UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

# CS/IT Honours Project Final Paper 2023

Title: Generated Adaptive Learning Materials

Author: Nervesh Naidoo

Project Abbreviation: GALMAT

Supervisor(s): Toky Raboanary and Professor Maria Keet

| **Category** | *Min* | *Max* | Chosen |
|---|---|---|---|
| Requirement Analysis and Design | *0* | *20* | 10 |
| Theoretical Analysis | *0* | *25* | 0 |
| Experiment Design and Execution | *0* | *20* | 15 |
| System Development and Implementation | *0* | *20* | 10 |
| Results, Findings and Conclusions | *10* | *20* | 15 |
| Aim Formulation and Background Work | *10* | *15* | 10 |
| Quality of Paper Writing and Presentation | *10* | | 10 |
| Quality of Deliverables | *10* | | 10 |
| Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*) | *0* | *10* | 0 |
| **Total marks** | | **80** | **80** |

# Generating Adaptive Questions to Assess a Learner Using an Ontology

Nervesh Naidoo
University of Cape Town
Cape Town, South Africa
Nervesh,NDXNER029@myuct.ac.za

## ABSTRACT

Automatic Question Generation (AQG) algorithms generate a set of questions for a learner from a domain of information. AQG algorithms allow adaptive learning systems to periodically assess a learner on a knowledge domain. Existing AQG algorithms require a large amount of data to generate tailored questions, or they generate questions that are of a poor quality that do not correctly adapt to a learner. The conducted research aims to devise an improved template-based AQG algorithm that uses an ontology as a knowledge source. The newly created AQG algorithm should generate questions that are grammatically correct, understandable, adaptable and suitable for a learner. The AQG algorithm has been quantitatively evaluated, through systematic and user evaluations, to determine whether its inclusion in an adaptive learning system will be of benefit. The results of the evaluation showcased that the generated questions were grammatically correct, understandable and adaptive. However, the generated questions were generally found to be too easy for participants. This shows that there is potential for future improvement within this research to generate questions of a more difficult standard.

## CCS CONCEPTS

• **Information Retrieval** → **Ontologies**; • **Theory of Computation** → *Automatic Question Generation*; • **Information Storage** → Learner Knowledge Model.

## KEYWORDS

Automatic Question Generation, Ontologies, Learner Knowledge Model, Computerized Adaptive Testing, Adaptive Learning

## 1 INTRODUCTION

Education is a fundamental part of society that teaches many individuals, young or old, important concepts or information that will be of benefit to them. However, traditional teaching methods are static in the sense that it follows a one size fits all approach, whereby each student receives the exact same content as everyone else. This is where the concept of Adaptive Learning is introduced.

Adaptive Learning systems are modern educational systems that leverage some form of Machine Learning (ML) to provide each learner with a set of unique learning materials [16]. Therefore, adaptive learning systems aim to solve the static problem found in traditional teaching by ensuring that each learner can receive learning materials that are tailored to their individual abilities and capabilities [13]. There are three foundational components of an adaptive learning system from our viewpoint, which are: an Automatic Question Generation (AQG) algorithm, an algorithm that is able to identify knowledge gaps and create a learner knowledge model, and finally a Natural Language Generation (NLG) algorithm that is able to leverage Artificial Intelligence (AI) to generate a unique set of learning materials for a learner.

The AQG algorithm is the first component in the adaptive learning system, as it allows the system to identify a learner's knowledge gaps [10]. Once questions have been generated for a learner and the learner answers the questions, the AQG algorithm outputs the learner's answers, in a predefined format, to the next component. This component uses the output produced by the AQG algorithm to identify knowledge gaps within a learner and create or update a learner knowledge model, which is a weighted knowledge graph. The learner knowledge model is the output of this component, and it is sent to both the AQG and NLG components. The NLG component uses the learner knowledge model to generate a tailored set of learning materials for the learner by leveraging AI. Furthermore, the AQG component consults the learner knowledge model on the next iteration to identify a learner's weak and strong points, and to thereafter generate tailored questions for a learner. Thus, the AQG component ensures that the system is able to constantly adapt and generate learning materials that will bridge the identified knowledge gaps found within each learner. AQG algorithms automatically generate questions for a learner using the information found in a given domain.

There are many existing AQG algorithms, however, they are not without their flaws. Computerized Adaptive Testing (CAT) is an example of one such algorithm that is able to accurately and uniquely test a specific learner by using their answers to previous questions as input [15]. Although the results of CAT showcase that it produces successful results, CAT requires a large amount of pretesting or calibration to determine initial difficulty levels of potential questions [13]. Thus, the feasibility of this approach is questionable as this pretesting phase would run the risk of displaying the questions to a learner before they are going to actually be tested [15].

The aim of this research is to analyze existing AQG algorithms, and to then devise a new AQG algorithm that seeks to improve the grammaticality, understandability, adaptability and suitability of the generated questions. The new AQG algorithm smoothly integrates into an adaptive learning system, allowing the system to constantly evaluate and receive an updated version of a learner's knowledge model. A learner knowledge model stores a learner's ability for each concept that they have received a question about. The adaptive learning system uses the updated version of a learner's knowledge model to adjust the learning materials to address the identified knowledge gaps.

The newly created AQG algorithm has been evaluated systematically to ensure that the code was adapting and asking the learner

questions that were relevant to them. Furthermore, a user evaluation has been conducted to determine whether users felt that the questions were grammatically correct, understandable, adaptable to their knowledge, and suitable to their difficulty tolerance.

The remainder of this paper is structured as follows. The next section provides a background of the use of AQG algorithms within adaptive learning systems, as well as other related work. Thereafter, the exact methodology that was followed to devise an improved AQG algorithm to successfully integrate within an adaptive learning system will be detailed. The results of the evaluation of the improved AQG will follow where quantitative analysis will determine whether the newly proposed AQG is successful in achieving its aims. Then, the discussion of the results will be presented, along with the limitations of the research conducted. Finally, we conclude the research paper and will present opportunities for future work.

## 2 BACKGROUND

In preparation for creating an adaptive learning system, a knowledge source needs to be decided upon. This knowledge source can either consist of structured or unstructured data [4]. Structured data is data that follows some form of order which makes it much easier to search for specific items as the flow can be predicted. On the flip side, unstructured data follows no formal order which results in searching for specific items to be a difficult operation [4]. The characteristics of structured data align with the purpose of an adaptive learning system; thus it is the selected format of data for the adaptive learning system. The specific type of structured data that will be the knowledge source of the adaptive learning system is a course ontology.

A course ontology is an ontology whose domain is limited to a specific course [6]. The course ontology was selected as the knowledge source for the adaptive learning system due to its native characteristics that make it ideal for easily identifying relationships between different concepts [11]. A course ontology is structured as a representation of concepts and their relationships and is often organized in a hierarchical manner [2]. These relationships between concepts allow for questions to be automatically and adaptively generated.

## 3 RELATED WORK

### 3.1 USING STRUCTURED AND UNSTRUCTURED KNOWLEDGE AS A SOURCE FOR QUESTION GENERATION

Kumar et al. [9] proposed an AQG algorithm that utilized both structured and unstructured knowledge. A PDF file, containing learning materials for a course, is fed into the algorithm, where it is preprocessed and converted into a DOCX file, where it is thereafter converted into a text file. There are two independent processes that occur within this algorithm which are the generation of questions using an ontology, and the generation of questions using ML [9]. The preprocessed DOCX file is the input to both processes. These two processes generate questions of a different type. The ontology-based approach generates Wh-questions (Who, why, what questions) whereas the ML-based approach generates fill-in-the-blank questions. The ontology-based approach models an ontology using the Protégé tool, using the DOCX file to identify various classes and relationships. The ML-based approach begins by sentence selection, where sentences are selected from the DOCX file if the topic of the sentence appears in a manually created “Topic_list”, which stores the topics that a domain expert creates. Thereafter the key sentences are selected that meet further specified criteria. Finally, the keywords are selected from these key sentences. A keyword is a word that would be replaced by a blank when presented to the learner. The algorithm proposed by Kumar et al. [9] produced questions that were generally grammatically correct and understandable. However, a large amount of preprocessing is required to take place before generating questions. Furthermore, the proposed AQG algorithm does not include any adaptive aspects to it. This indicates that the questions are solely generated using the PDF file provided, it does not take into account a learner’s abilities.

### 3.2 AQG INTEGRATED WITH COMPUTERIZED ADAPTIVE TESTING

Susanti et al. [15] proposed that integration of CAT and an AQG algorithm would solve the primary problem found in CAT. The primary problem with CAT is that it requires a large amount of previously administered items known as the item bank [15]. Furthermore, a large amount of preprocessing is required, known as item calibration, which estimates the item parameters. Susanti et al. [15] suggested that an AQG algorithm could successfully be integrated with CAT to reduce or possibly eliminate this preprocessing phase.

The AQG algorithm would allow for the generation of questions, as well as their respective difficulty, and all these questions would be stored in a large question bank. CAT would then select questions from the question bank that match the estimated proficiency of the learner, which is re-estimated after each answer that they provide to a question. This ensures that questions are selected according to a learner’s general proficiency level, meaning that only learners who have a high proficiency level will receive difficult questions [15]. The proposed solution produced promising results through the evaluation. However, the evaluation still required some form of pretesting to predetermine the item’s difficulty. Just not as much as was required in standalone CAT.

### 3.3 NLG BASED AQG

NLG is the process of automatically generating human-readable text that is indistinguishable from text generated by humans [14]. Al-Yahya et al. [17] proposed an approach that uses three different types of generation techniques including class membership, individuals and property to generate multiple choice questions (MCQ) [17]. The class membership technique looks at the classes within the ontology, as well as their instance members, and uses these to formulate questions. The individual’s technique looks at the individual objects within the ontology. The algorithm will then retrieve all of the objects to which the individual is either a subject or predicate of. All of these entities would then be combined to form a question. All the generated questions would then be converted to natural language, to make the question appear more understandable [17]. The generation of distractors within the MCQ consists of randomly generating distractors from the sets of classes within the ontology

that are similar to the key in the question –the key is the correct option in the MCQ. [17].

Ibrahim et al. [6] proposed a template based AQG approach. The research looked at real-world assessment questions with the intention of converting these questions into a general template. Thereafter, both the question word and the keywords would be replaced by placeholders. These placeholders would then eventually be replaced by objects within the ontology that are able to accurately fit within the template. The research then proposed that to provide question variation, multiple templates should be categorized and created [6]. This ensures that multiple question templates can be created that ask different types of questions that use different concepts. This allows for the essential variation that is required in AQG algorithms and ensures that all concepts within the ontology are able to be tested. The evaluation conducted within the research presented in [6] showed that a template-based approach to AQG using ontologies is effective. However, the research did not generate adaptive questions. It is solely an AQG algorithm using only an ontology as input.

Divate et al. [3] researched and evaluated several different AQG systems. Divate et al. [3] states that AQG has four different steps: sentence simplification, answer phrase selection, sentence transformation, and question ranking and evaluation [3]. Sentence simplification consists of categorizing sections of a sentence, such as into independent clauses. Thereafter, the answer phrase selection step looks at the simplified sentence that is categorized and selects potential answers to potential questions. The sentence transformation takes in the simplified sentence and answer phrase selection and generates a set of Wh-questions using either a template-based, syntactic-based, or semantic-based approach. The template-based approach resembles the approach proposed by Ibrahim et al. [6]. However, the syntactic-based approach is concerned with rearranging the syntax of a sentence in order to phrase it as a question. The semantic-based approach extracts semantic associations within the sentence, and then formulates a new sentence using these semantic extractions. The evaluation conducted by Divate et al. [3] shows that the template-based approach was simpler and more effective than the syntactic and semantic-based approaches when a high-quality question template is created.

### 3.4 ONTOLOGY-BASED APPROACH

Alsubait et al. [1] proposed a method that would generate potential MCQ questions using an ontology as a knowledge source. Thereafter, the potential questions would be converted into grammatically correct and understandable questions. Alsubait et al. [1] generated questions using three different approaches. The first approach was using the naïve AQG approach proposed by Žitko et al. [18], where MCQ distractors were randomly generated from the ontology. The second approach was proposed by Papasalouros et al. [12], and the approach consisted of generating distractors for the MCQ that are similar to the key, the key is the correct MCQ answer. An example of such an approach is to use instances of a superclass that is a superclass to the class that contains the key [12]. The final approach is proposed by Alsubait et al. [1] and is known as the similarity-based MCQ generation approach. The similarity-based approach improves on the naïve approach as it can generate higher quality distractors that are able to distract the learner. Furthermore, the similarity-based approach improves on the second approach proposed by Papasalouros et al. [12] as it does not require as large of an ontology to generate high quality distractors.

The similarity-based approach considers generating distractors that are very similar to the key for learners who are more knowledgeable with the learning materials, as this would increase the difficulty of the question as they are not able to arrive at the correct answer through process of elimination [1]. The similarity between two classes is determined by whether they have more in common, or more unique characteristics. The evaluation of the similarity-based approach proposed by Alsubait et al. [1] shows that participants found grammatical errors with the generated questions. However, participants found that the difficulty level of the questions were accurate.

### 3.5 BAYESIAN-BASED APPROACH

Khodeir et al. [7] proposed a method that can generate questions for probabilistic domains that do not provide support for syntactic or linguistic analysis. The Bayesian-based AQG approach consists of six components: student knowledge estimator, evidence selector, question evaluation module, question difficulty level estimator, and the question selector. This approach uses Item Response Theory (IRT) to accurately estimate the difficulty of a question for a learner. The characteristics of a Bayesian network allow for this approach to constantly track and have an updated student knowledge model at all times. Therefore, the generated questions are based on this updated student knowledge model so as to generate the most relevant questions for a learner. The evaluation conducted by Khodeir et al. [7] used a simulation of 50 participants to interact with the adaptive Bayesian-based AQG, and compared the results to the baseline that did not adapt to a learner knowledge model. The evaluation shows that the adaptive Bayesian-based AQG improved on the baseline model by increasing the relevance of generated questions by 40% [7].

## 4 METHODS AND IMPLEMENTATION

### 4.1 REQUIREMENTS GATHERING

The nature of an adaptive learning system is very tightly coupled. The AQG has to closely work with the component which identifies knowledge gaps within a learner, and this component needs to closely work with the NLG component. Thus, it is imperative that the entire system is carefully planned to ensure smooth integration of the separate components. The requirements gathering phase of the research consisted of several meetings with the supervisor and team members. Here, both the functional and non-functional requirements were set out. The functional requirements of the AQG algorithm included ensuring that the algorithm would be able to present each learner with a unique set of questions, dependent on their learning abilities on various concepts. Furthermore, the AQG algorithm should take into account a learner's proficiency with a particular concept. Thus, if a learner is particularly strong at a concept, they should receive more difficult questions on that topic to ensure that they do not find the questions too easy or become tired of the questions. The output of the AQG algorithm had to also

smoothly integrate with the next component, as the output of the AQG is the input of the next component.

The non-functional requirements of the AQG algorithm included ensuring that the algorithm was operating efficiently, both in the time and space aspects. It is an important consideration to ensure that the questions presented to a learner are generated both quickly and efficiently. This ensures that the learner has a positive experience interacting with the AQG, which in turn results in a better adaptive learning experience.

## 4.2 SYSTEM ARCHITECTURE

Figure 1 below shows the system architecture of the AQG component. The AQG will read and locally store all the content in the ontology to use for the question generation process. Furthermore, the AQG takes in a learner knowledge model as an input. The learner knowledge model is a weighted knowledge graph, that contains a specific learner's learning ability for each concept that they have been asked a question about. The format of the learner knowledge model is: {concept, relationship, predicate, $o$, $p$}, where both $o$ and $p$ are numeric values between -4 and 4 which represent the learner's ability of the concept and predicate, respectively.

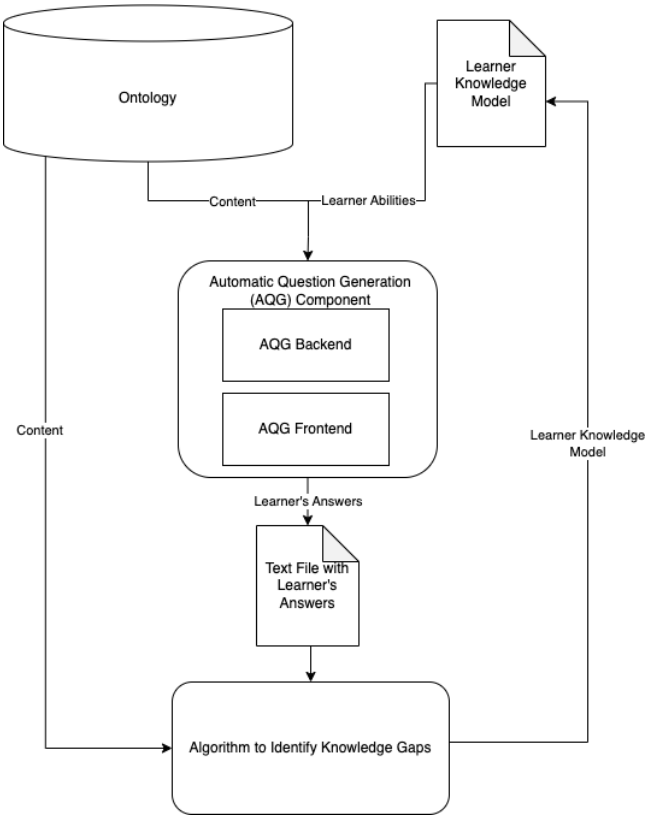


**Figure 1: System architecture of AQG component**

Within the AQG component, the backend automatically and adaptively generates questions for a learner using the ontology and learner knowledge model. The backend then sends the final questions to the frontend, where the questions are displayed for the learner in a simple graphical user-interface (GUI). The backend retrieves all of the answers, and writes the answers, which are stored in a list of *learnerAnswer* objects, to a text file. This text file is one of the inputs of the next component. Each *learnerAnswer* object contains the concept of the question, the learner's answer to the question, the correct answer to the question, as well as the difficulty level associated with the question. The next component then uses these values to update the learner knowledge model, which is sent once again to the AQG component to be used for generating the next set of questions.

## 4.3 APPROACH

The domain selected for the ontology consisted of information related to food. Therefore, the ontology stored information about various foods and their main ingredients, their general ingredients, what country or region they are consumed in, alternative names for the food, as well as other relationships. Furthermore, the ontology also stored information about the various countries in which the foods are eaten in, such as what language they speak and who the past and present presidents of the country are. The choice of a food-based ontology was chosen so that the AQG component is smoothly integrated with the Natural Language Generation (NLG) component in the adaptive learning system. The NLG component was limited to one of a few available datasets, and thus a food based WebNLG dataset was chosen. The structure of the ontology which consists of many relationships was exploited to automatically generate true or false questions.

The approach that was undertaken to create the AQG algorithm produced in this research was inspired by several existing AQG algorithms, such as those presented by Susanti et al. [15] and Ibrahim et al. [6]. The pseudocode of the algorithm is presented in Algorithm 1 below. A high-level overview of the AQG algorithm was that it should predefine several different question templates which are classified as either normal or difficult. Normal question templates are templates that only include a single subject and a single predicate. Whereas difficult question templates are those that include a single subject, but two or more predicates. This is also known as multi-term question templates. We consider this to be a more difficult question template as both predicates would need to be true for the answer to the question to be true. Therefore, the difficult question template requires a learner to pay closer attention and think harder to answer these questions correctly. An example of a normal question template would be "$y$ has $k$ as an ingredient" and an example of a difficult question template would be "$y$ is from $v$ and has $k$ as an ingredient". In the previous examples, $y$ denotes a particular food, $k$ denotes a particular ingredient and $v$ denotes a country. Furthermore, the questions are phrased as statements as the learner is to select whether the statement is true or false.

Thereafter, these question templates should be filled using various classes within the ontology as input, and this can be seen in lines 12 to 21 in the pseudocode. This is where the native characteristics of an ontology are useful in question generation, as you are able to easily retrieve predicates of a class within the ontology, and form questions with them using the predefined templates. All of the generated questions will be stored in two large question

banks, one question bank for storing questions of a normal difficulty, and the other for storing questions of a difficult difficulty. Once these large question banks have been created and initialized, the AQG algorithm should then adaptively select questions to ask the learner, dependent on their knowledge model. The learner knowledge model would store their learning ability for each particular concept that they have been tested on. A large consideration of the newly created AQG algorithm was that it should be light weight by not requiring a large amount of data. The intention was to build an AQG algorithm that did not require pretesting of questions to determine their difficulty or did not require a significant amount of information relating to the learner to determine their abilities and capabilities. The newly created AQG algorithm should simply start of equally for every learner and assume no knowledge of them. Only thereafter will the algorithm adapt and adjust the questions based on the learner's answers to previous questions.

Python lists were the chosen data structures to store all of the generated questions and local learner knowledge model. The AQG algorithm does not contain large nested loops that would result in an exponential time complexity, thus the simplicity and light-weight nature of lists makes it an ideal candidate for storing the questions. Lists in python are simple to append or remove items from. Furthermore, they also allow for simple sorting of the contents within the list, which has been leveraged with the AQG algorithm to identify the three worst concepts within the knowledge model.

The adaptive aspect of the AQG algorithm, which can be seen from lines 22 to 33 of the pseudocode, is concerned with selecting specific questions from the question banks that meet a specific criterion. There are two possibilities, either the learner knowledge model has values within it, or it does not. If the learner knowledge model is empty, it indicates that this is the first iteration of the assessment for the learner, meaning that they have not yet interacted with the assessment. In the case that the knowledge model is empty, then a total of $x$ questions are randomly selected from the item bank for the learner. The questions will be selected from the item bank only containing the questions of a normal difficulty, as it does not assume that the learner is capable of answering more difficult questions. If the knowledge model is filled with values, then questions will be selected from the normal question bank where the subject of the question is one of the learner's worst three subjects, ranked by learner ability. Furthermore, questions will also be selected from the normal question bank where the subject of the question is related to one of the learner's worst three subjects. We consider two subjects to be related to one another if they are both found in the same question, and both have predicates. If the sum of questions that meet these criteria does not amount to $x$, then the remaining questions are randomly selected from the normal question bank until the total amount of questions amount to $x$.

The difficulty of questions presented to a learner are controlled by the two separate question banks, one for questions of a normal difficulty and one for a more difficult difficulty. There are no criteria that must be met for questions to be selected from the normal question bank, as this is the default question bank from where questions are selected from. For a question to be selected from the difficult question bank, the learner has to have a learning ability above $-1$ for the subject of the question. This ensures that difficult questions are exclusively asked to learners who are capable of answering the question. This prevents the occurrence of difficult questions being asked to learners who are not yet capable of answering them, which would ultimately put them off and negatively affect their confidence. The value of $-1$ was selected as the threshold as it is the value that produced the best results regarding the learner ability aspect, which is calculated in the next component. The value of $-1$ allows for an efficient balance between the number of learners who receive a difficult question, and the number of learners who do not receive a difficult question as they are not yet perceived capable of answering it.

The criteria that are used to select questions from the question banks have been defined. However, there is a component of randomness that is also integrated into the AQG algorithm. This is to prevent the learner from only receiving questions relating to the same few topics. If any of the worst three subjects in the knowledge model have a learning ability over 1, then we assume that it is an opportunity to ask the learner a randomly fetched question from the normal question bank instead. This addition ensures that the learner is always introduced to questions and does not constantly receive the same questions infinitely. Furthermore, it helps with readjusting the learner knowledge model as the goal is to have a constantly flowing and updating knowledge model that is actively reflecting the learner's abilities. Thus, the questions that are presented to the learner are of great importance.

The newly created AQG component was explicitly designed to be a light-weight and modular component that was able to automatically generate tailored questions for a learner using as little information necessary. To achieve this, the algorithm was separated into many methods, including a separate function for each question template. This allowed for templates to be easily removed, modified or added to the AQG algorithm without impacting any of the other methods. This allows for easy extension of new templates, as well as modification of existing templates to incorporate feedback gathered from the evaluation results. A class diagram of the AQG system can be seen in Appendix F.

The AQG algorithm was designed with the goal of ensuring both maintainability and portability. Maintainability of the software was achieved through ensuring that there was an acceptable division of duties amongst the various classes and methods. This ensures that changes in the implementation of a class or method will not affect other sections of the software, as long as that class or method performs the same overall function.

## 4.4 IMPLEMENTATION

The ontology was created in *Java* with the *Protégé* tool. *Protégé* is an open-source tool that is used for defining ontologies. Regarding the portability of the AQG algorithm, the choice of language was *Python*. *Python* was selected for multiple reasons, including it's simplicity and ability to effectively work with ontologies through the *Owlready2 Python* library. *Owlready2* is an extensive *Python* library that is used for ontology-oriented programming and allowed a connection to be made with the ontology. Furthermore, *Python* is a platform-independent language, meaning that it is able to run on any platform. Therefore, the AQG algorithm ensures portability.

The project development methodology that was followed was Agile. This iterative approach allowed us to constantly refine and

**Algorithm 1** High-Level Pseudocode of Approach

```
load ontology
for all classes in ontology do
    categorize class as either food or country
end for
for all food items do
    categorize food item predicates into distinct lists
end for
for all countries do
    categorize country predicates into distinct lists
end for
read and store learner knowledge model locally
repeat
    for all food or country objects do
        fill in question template with food/country object
        if difficultyLevel = 0 then
            store question in normal question bank
        else
            store question in difficult question bank
        end if
    end for
until question templates exhausted
identify three worst concepts in learner knowledge model
finalQuestions ← 0
for all questions in normal question bank do
    if question subject is, or related to, one of the worst three concepts then
        append question to finalQuestion list
    end if
end for
for all questions in difficult question bank do
    if learnerAbility for question subject>1 then
        append question to finalQuestion list
    end if
end for
if finalQuestions>x then
    while finalQuestions>x do
        Remove question
    end while
end if
while finalQuestions<x do
    fetch and append random question from normal question bank
end while
for all finalQuestions do
    present question to learner
    store learners answer in a object and append to list
end for
send list of learner answers to next component
```

seamlessly update the algorithm should new information become available [8]. This made it easier to ensure integration with the other components in the adaptive learning system, as if their structure changed, so would the AQG algorithm.

The project was completed in multiple phases. Creating a link with the ontology and then separating all the classes within the ontology into a multitude of lists was the first step. Thereafter, the question templates were created, and code was created that would automatically populate each template with the relevant information using the lists initialized in the previous steps. Once the question templates were filled and stored in the question banks, the adaptive algorithm would need to be created so as to choose which questions to select for the learner. We defined $x = 15$, where $x$ is the number of questions presented to a learner. The reason as to why we chose $x = 15$, is so that the learner is presented with a variety of different questions, which will help maintain their interest in the assessment. Furthermore, asking the learner 15 different questions also helps provide enough information for the next component in the adaptive learning system, which will use the learner's answers to questions to update the knowledge graph. Thereafter, a GUI was created to present the learner with the questions and store their answers to the questions in a custom object.

## 4.5 EXPERIMENT DESIGN

An evaluation of the proposed AQG algorithm was conducted to determine whether the research aims have been met. The hypothesis of the evaluation is that participants will be satisfied with the grammaticality, understandability and adaptability of the algorithm. However, participants will find that the generated questions are not very difficult. Therefore, the evaluation was designed to either prove or disprove our hypothesis. The relevant ethics clearance was obtained to conduct the evaluation however, no sensitive data would be gained from the evaluation. The purpose of the user evaluation was to allow the participant to interact with the AQG algorithm, and to provide feedback regarding their experience.

There were two different types of evaluations that took place, including a systematic evaluation through Python code and a user evaluation. The systematic evaluation, which consisted of white-box testing, was conducted to identify whether the questions were adapting to the different learner knowledge models that were used during the evaluation. Therefore, a script was created to determine the worst three concepts in the knowledge model, and then print out how many of the $x$ presented questions related to each of those concepts. Furthermore, the difficulty of each question was also output so as to ensure that the AQG algorithm was strictly presenting difficult questions to learners who were capable of answering them. This systematic evaluation produced quantitative evidence that the AQG algorithm was indeed adapting to the knowledge model and asking questions relating to the worst three subjects of a learner.

The user evaluation was conducted to identify whether the research aim has been met. The research aim was to analyze existing AQG algorithms, and to then devise a new AQG algorithm that sought to improve the grammaticality, understandability, adaptability and suitability of the generated questions. This means that the user evaluation sets out to identify whether the presented questions were grammatically correct, understandable, adaptable and suitable to a learner's capabilities. The process of the user evaluation consisted of creating two independent learner knowledge models that would be used during the evaluation process to showcase the adaptability and suitability aspects. These two learner knowledge

models were standardized meaning that every participant of the evaluation used the same knowledge model, so as to ensure consistency of results. The first and second learner knowledge models can be seen in Appendix A and B, respectively. The concepts included in each knowledge model were selected to provide the learner with a mix of common foods, as well as foods that they may not have been familiar with. This helped simulate realism within the user evaluation as the learner would not know all of the foods in a real-world environment. Furthermore, the knowledge models made sure to simulate scenarios where the learner is not proficient in a concept, and scenarios when they are proficient in a concept. This resulted in the participant receiving questions that are both normal in difficulty, as well as difficult in difficulty.

The participant would interact with the AQG algorithm by answering the $x$ number of questions generated for each learner knowledge model. The participant would be told beforehand what was included in the knowledge model so that they knew what type of questions to expect, therefore being able to evaluate the adaptability of the AQG algorithm. Furthermore, during the process of answering the questions, the participant would take note of any questions that were not grammatically correct or understandable. After answering the questions in both iterations of the AQG algorithm, the participant would answer a short survey. The survey was deliberately created to be less than 5 minutes long so that it only focused on questions that provided important information. Important information is considered to be any information that directly relates to one of the four metrics: grammaticality, understandability, adaptability and suitability. The survey questions tested some common NLG evaluation metrics, such as grammaticality and understandability [5]. Furthermore, the survey clearly defined the metrics so that the participants clearly understood what they were evaluating within the questions. This is an important consideration when evaluating NLG questions, as there is no standardized method of evaluating the quality of NLG questions [5].

There were no strict requirements to be considered for the user evaluation. Any individual who can speak English and was able to physically interact with the AQG component in-person would be a potential candidate for the user evaluation. The participant did not require any knowledge beforehand, as the primary concern of the evaluation was to assess the grammaticality, understandability, adaptability and suitability of the questions.

The survey was intentionally designed to be quick for two reasons. The first of which was to ensure that it was focused on only identifying important information, and secondly it was to encourage participation in the evaluation. A brief description of the questions and what information it set out to assess will be listed. The following questions assessed the grammaticality of the questions: "Did any of the questions contain any grammatical errors?", "Which questions did you notice to have any grammatical errors?". The following questions assessed the understandability of the questions: "Did the questions use language that was appropriate for your level of knowledge and understanding?", "How confident did you feel in your ability to understand and respond to the questions?", "Which questions did you struggle to understand?". The following questions assessed the adaptability of the questions: "Were there instances where the adaptation seemed inaccurate or mismatched to your learning path?", "How satisfied were you with the adaptability of the questions for your learning needs?". The following questions assessed the suitability of the questions: "On a scale of 1-5 (with 5 being most difficult), how difficult were the difficult questions?", "How satisfied were you with the suitability of the questions for your learning needs?". The final question "Are there any suggestions that you would make to improve the quality of the generated questions?" did not focus on any specific metric, but rather aimed to gather important feedback for future work for the research.

## 5 RESULTS

The systematic evaluation that was conducted through a Python script was run with many different learner knowledge models. This ensured that the evaluation covered a large range of cases, to ensure that an accurate description of the adaptivity levels of the AQG algorithm was produced. The purpose of the systematic evaluation was to evaluate how many questions were being generated, whether the AQG algorithm was accurately adapting the selected questions to a learner's knowledge model, and the amount of time that it took to execute the algorithm for varying knowledge model sizes. Table 1 displays the total number of questions generated for each question template. In Table 2, each row stores the results obtained after the evaluation using a knowledge model of size $l$, where $l$ is the first column in the table. Thereafter, the next column displays the total number of distinct concepts that appear in the questions. The last three columns each display the total number of questions that relate to the worst, second worst, and third worst concepts respectively. Table 3 in Appendix C shows the execution time of the AQG algorithm for multiple knowledge models. Appendix D and F show the questions generated using the knowledge models found in Appendix A and B, respectively.

**Table 1: The total number of questions generated for each question template, as well as the total number of questions generated by the AQG**

| Question Template | No. of Generated Questions |
|---|---|
| 1 | 45 |
| 2 | 54 |
| 3 | 33 |
| 4 | 9 |
| 5 | 17 |
| 6 | 37 |
| 7 | 50 |
| 8 | 37 |
| 9 | 3 |
| 10 | 7 |
| 11 | 43 |
| 12 | 47 |
| Total | 382 |

**Table 2: Results displaying the total number of concepts that appear in the presented questions, as well as how many of these questions relate to the worst, second-worst, and third-worst concepts**

| Size | Learner Knowledge Model | No. Different Concepts | Worst Concept | Second Worst Concept | Third Worst Concept |
|---|---|---|---|---|---|
| 0 | 13 | 0 | 0 | 0 | 0 |
| 1 | 8 | 7 | 0 | 0 | 0 |
| 1 | 9 | 7 | 0 | 0 | 0 |
| 1 | 9 | 5 | 0 | 0 | 0 |
| 2 | 3 | 5 | 7 | 0 | 0 |
| 3 | 3 | 6 | 2 | 7 | 0 |
| 21 | 4 | 7 | 4 | 3 | 0 |
| 21 | 3 | 6 | 6 | 3 | 0 |

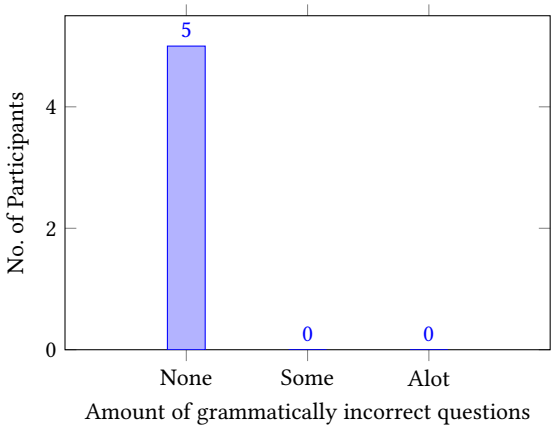


**Figure 2: Results of participants rating whether the generated questions contained any grammatical errors. None of the participants found grammatical errors with the questions.**

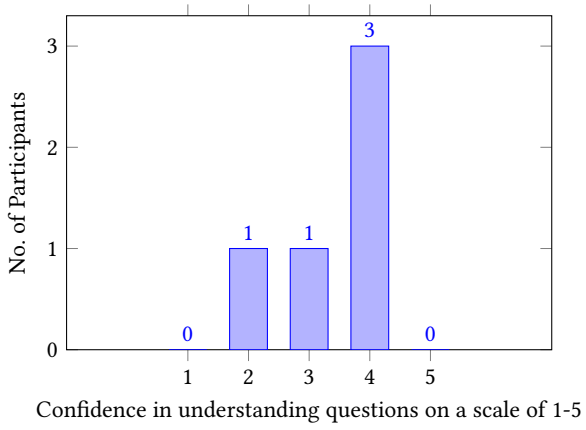


**Figure 4: Results of participants rating how confident they were in understanding and answering the questions on a scale of 1-5, where 1 is extremely not confident and 5 is extremely confident**

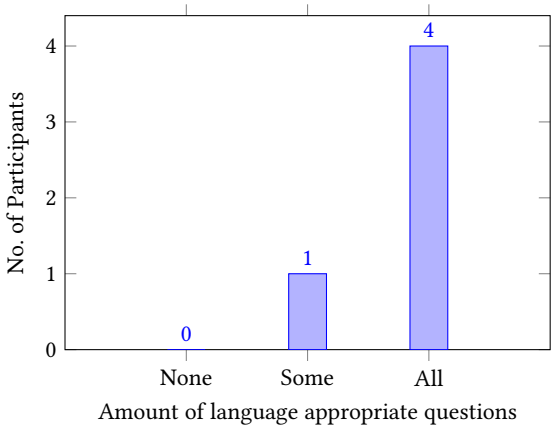


**Figure 3: Results of participants rating whether the language of the questions was appropriate for their level of understanding. 4 out of the 5 participants stated that all of the questions used appropriate language.**

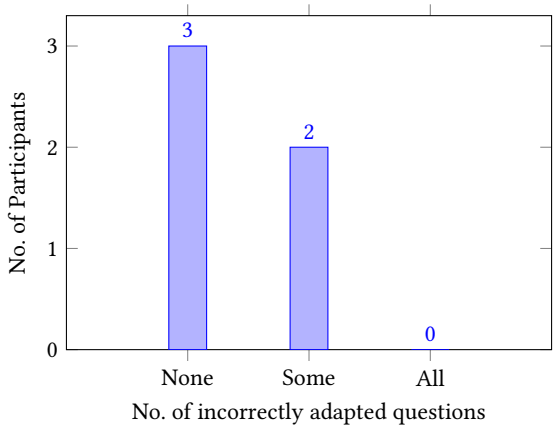


**Figure 5: Results of participants rating whether they felt that there were questions where the adaptation was incorrect. All of the participants noticed some form of adaptation, with 60% of participants stating that all questions were relevant**

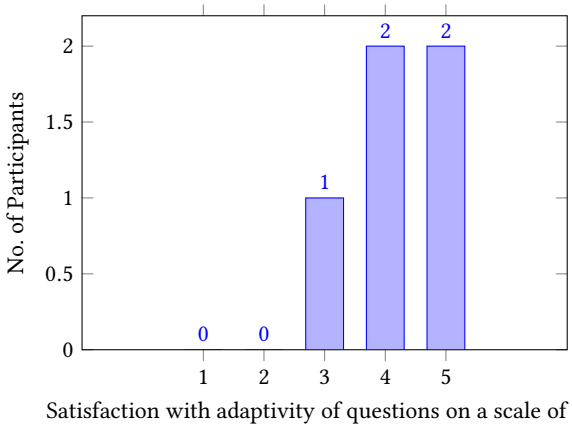


**Figure 6: Results of participants rating how satisfied they were with the adaptivity of the generated questions on a scale of 1-5, where 1 is very dissatisfied and 5 is very satisfied**

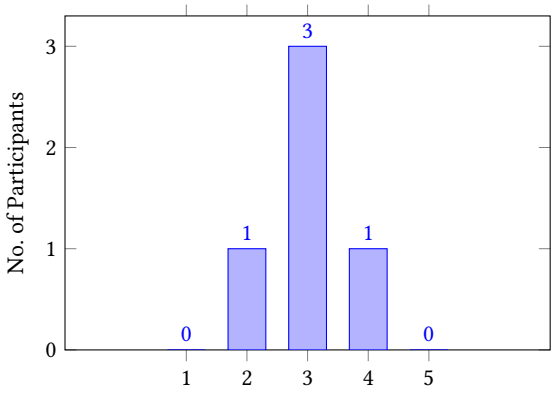


**Figure 7: Results of participants rating how difficult the difficult questions were (between 1-5 where 5 is the hardest)**

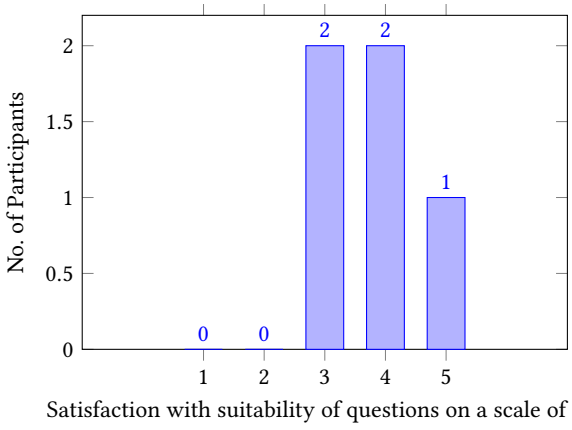


**Figure 8: Results of participants rating how satisfied they were with the suitability of the questions on a scale of 1-5, where 1 is very dissatisfied and 5 is very satisfied**

## 6 DISCUSSION

The system was systematically evaluated to identify how many questions were generated per iteration for each question template. The results of this evaluation can be seen in Table 1. The results show that the lowest number of questions generated for a template was 3 different questions, whereas the highest number of questions generated for a template was 54. This large discrepancy is due to the content of the ontology, as the questions are generated for a template only if it satisfies the criteria by containing the required predicates. As an example, if the ontology does not contain information about the ingredients of a particular food, then that question template regarding ingredients would not generate a question for that food. The total number of questions generated was 382 questions, which is a significant number considering the limited size of the ontology. This indicates that the AQG algorithm would work on ontologies of all sizes, which is beneficial due to the high costs and time-consuming nature associated with ontology engineering.

The results obtained from the systematic evaluation, seen in Table 2, showed that the newly created AQG algorithm successfully adapted to the learner knowledge model. The baseline for the systematic evaluation was when the learner knowledge model was empty, as this allowed us to see how many different concepts would be present in the selected questions. As can be seen in Table 2, this baseline proved to be an amount of 13 different concepts, meaning that when the learner knowledge model is empty, the AQG algorithm generates questions that test 13 different concepts. This was the intention of the AQG algorithm as the goal of the first iteration was to test as many different concepts as possible, so as to gauge what the learner's weak points and strong points are. It is only once this is known that the system should adapt and tailor questions to a specific learner. The results, shown in Table 2, also prove that as the learner knowledge model grows, the total number of concepts that appear in the questions diminishes. The results show that when the learner knowledge model includes only one concept, there is an average of nine different concepts tested in the questions. The results also show that when the learner knowledge model includes two or three concepts, an average of three different concepts are tested. Finally, when the learner knowledge model includes 21 concepts, an average of three different concepts are tested. This is the expected result as it showcases that the AQG algorithm is successfully adapting and focusing specifically on a learner's three worst concepts, when compared with the baseline model.

The total amount of time, in seconds, that it took to execute the algorithm for learner knowledge models of varying sizes can be seen in Appendix C. The results from the experiment show that the AQG has succeeded in executing quickly without requiring a large amount of information. This is achieved through the approach that was employed when designing the algorithm, which is ensuring modularity between the different functions. The results show that the algorithm execution time does not exceed 0.1 seconds, regardless of the size of the learner knowledge model. This is achieved through the distinct separation of the question generation and question selection processes.

The user evaluation aimed to specifically evaluate the grammaticality, understandability, adaptability and suitability of the

questions for a given learner. Therefore, the questions within the survey aimed to uncover exactly what a participant's thoughts were regarding these metrics.

Figure 2 shows that participants did not find any problems regarding the grammaticality of the generated questions, as all the participants of the survey stated that none of the questions contained any grammatical errors. Figure 3 and Figure 4 showcase the results assessing the understandability of the generated questions. The survey results show that 80% of participants found that the questions were appropriate for their level of understanding and were easily interpretable. 60% of participants stated that they were somewhat confident in answering the questions and were not confused by their phrasing. However, the survey results did also identify that 40% of participants were either neutral or somewhat unconfident when answering the questions. Upon closer inspection, 40% of participants did not understand some questions that contained uncommon foods. However, it is important to emphasize that the structure of the question did not confuse the participant, but rather the food as they had not heard about it before. This is not a cause for concern as the evaluation is of the standalone AQG algorithm. The AQG algorithm is intended to be integrated into an adaptive learning system, which means that participants would have received learning materials about the foods that they are confused about, if they were to interact with the complete adaptive learning system.

Regarding the adaptability aspect of the questions, Figure 5 and Figure 6 show that most participants were satisfied with the adaptability level of the AQG algorithm and did not feel that questions were randomly being asked to them. 60% of participants could sense that the questions were following a structured path and saw that the questions dramatically changed when the learner knowledge model changed. However, 40% of participants stated that they were asked one or two questions that did not contain concepts within the knowledge model. This is not a cause for concern as the algorithm does have an element of lightly weighted randomness to it, whereby one or two random questions will be generated for a learner that are not in the learner's knowledge model. Regarding the suitability of the questions, Figure 7 and Figure 8 show that most participants did not find the difficult questions to be difficult. The participants rated the difficulty of the difficult questions a score between two and four, with an average score of three. 60% of participants rated the difficulty a score of three, which implies that they were not effectively challenged by the questions. This is due to the structure of the question templates; however, it introduces the potential for future work in which newer question templates could be introduced that contained more predicates for a particular subject. Thereby, increasing the difficulty of the difficult questions. Figure 8 shows that only 20% of participants were very satisfied with the suitability of the questions, with 40% of participants feeling somewhat satisfied and another 40% feeling neutral. This further emphasized the need to increase the difficulty of the generated questions in order to more effectively challenge a learner on the learning materials.

## 7 LIMITATIONS

A limitation is that the created AQG algorithm is dependent on the contents and structure of the ontology. Thus, a small ontology would produce a limited number of questions that contain a limited number of foods along with its predicates. This limits the number of questions that may be generated, as well as the types of questions that may be generated, due to the template-based approach utilized.

The template-based approach also results in further limitations for the research, as it means that the generated questions will always be in English. Further templates would have to be created in different languages if the AQG was to be used in a foreign adaptive learning system. Furthermore, the number of participants involved in the evaluation process was limited. Therefore, the results may not truly represent the larger population as a whole, which may result in misleading conclusions.

## 8 CONCLUSIONS

To conclude, the results produced from the evaluations assisted in determining whether the metrics stated in the research aim have been met or not. The systematic evaluation showed that the questions adapted to the learner knowledge model and followed a specific and tailored learning path. This assumption was further corroborated by the user evaluation whereby participants could clearly identify that they were on a specific and tailored learning path. Our evaluation showed that the questions were both grammatically correct and understandable. These are promising results as several existing AQG algorithms struggled from the generation of questions that were either not grammatically correct or difficult to understand. However, we were also able to conclude that the suitability of the questions generated by the AQG were not suitable to all participants. Although there were different question templates that had different difficulty levels, participants of the survey did not find the difficult questions to be difficult. They were still able to answer the questions easily as long as they knew what the food item was.

Our evaluation showed that the inclusion of the AQG in an adaptive learning system would be of benefit and would improve the entire adaptive learning system. This is true for ontologies that are smaller in size, as our AQG does not require a large amount of information to adaptively generate questions for a learner.

## 9 FUTURE WORK

The evaluation has shown promising results, as well as potential for future work. The newly created AQG algorithm has created questions that are grammatically correct, understandable, and adaptable. However, the generated questions are not suitable for the difficulty levels or capabilities of participants. Future work could consist of increasing the number of question templates available, especially more templates of a difficult nature. Furthermore, newer question templates could make use of more predicates to increase the difficulty of the questions. The addition of other question types, such as MCQ or Wh-questions could be introduced to supplement the true or false questions currently generated. Furthermore, the size of the ontology may be increased to include more food items as well as more relationships. These new food items would automatically fill in the current question templates however, if new relationships are to be added to the ontology, then new question templates would need to be created.

## REFERENCES


[1] Alsubait, T., Parsia, B., and Sattler, U. Automatic generation of analogy questions for student assessment: An ontology-based approach. *Research in learning technology 20*, sup1 (2012), 95–101.

[2] Cubric, M., and Tosic, M. Design and evaluation of an ontology-based tool for generating multiple-choice questions. *Interactive technology and smart education 17*, 2 (2020), 109–131.

[3] Divate, M., and Salgaonkar, A. Automatic question generation approaches and evaluation techniques. *Current science (Bangalore) 113*, 9 (2017), 1683–1691.

[4] Gärtner, M., Rauber, A., and Berger, H. Bridging structured and unstructured data via hybrid semantic search and interactive ontology-enhanced query formulation. *Knowledge and information systems 41*, 3 (2014), 761–792.

[5] Howcroft, D. M., Belz, A., Clinciu, M., Gkatzia, D., Hasan, S. A., Mahamood, S., Mille, S., Van Miltenburg, E., Santhanam, S., and Rieser, V. Twenty years of confusion in human evaluation: Nlg needs evaluation sheets and standardised definitions. In *13th International Conference on Natural Language Generation 2020* (2020), Association for Computational Linguistics, pp. 169–182.

[6] Ibrahim Teo, N. H., and Joy, M. S. Validation of course ontology elements for automatic question generation. *EAI Endorsed Transactions on e-Learning 4*, 15 (2017), 1–7.

[7] Khodeir, N., Wanas, N., Darwish, N., and Hegazy, N. Bayesian based adaptive question generation technique. *Journal of Electrical Systems and Information Technology 1*, 1 (2014), 10–16.

[8] Kuhrmann, M., Tell, P., Hebig, R., Klunder, J. A.-C., Munch, J., Linssen, O., Pfahl, D., Felderer, M., Prause, C., Macdonell, S., Nakatumba-Nabende, J., Raffo, D., Beecham, S., Tuzun, E., Lopez, G., Paez, N., Fontdevila, D., Licorish, S., Kupper, S., Ruhe, G., Knauss, E., Ozcan-Top, O., Clarke, P., Mc Caffery, F. H., Genero, M., Vizcaino, A., Piattini, M., Kalinowski, M., Conte, T., Prikladnicki, R., Krusche, S., Coskuncay, A., Scott, E., Calefato, F., Pimonova, S., Pfeiffer, R.-H., Pagh Schultz, U., Heldal, R., Fazal-Baqaie, M., Anslow, C., Nayebi, M., Schneider, K., Sauer, S., Winkler, D., Biffl, S., Bastarrica, C., and Richardson, I. What makes agile software development agile. *IEEE transactions on software engineering 48*, 9 (2022), 1–1.

[9] Kumar, A. P., Nayak, A., K, M. S., Chaitanya, C., and Ghosh, K. A novel framework for the generation of multiple choice question stems using semantic and machine-learning techniques. *International journal of artificial intelligence in education* (2023).

[10] Kurdi, G., Leo, J., Parsia, B., Sattler, U., and Al-Emari, S. A systematic review of automatic question generation for educational purposes. *International journal of artificial intelligence in education 30*, 1 (2020), 121–204.

[11] Kusuma, S. F., Siahaan, D. O., and Fatichah, C. Automatic question generation with various difficulty levels based on knowledge ontology using a query template. *Knowledge-based systems 249* (2022), 108906–.

[12] Papasalouros, A., Kanaris, K., and Kotis, K. Automatic generation of multiple choice questions from domain ontologies. *e-Learning 1* (2008), 427–434.

[13] Pelanek, R. Applications of the elo rating system in adaptive educational systems. *Computers and education 98* (2016), 169–179.

[14] Sai, A. B., Mohankumar, A. K., and Khapra, M. M. A survey of evaluation metrics used for nlg systems. *ACM computing surveys 55*, 2 (2023), 1–39.

[15] Susanti, Y., Tokunaga, T., and Nishikawa, H. Integrating automatic question generation with computerised adaptive test. *Research and practice in technology enhanced learning 15*, 1 (2020), 1–22.

[16] Wang, S., Christensen, C., Cui, W., Tong, R., Yarnall, L., Shear, L., and Feng, M. When adaptive learning is effective learning: comparison of an adaptive learning system to teacher-led instruction. *Interactive learning environments 31*, 2 (2023), 793–803.

[17] Yahya, M. A. Ontology-based multiple choice question generation. *TheScientificWorld 2014* (2014), 274949–9.

[18] Žitko, B., Stankov, S., Rosić, M., and Grubišić, A. Dynamic test generation over ontology-based knowledge representation in authoring shell. *Expert Systems with Applications 36*, 4 (2009), 8185–8196.


## A SUPPLEMENTARY INFORMATION

An extensive design process was conducted to ensure that a fully operational AQG component was completed, that satisfied all of the research aims. A holistic understanding of the entire adaptive learning system was necessary before designing the individual AQG component, as the AQG component is tightly coupled with the separate component that maintains an updated learner knowledge model. Thus, only once an agreed upon input and output between the components was found, would the AQG component be designed.

Thereafter, a high-level solution to the AQG algorithm was devised. The solution included creating multiple question templates which would create and store generated questions, for foods or countries that satisfied the template, into an object which would be stored in a list of that object. Thereafter, the AQG algorithm should select $x$ questions from the list that were adaptively selected based on a learner's learner knowledge model. The selected $x$ questions would be presented to a learner and the answer to each of the questions would be stored in an object, with each object stored in a list of that object. This final list of objects would be the agreed-upon output to the next component.

Once the high-level solution was devised, a class diagram was designed to understand the different classes within the AQG component, as well as how they would interact with one another. This can be seen in Appendix F. Each generated question would be stored in a *questionForLearner* object, which contained: concept, question, memo answer and difficulty level as the fields. The concept is the subject of the question and the question field stores the generated question in a human-readable form. The memo answer is the correct answer (true or false) and the difficulty level stores the level of difficulty of the question, where 0 represents normal and 1 represents difficult. When a learner answers a question, their answer will be stored in a *learnerAnswer* object. The fields of the *learnerAnswer* class are: concept, answer, memo. The answer field stores the learner's answer to the question, which will be either True or False. Therefore, you would be able to compute whether the learner answered the question correctly using this object, as you would compare the answer and memo fields to each other. If they are the same, the learner answered the question correctly.

Questions are generated according to multiple predefined templates. An example of the process involved in generating questions for a specific template will be given below. The example will consist of generating questions asking whether a food item has a specific ingredient. The template of the question is "Is $zy$ an ingredient of $y$" where $y$ is a food item and $z$ is an ingredient. The question generation process will begin by looping through each food item stored locally. Thereafter, an ingredient of that specific food item will be fetched and stored. If multiple ingredients exist for that specific food item then a single random ingredient will be fetched from the total number of correct ingredients for that specific food item. Following this, a random ingredient will be fetched from the local list that stores all of the ingredients within the ontology. Both the correct and random ingredients will then be appended to a new list. At this stage in the process, the learner's ability for that food item will need to be read in from the learner knowledge model, if it exists. After reading in the learner's ability, either the correct or random ingredient needs to be selected as the final ingredient that will form part of the question. In other words, you would need to select whether the correct or random ingredient that would replace the $z$ defined in the template earlier. If the learner's ability for that food is either non-existent or is above or equal to 0, then there is a 60% chance that the correct ingredient is selected and a 40% chance that the random ingredient is selected. If the learner's ability for that food is below 0 then there is a 40% chance that the correct ingredient is selected and a 60% chance that the random ingredient is selected. These weightings are based on the premise that false questions require a learner to be more knowledgeable about the

food. Thus, learner's who struggle with the food, those having a learner ability below 0, have a higher likelihood of receiving a false question. This will force the learner to use their knowledge to correctly answer the question and recall whether that food item actually has that specific ingredient as an ingredient.

## B APPENDIX A

ChickenBurger,hasDishVariation,TurkeyBurger,-1.45,None
ChickenBurger,hasDishVariation,UnitedStates,-1.45,0.
GardenSalad,hasIngredient,Lettuce,-1.4,None
BlueberryPancakes,hasIngredient,Butter,1.4,None
BeefStew,hasIngredient,Garlic,-0.142,None
BeefKwayTeow,hasIngredient,OysterSauce,1.347,None
AsamPedas,isEatenInCountry,NeighboringSoutheastAsianCountries,-1.732,None
ArrabbiataSauce,isEatenInCountry,Rome,-0.64,None
BakewellPudding,course,ItalianMealStructure,1.37,None

## C APPENDIX B

BaconExplosion,hasMainIngredients,Sausage,-0.732,None
BaconExplosion,hasMainIngredients,Bacon,-0.732,None
BaconExplosion,isEatenInRegion,KansasCityMetropolitanArea,-0.732,None
BaconExplosion,course,MainCourse,-0.732,None
BaconExplosion,hasIngredient,Sausage,-0.732,None
BaconExplosion,hasIngredient,Bacon,-0.732,None
BaconExplosion,isEatenInCountry,.UnitedStates,-0.732,-0.268
BarnyCakes,hasIngredient,SpongeCake,0.12,None
BarnyCakes,isCreatedBy,MondelezInternational,0.12,None
BarnyCakes,hasDishVariation,Apple,0.12,None
BarnyCakes,hasDishVariation,Chocolate,0.12,None
BarnyCakes,isEatenInCountry,France,0.12,-2
UnitedStates,hasLanguage,EnglishLanguage,-0.268,None
UnitedStates,hasCapital,WashingtonD.C.,-0.268,None
UnitedStates,hasLeaderName,joeBiden,-0.268,None
UnitedStates,hasLeaderName,JohnRoberts,-0.268,None
UnitedStates,hasLeaderName,PaulRyan,-0.268,None
UnitedStates,hasLeaderName,BarackObama,-0.268,None
Bhajji,course,MainCourse,-0.571,None
AsamPedas,course,MainCourse,0.118,None
Bakso,hasIngredient,Noodle,0.3,None

## D APPENDIX C

**Table 3: The total execution time of the algorithm for different knowledge model sizes**

| Distinct Concepts in Knowledge Model | Elapsed Time (in seconds) |
|---|---|
| 0 | 0.08 |
| 5 | 0.09 |
| 10 | 0.1 |
| 15 | 0.1 |
| 20 | 0.1 |

## E APPENDIX D

**Table 4: Questions generated for Learner Knowledge Model found in Appendix A**

| No. | Question |
|---|---|
| 1. | Chicken Burger is eaten in United States |
| 2. | Garden Salad is eaten in Nationwide In Singapore |
| 3. | Washington D. C. is the capital city of United States |
| 4. | Chicken Sandwich is an alternative name for Chicken Burger |
| 5. | Garden Salad is eaten for Dessert |
| 6. | Tuna is a main ingredient of Chicken Burger |
| 7. | Tomato is an ingredient of Chicken Burger |
| 8. | B L T is a variation of Chicken Burger |
| 9. | Mixed Greens Salad is an alternative name for Garden Salad |
| 10. | Risotto Ai Funghi is an alternative name for Asam Pedas |
| 11. | Waldorf Salad is a variation of Garden Salad |
| 12. | Various Vegetables is a main ingredient of Garden Salad |
| 13. | Vegetable Broth is an ingredient of Garden Salad |
| 14. | Beef Kway Teow is eaten in Nationwide In Singapore and has Palm Sugar as an ingredient |
| 15. | Asam Pedas is eaten in Philippines |

## F APPENDIX E

**Table 5: Questions generated for Learner Knowledge Model found in Appendix B**

| No. | Question |
|---|---|
| 1. | Jakarta is the capital city of United States |
| 2. | Bacon Explosion is eaten as an Appetizer |
| 3. | Citizens in United States speak Philippine |
| 4. | Bacon is an ingredient of Bacon Explosion |
| 5. | Bacon Explosion is eaten in Hong Kong |
| 6. | Sausage and Guanciale are both ingredients of Bacon Explosion |
| 7. | Blueberries is a main ingredient of Bacon Explosion |
| 8. | Bhajji is eaten in India |
| 9. | Silvano Aureoles Conejo is or has been the leader of the United States |
| 10. | Bacon Explosion is eaten in the United States and has Butter as an ingredient |
| 11. | Beef Broth is an ingredient of Bhajji |
| 12. | Bacon Explosion is eaten in United States |
| 13. | Bhajji is eaten in Philippines and has Gram Flour as an ingredient |
| 14. | Bhajji is eaten in Karnataka |
| 15. | Bhaji is an alternative name for Bhajji |

## G APPENDIX F

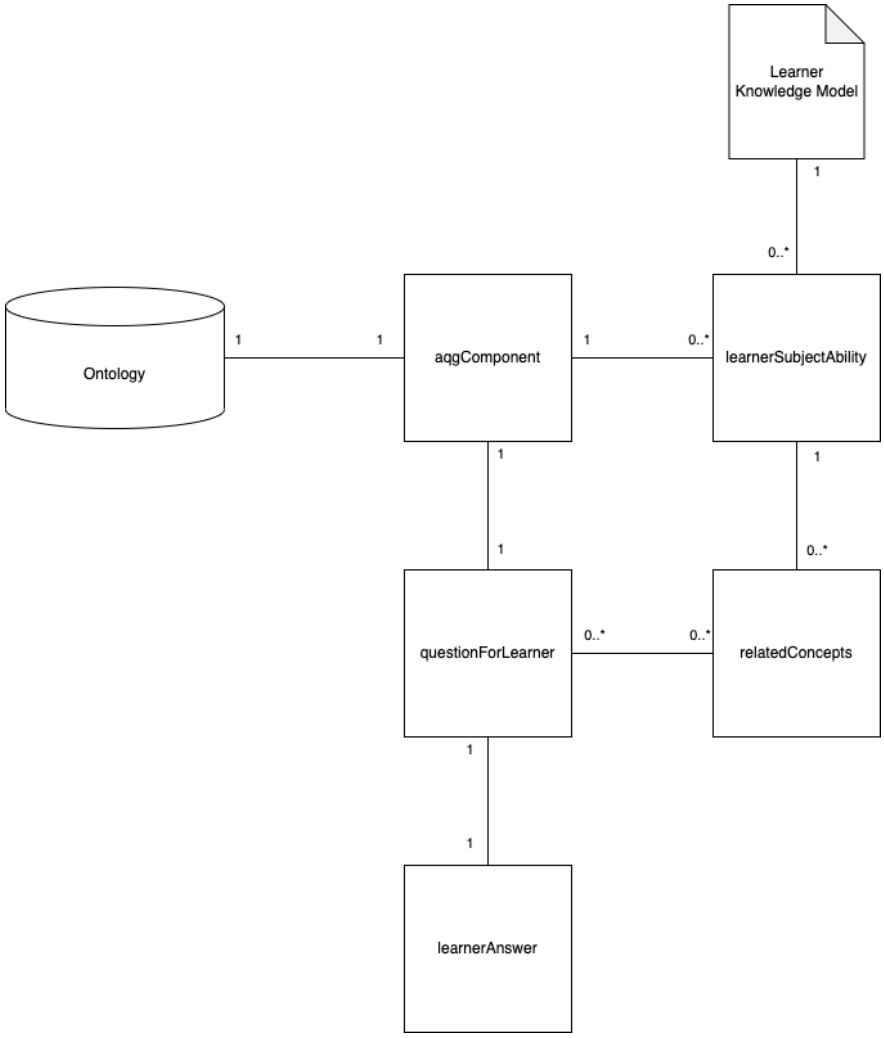


**Figure 9: Class diagram of AQG component**