impact of panelists' experience on script concordance test scores of medical students. *BMC Med. Educ.* **2020**, *20*, 313. [CrossRef] [PubMed]
6. Lubarsky, S.; Dory, V.; Duggan, P.; Gagnon, R.; Charlin, B. Script concordance testing: From theory to practice: *AMEE Guide No. 75. Med. Teach.* **2013**, *35*, 184–193. [CrossRef] [PubMed]
7. Sibert, L.; Charlin, B.; Corcos, J.; Gagnon, R.; Grise, P.; Vleuten, C.V.D. Stability of clinical reasoning assessment results with the script concordance test across two different linguistic, cultural and learning environments. *Med. Teach.* **2002**, *24*, 522–527. [CrossRef]
8. Sibert, L.; Darmoni, S.J.; Dahamna, B.; Hellot, M.F.; Weber, J.; Charlin, B. On line clinical reasoning assessment with Script Concordance test in urology: Results of a French pilot study. *BMC Med. Educ.* **2006**, *6*, 1–9. [CrossRef]
9. Charlin, B.; Roy, L.; Brailovsky, C.; Goulet, F.; van der Vleuten, C. The Script Concordance test: A tool to assess the reflective clinician. *Teach. Learn. Med.* **2002**, *12*, 189–195. [CrossRef]
10. Subra, J.; Chicoulaa, B.; Stillmunkès, A.; Mesthé, P.; Oustric, S.; Rougé Bugat, M.E. Reliability and validity of the script concordance test for postgraduate students of general practice. *Eur. J. Gen. Pract.* **2017**, *23*, 209–214. [CrossRef]
11. Aldekhayel, S.A.; Aselaim, N.A.; Magzoub, M.E.; AL-Qattan, M.M.; Al-Namlah, A.M.; Tamim, H.; Al-Khayal, A.; Al-Habdan, S.I.; Zamakhshary, M.F. Constructing a question bank based on script concordance approach as a novel assessment methodology in surgical education. *BMC Med. Educ.* **2012**, *12*, 100. [CrossRef] [PubMed]
12. Lubarsky, S.; Chalk, C.; Kazitani, D.; Gagnon, R.; Charlin, B. The script concordance test: A new tool assessing clinical judgement in neurology. *Can. J. Neurol. Sci.* **2009**, *36*, 326–331. [CrossRef] [PubMed]

13. Chime, N.O.; Pusic, M.V.; Auerbach, M.; Mehta, R.; Scherzer, D.J.; Van Ittersum, W.; McAninch, B.; Fein, D.M.; Seelbach, E.; Zaveri, P.; et al. INSPIRE Network. Script Concordance Testing to Determine Infant Lumbar Puncture Practice Variation. *Pediatr. Emerg. Care* **2018**, *34*, 84–92. [CrossRef] [PubMed]
14. Atayee, R.S.; Lockman, K.; Brock, C.; Abazia, D.T.; Brooks, T.L.; Pawasauskas, J.; Edmonds, K.P.; Herndon, C.M. Multicentered Study Evaluating Pharmacy Students' Perception of Palliative Care and Clinical Reasoning Using Script Concordance Testing. *Am. J. Hosp. Palliat. Care* **2018**, *35*, 1394–1401. [CrossRef]
15. Kania, R.E.; Verillaud, B.; Tran, H.; Gagnon, R.; Kazitani, D.; Huy, P.T.; Herman, P.; Charlin, B. Online script concordance test for clinical reasoning assessment in otorhinolaryngology: The association between performance and clinical experience. *Arch. Otolaryngol. Head Neck Surg.* **2011**, *137*, 751–755. [CrossRef]
16. Fournier, J.P.; Demeester, A.; Charlin, B. Script concordance tests: Guidelines for construction. *BMC Med. Inform. Decis. Mak.* **2008**, *8*, 1–7. [CrossRef]
17. Rose, S. Medical Student Education in the Time of COVID-19. *JAMA* **2020**, *323*, 2131–2132. [CrossRef]
18. Ribeiro, J.C.; Villanueva, T.; Gi, A.; Escada, P. Constraints Lead to Opportunities for Medical Education in Times of COVID-19 Pandemic. *Acta Médica Port.* **2020**, *33*, 638–639. [CrossRef]
19. Chen, D.; Zhao, H.; Zhang, X. Research on the Aided Diagnosis Method of Diseases Based on Domain Semantic Knowledge Bases. *IEEE Access* **2019**, 1–11. [CrossRef]
20. Shen, B.; Guo, J.; Yang, Y. MedChain: Efficient Healthcare Data Sharing via Blockchain. *Appl. Sci.* **2019**, *9*, 1207. [CrossRef]
21. Seifan, A.; Mandigo, M.; Price, R.; Galetta, S.; Jozefowicz, R.; Jaffer, A.; Symes, S.; Safdieh, J.; Isaacson, R.S. Education Research: Can my electronic health record teach me something? A multi-institutional pilot study. *Neurology* **2013**, *80*, e98–e103. [CrossRef] [PubMed]
22. Gladun, A.; Rogushina, J.; Garcı, F.; Martínez-Béjar, R.; Fernández-Breis, J.T. An application of intelligent techniques and semantic web technologies in e-learning environments. *Expert Syst. Appl.* **2009**, *36*, 1922–1931. [CrossRef]
23. Cubric, M.; Tosic, M. Towards automatic generation of e-assessment using semantic web technologies. *Int. J. E-Assess.* **2011**, *1*, 1–9.
24. Ev, V.; Alsubait, T.; Kumar, P.S. Modeling of Item-Difficulty for Ontology-based MCQs. *arXiv* **2016**, arXiv:1607.00869.
25. Demaidi, M.N.; Gaber, M.M.; Filer, N. Evaluating the quality of the ontology-based auto-generated questions. *Smart Learn Environ.* **2017**, *4*, 7. [CrossRef]
26. Radović, M.; Petrovic, N.; Tosic, M. Ontology-based generation of multilingual questions for assessment in medical education. *J. Teach. Engl. Specif. Acad. Purp.* **2020**, *8*, 1–15. [CrossRef]
27. Kurdi, G.; Leo, J.; Parsia, B.; Sattler, U.; Al-Emari, S. A systematic review of automatic question generation for educational purposes. *Int. J. Artif. Intell. Educ.* **2020**, *30*, 121–204. [CrossRef]
28. Divate, M.; Salgaonkar, A. Automatic question generation approaches and evaluation techniques. *Curr. Sci.* **2017**, *113*, 1683–1691. [CrossRef]
29. Litherland, K.; Carmichael, P.; Martínez-García, A. Ontology-based e-assessment for accounting: Outcomes of a pilot study and future prospects. *J. Account. Educ.* **2013**, *31*, 162–176. [CrossRef]
30. Al-Yahya, M. Ontology-based multiple choice question generation. *Sci. World J.* **2014**, *2014*, 274949. [CrossRef]
31. Fattoh, I.E. Automatic multiple choice question generation system for semantic attributes using string similarity measures. *Comput. Eng. Intell. Syst.* **2014**, *5*, 66–73.
32. EV, V.; Kumar, P.S. Automated generation of assessment tests from domain ontologies. *Semant. Web.* **2017**, *8*, 1023–1047.
33. Deepak, G.; Kumar, N.; Bharadwaj, G.V.S.Y.; Santhanavijayan, A. OntoQuest: An ontological strategy for automatic question generation for e-assessment using static and dynamic knowledge. In Proceedings of the 2019 Fifteenth International Conference on Information Processing (ICINPRO), Bengaluru, India, 20–22 December 2019; IEEE eXpress Conference Publishing: New York NY, USA, 2019; pp. 1–6.
34. Santhanavijayan, A.; Balasundaram, S.R. Fuzzy-MCS algorithm-based ontology generation for e-assessment. *Int. J. Bus. Intell. Data Min.* **2019**, *14*, 458–472. [CrossRef]
35. Diatta, B.; Basse, A.; Ouya, S. Bilingual ontology-based automatic question generation. In Proceedings of the 2019 IEEE Global Engineering Education Conference (EDUCON), Dubai, United Arab Emirates, 8–11 April 2019; pp. 679–684.
36. Leo, J.; Kurdi, G.; Matentzoglu, N.; Parsia, B.; Sattler, U.; Forge, S.; Donato, G.; Dowling, W. Ontology-Based Generation of Medical, Multi-term MCQs. *Int. J. Artif. Intell. Educ.* **2019**, *29*, 145–188. [CrossRef]
37. Radovic, M.; Tosic, M.; Milosevic, D.; Milosevic, M.; Milosevic, M. Semantic Approach to Script Concordance Test. In Proceedings of the International Scientific Conference—UNITECH, Gabrovo, Bulgaria, 16–17 November 2018; University Publishing House "V. Aprilov": Gabrovo, Bulgaria, 2018; pp. 137–141.
38. Falbo, R.A. SABiO: Systematic approach for building ontologies. In Proceedings of the 1st Joint Workshop Onto.Com/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering, Rio de Janeiro, Brazil, 21 September 2014.
39. Fernandes, P.C.B.; Guizzardi, R.S.; Guizzardi, G. Using goal modeling to capture competency questions in ontology-based systems. *J. Inf. Data Manag.* **2011**, *2*, 527.
40. Medical Records 10 Years. Available online: https://data.world/arvin6/medical-records-10-yrs (accessed on 7 February 2021).
41. Grafoo Specification. Available online: https://essepuntato.it/graffoo/specification/ (accessed on 29 December 2021).
42. Uschold, M.; Gruninger, M. Ontologies: Principles, methods and applications. *Knowl. Eng. Rev.* **1996**, *11*, 93–136. [CrossRef]
43. TasorOne. Available online: http://www.tasorone.com/tasorone/index.html (accessed on 29 December 2021).

44. SCTOnto Ontology. Available online: http://www.tasorone.com/tsc/resources/ScriptConcordanceTestAssessmentMethod (accessed on 29 December 2021).
45. Data World. Available online: https://data.world/ (accessed on 29 December 2021).
46. Disease Ontology. Available online: https://disease-ontology.org/ (accessed on 29 December 2021).
47. The Intelligent No-Code Platform. Available online: https://www.appsheet.com/ (accessed on 29 December 2021).
48. Java Quickstart. Available online: https://developers.google.com/sheets/api/quickstart/java (accessed on 29 December 2021).
49. ten Cate, O.; Durning, S.J. Approaches to assessing the clinical reasoning of preclinical students. *Princ. Pract. Case-Based Clin. Reason. Educ.* **2018**, *5*, 65–72.
50. Egarter, S.; Mutschler, A.; Brass, K. Impact of COVID-19 on digital medical education: Compatibility of digital teaching and examinations with integrity and ethical principles. *Int. J. Educ. Integr.* **2021**, *17*, 18. [CrossRef]
51. Naylor, K.; Torres, K. Approaches to stimulate clinical reasoning in continuing medical education during the coronavirus disease 2019 pandemic. *Kardiol. Pol.* **2020**, *78*, 770–772. [CrossRef]