ONTOLOGY VALIDATION & UTILISATION FOR PERSONALISED FEEDBACK IN EDUCATION

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS OF
BIRMINGHAM CITY UNIVERSITY
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF COMPUTING, ENGINEERING AND THE BUILT ENVIRONMENT

2018

By
Mona Nabil Demaidi
School of Computing and Digital Technology

Contents

List of Tables

List of Figures

Abstract

Virtual Learning Environments provide teachers with a web-based platform to create different types of feedback which vary in the level of details given in the feedback content. Types of feedback can range from a simple correct or vice-versa to a detailed explanation about the reason why the correct answer is correct and the incorrect answer is incorrect. However, these environments usually follow the ‘one size fits all’ approach and provide all students with the same type of feedback regardless of students’ individual characteristics and the assessment question’s individual characteristics. This approach is likely to negatively affect students’ performance and learning gain.

Several personalised feedback frameworks have been proposed which adapt the different types of feedback based on the student characteristics and/or the assessment question characteristics. The frameworks have three drawbacks: firstly, creating the different types of feedback is a time consuming process, as the types of feedback are either hard-coded or auto-generated from a restricted set of solutions created by the teacher or a domain expert; secondly, they are domain dependent and cannot be used to auto-generate feedback across different educational domains; thirdly, they have not attempted any integration which takes into consideration both the characteristics of the assessment questions and the student’s characteristics.

This thesis contributes to research carried out on personalised feedback frameworks by proposing a generic novel system which is called the Ontology-based Personalised Feedback Generator (OntoPeFeGe). OntoPeFeGe has three aims: firstly, it uses any pre-existing domain ontology which is a knowledge representation of the educational domain to auto-generate assessment questions with different characteristics, in particular, questions aimed to assess students at different levels in Bloom’s taxonomy[1]; secondly, it associates each auto-generated question with specialised domain independent types of feedback; thirdly, it provides students with personalised feedback which adapts the types of feedback based on the student and the assessment question characteristics. OntoPeFeGe allowed the integration of student’s characteristics, the assessment question’s characteristics, and the personalised feedback, for the first time. The

[1] Bloom’s taxonomy categorises the assessment questions into the following six major levels, which are arranged in a hierarchical order according to the complexity of the cognitive process involved: knowledge, comprehension, application, analysis, synthesis, and evaluation.

experimental results applying OntoPeFeGe in a real educational environment revealed that the personalised feedback particularly improved the performance of students with initial low background knowledge. Moreover, the personalised feedback improved students' learning gain significantly at questions designed to assess the students at high levels in Bloom's taxonomy. In addition, OntoPeFeGe is the first prototype to quantitatively analyse the quality of auto-generated questions and tests, and to provide question design guidance for developers and researchers working in the field of question generators.

OntoPeFeGe could be applied to any educational field captured in an ontology. However, assessing how suitable the ontology is for generating questions and feedback, as well as how it represents the subject domain of interest, is a necessary requirement to using the ontology in OntoPeFeGe. Therefore, this thesis also presents a novel method termed Terminological ONtology Evaluator (TONE) which uses the educational corpus (e.g., textbooks and lecture slides) to evaluate the domain ontologies. TONE has been evaluated experimentally showing its potential as an evaluation method for educational ontologies.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

I am grateful to all the people who helped and inspired me during the preparation of this thesis, especially my supervisor Prof. Mohamed Medhat Gaber, who gave up his valuable time to help me extend the research work, analyse the experimental results, read my thesis draft, and offered comments that had led to many improvements. Thanks for Birmingham City University staff for providing me with help, support and facilities to carry out my research.

I would also like to thank Dr. Nick Filer who supervised me at the beginning of my PhD until his plans for retirement, and the University of Manchester staff who facilitated the experimental studies.

Words fail me to express my appreciation to my family who provided me with their support, understanding and endless love, through my PhD years. Father Nabil and mother Amneh you are my role model and have always inspired me to become the person I am today. Thank you for believing in me and giving me the chance to accomplish my dreams.

Sisters Lana and Hala, brothers Iyad and Murad, thank you for being always there for me and standing by my side. I am lucky to have an amazing family like you. Finally, I would like to thank my dear friends.

Chapter 1

Introduction

Personalised learning environments are Virtual Learning Environments (VLEs) which tailor the learning content and generate feedback to meet student's knowledge, needs and goals (Tarus et al., 2018; Bishop et al., 2018; Mason and Bruning, 2001; Gouli et al., 2006; Ana and Macario, 2009; Pardo et al., 2017; Mitrovic and Martin, 2000; Narciss and Huth, 2004; Narciss and Huth, 2006; Narciss et al., 2014; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010; Anderson et al., 1995; Mitrovic, 2012). This thesis focuses on one important aspect in personalised learning environments, which is providing students with formative feedback while they are working on assessment tasks (Demaidi and Filer, 2012). Formative feedback is provided to students after answering an assessment task and it is a key element in formative assessment systems (Black and Wiliam, 2009; Hattie and Timperley, 2007; Black and Wiliam, 1998; Sadler, 1989; Vasilyeva et al., 2008b). It provides students with the information required to close the gap between their current performance and the desired performance(Brown and Glover, 2006; Goldin et al., 2017; Keuning et al., 2016). The information provided focuses on the particular qualities of the student work, with advice on what he or she can do to improve, avoiding comparisons with other students(Black and Wiliam, 1998). In addition to the importance of formative feedback content, Price *et al.* specified that feedback can only be effective when the learner understands the feedback and is willing and able to act on it (Price et al., 2010). Formative feedback can be delivered to students immediately or after some delay (Hooshyar et al., 2016; Shute, 2008). This thesis focuses on the immediate formative feedback which students receive after answering an assessment task. Formative feedback provided to students in learning environments can be classified into several types, where each type provides students with different pedagogical content at a different level of

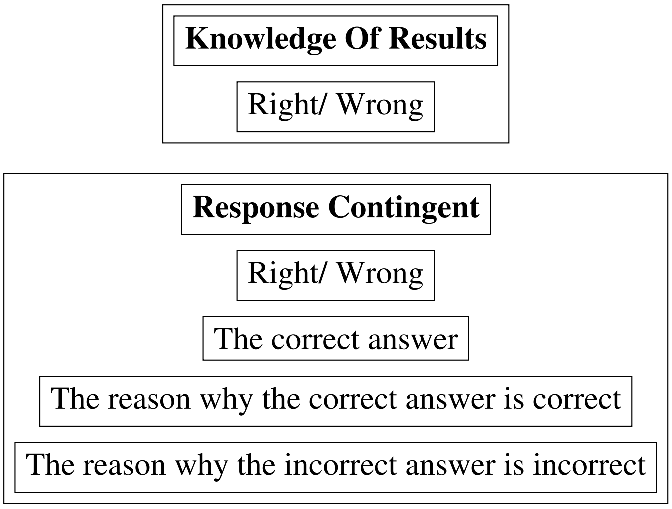

Knowledge Of Results
Right/ Wrong
Response Contingent
Right/ Wrong
The correct answer
The reason why the correct answer is correct
The reason why the incorrect answer is incorrect

Figure 1.1: Types of formative feedback and their pedagogical content

detail (Mason and Bruning, 2001; Narciss et al., 2014; Shute, 2008; Narciss, 2013; Bangert-Drowns et al., 1991). Fig 1.1 illustrates an example of two types of formative feedback and the pedagogical content associated with each type (Mason and Bruning, 2001; Narciss et al., 2014; Shute, 2008; Narciss, 2013; Bangert-Drowns et al., 1991; Mory, 2004). Verification feedback is also called Knowledge Of Results (KOR) feedback and it verifies whether a student's answer is right or wrong. Knowledge of Correct Response (KCR) feedback verifies a student's answer and also provides him or her with the correct answer. Bugs-Related (BR) feedback verifies the student's answer and provides him or her with the reason why an incorrect answer is incorrect without giving the student the correct answer. Topic Contingent (TC) feedback verifies the student's answer, provides him or her with the correct answer, and explains to the student the reason why the correct answer is correct. Response Contingent (RC) feedback has similar pedagogical content to the TC feedback. However, it also explains to the student the reason why the incorrect answer is incorrect. Hint feedback provides the student with information on what to do next to guide him or her towards the right solution.

Providing students with personalised feedback has been identified as a powerful method that helps them understand the gaps in their knowledge, monitor their progress and improve their overall performance (VanLehn, 2011). Personalised feedback is defined as adapting the type of feedback provided to a student based on the student's characteristics (e.g., the background knowledge and the current level of knowledge) and/or the task's characteristics (e.g., the level of the question in Bloom's taxonomy) (Narciss and Huth, 2004; Narciss et al., 2014; Narciss, 2013; Vasilyeva et al., 2008a).

Researchers such as Gouli *et al.* (Gouli et al., 2006), Mason and Bruning (Mason and Bruning, 2001) proposed guidelines to develop personalised feedback frameworks. Gouli *et al.* framework adapted the type of feedback based on students' current level of knowledge. While Mason and Bruning's framework considered students' background knowledge and current level of knowledge as well as the task's difficulty. Both frameworks are theoretical and have never been evaluated on students (see Table 2.1 in Chapter 2, page 17). Other researchers such as Narciss *et al.* (Narciss et al., 2014) and Arroyo *et al.* (Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010) focused on providing students with personalised feedback based on the student's current level of knowledge. Their frameworks were evaluated on students and the results revealed that the personalised feedback improved students' performance. However, their evaluations had contradictory results regarding the impact of personalised feedback on the performance of male and female students. Narciss *et al.* showed that female students had higher performance than male students (Narciss et al., 2014). Arroyo *et al.*, had similar results in one study (Arroyo et al., 2000; Arroyo et al., 2001), however, in another study they carried out no difference in performance was found between male and female students (Arroyo et al., 2011; Woolf et al., 2010).

The personalised feedback evaluation studies mentioned above suggest that there is still no clear understanding regarding the relationship between the student's characteristics, the task's characteristics and the personalised feedback (Mason and Bruning, 2001; Narciss et al., 2014; Arroyo et al., 2011; Shute, 2008). Moreover, none of the personalised feedback frameworks which were evaluated on students considered the task's characteristics in the feedback adaptation process. This issue has been addressed by Narciss *et al.* who suggested considering the task's difficulty while providing students with personalised feedback (Narciss et al., 2014). **Therefore, this thesis aims to evaluate Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001) which adapts the different types of feedback based on the student and the task's characteristics**.

The personalised feedback frameworks in the literature provided students with different types of feedback by either hard-coding the feedback in the system (Ana and Macario, 2009; Pardo et al., 2017; Narciss and Huth, 2004; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010) or auto-generating the feedback from a restricted set of solutions created by the teacher or a domain expert (Mitrovic and Martin, 2000; Narciss et al., 2014). This has two main disadvantages: (1) it is a time consuming process (Kazi et al., 2012; Mitrovi, 1998; Passier and Jeuring, 2004); and (2) the frameworks are domain dependent and cannot be used to auto-generate feedback across different educational domains (Cocea, 2011).

This thesis aims to address the drawbacks mentioned above by proposing a generic framework which auto-generates different types of feedback across different educational domains using a broad knowledge base called ontology. An ontology is a conceptualisation of the domain knowledge in terms of concepts and properties and it captures the concepts in an educational course (Gruber, 1993). It has been used in the past by several feedback generators to generate different types of feedback (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Duboc et al., 2011; Duboc, 2013; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011). Kazi *et al.* generated hint feedback (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013), Sãnchez-Vera *et al.* (Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011) generated KOR and KCR feedback, and Duboc *et al.* (Duboc et al., 2011; Duboc, 2013) generated KCR, BR, and TC feedback (see Section 2.3.3 in Chapter 2, page 29). However, these feedback generators have the following drawbacks:

1. The auto-generated feedback is domain dependent. This means that in addition to the ontology, the generators either use an expert knowledge base which captures the experts' solutions to the problem scenario or human intervention (e.g., domain experts and teachers) to auto-generate the different types of feedback.

2. The auto-generated feedback is not personalised to meet the student or the task characteristics.

Providing students with personalised feedback after auto-generating different types of feedback requires information about the assessment task characteristics. The feedback generators mentioned above hard-coded the assessment tasks, which means that the tasks are only valid in the educational domain they are created in. In addition, the feedback generators did not specify the task characteristics (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Duboc et al., 2011; Duboc, 2013; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011). Both drawbacks hinder providing students with personalised feedback in a generic framework. To address this issue, this thesis investigated several domain independent question generators (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014), which use an ontology to auto-generate several types of questions (true or false, multiple choice, and short answer questions) with different characteristics, in particular, questions aimed to assess students' cognition at different levels in Bloom's taxonomy (knowledge, comprehension, application, and analysis) (Bloom et al., 1956; Krathwohl, 2002). **The ontology-based generation strategies used by the question generators and the stem templates, which are the text stating the question and designed to assess students at different levels in Bloom's taxonomy, were integrated into one system called Ontology-based Personalised Feedback Generator (OntoPeFeGe)**. OntoPeFeGe auto-generates various types of assessment questions and associates each question with different types of feedback auto-generated from ontology. Moreover, OntoPeFeGe provides students with personalised feedback immediately after answering an assessment question by adopting Mason and Bruning's personalised feedback framework.

Prior to the design and implementation of OntoPeFeGe, two preliminary studies were carried out and reported in Appendix A. The first study aimed to examine the importance of building OntoPeFeGe which focuses on the formative feedback students receive immediately after answering an assessment question. The importance of building such a system was examined by assessing whether the formative feedback in VLEs helps students understand and learn the educational material, and whether students use the formative feedback during exam revision. The results revealed that students agreed that the formative feedback helped them understand and learn the educational material and that they used the formative feedback during exam revision. The second study aimed to justify using Mason and Bruning's personalised feedback framework in OntoPeFeGe. Mason and Bruning's personalised feedback framework focuses on providing students with BR, TC, and RC personalised feedback. Therefore, before adopting and evaluating their framework in OntoPeFeGem the second study investigated how frequently teachers in VLEs use BR, TC, and RC types of feedback when the feedback is provided to students immediately after answering an assessment question. The results obtained from the study justified using Mason and Bruning's personalised feedback framework as teachers highly used BR, TC, and RC types of feedback.

OntoPeFeGe could be applied to any educational domain where the concepts and properties are captured in an ontology. However, assessing how suitable the ontology is for generating questions and feedback, as well as how it represents the subject domain of interest, is a necessary requirement to using the ontology in OntoPeFeGe. This issue has not been addressed by the question and feedback generators mentioned above.

The questions and the different types of feedback are generated without assessing the conceptual coverage of the underlying ontology, and the level of details an ontology captures about each concept which is called 'the semantic richness'.

Several approaches have been proposed for evaluating ontologies and selecting the appropriate ontology for specific applications (Brank et al., 2005; Maedche and Staab, 2002; Lozano-Tello and Gmez-Prez, 2004; Porzel and Malaka, 2004; Brewster et al., 2004; Jonquet et al., 2010; Martnez-Romero et al., 2012; Martnez-Romero et al., 2014; Rospocher et al., 2012; Jones and Alani, 2006). These approaches focused on the ontology coverage of concepts in the domain of interest, and ignored the semantic richness associated with each concept such as the number of parent concepts, the number of children concepts, and additional information contained within each concept. Martĩnez-Romero *et al.* (Martnez-Romero et al., 2012; Martnez-Romero et al., 2014) addressed this issue by developing a system which evaluates ontologies by assessing their coverage and semantic richness based on a set of input terms that capture the subject domain of interest. However, their system had the following three drawbacks:

1. The domain ontologies are evaluated against a set of input terms defined by the user and cannot be evaluated against the educational course corpus.

2. The input terms used to evaluate the candidate domain ontologies are assumed to have the same weight in the domain of interest. Each weight reflects how important the term is to the domain of interest.

3. The ontology coverage metric used by Martĩnez-Romero *et al.* tends to select large ontologies with broad scope (e.g., OpenCyc which covers several domains such as mathematics, physics, and medicine (Matuszek et al., 2006)) rather than ontologies specifically developed to capture a particular domain of interest (e.g., Sakathi's Computer Networks ontology (Murugan.R et al., 2013)). Therefore, using Martĩnez-Romero *et al.* metric to evaluate the coverage of the candidate domain ontologies may result in generating questions and feedback which are not related to the domain of interest.

This thesis presents a Terminological ONtology Evaluator (TONE), which assesses the ontology coverage and semantic richness. TONE supports the ontology evaluation against both the input terms and corpus and considers the terms' weight. The terms' weight was supported by Rospocher (Rospocher et al., 2012) and Jones and Alani's (Jones and Alani, 2006) ontology evaluation approaches. Both approaches extracted terms from the corpus using term extraction tools (e.g., N-gram which extracts sequences of N words from the text (Lopes et al., 2010)) and assigned weight to each extracted term using the 'term frequency-inverse document frequency' recognition algorithm (Salton and Buckley, 1988), which reflects how important the term is to a document in a corpus. Rospocher (Rospocher et al., 2012) and Jones and Alani (Jones and Alani, 2006) applied an arbitrary cut-off point to the list of terms extracted from the corpus without any justification. Applying the same arbitrary cut-off to the lists of terms extracted from the corpus using different term extraction tools and assigned weight using several term recognition algorithms has the following drawbacks:

1. High variance in the ontology coverage and semantic richness evaluation results: The candidate domain ontology could have different coverage (e.g., high and low coverage) of the same corpus when the lists of terms are extracted using different term extraction tools and assigned weights using several term recognition algorithms.

2. No agreement between the rankings of the candidate ontologies evaluation results: The ontology evaluation approaches evaluate the candidate domain ontologies coverage and semantic richness and rank the ontologies based on their coverage and semantic richness scores. Using the same arbitrary cut-off point in the lists of terms extracted from the same corpus may result in having the same candidate domain ontology being ranked the first (highest coverage score) in one list of terms, and ranked the last (lowest coverage score) when another list of terms is used in the ontology evaluation process.

To address the drawbacks mentioned above, TONE dynamically select a cut-off point in the list of terms extracted from the corpus to reduce the number of irrelevant terms used to evaluate the candidate domain ontologies when different term extraction tools and term recognition algorithms are used. **This aims to reduce the variance in the ontology coverage and semantic richness results and improve the agreement between the rankings of the candidate domain ontologies evaluation results.**

TONE is used in this thesis to select a candidate domain ontology, which has high coverage and semantic richness of the educational domain. The selected ontology is used after that by the OntoPeFeGe to auto-generate the assessment questions and feedback.

In addition to the importance of evaluating the candidate domain ontology used by the OntoPeFeGe to auto-generate assessment questions and feedback, it is essential to ensure the quality of assessment questions generated using OntoPeFeGe. As mentioned earlier, OntoPeFeGe associates each auto-generated question with different types of formative feedback, and according to Black and William, it is essential to ensure the quality of both assessment questions and feedback in learning environments (Black and Wiliam, 1998). Previous evaluations of the auto-generated questions were confined to measuring the qualitative satisfaction of domain experts and students (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014). None of the ontology-based question generators evaluated the questions on students to analyse the quality of auto-generated questions by measuring the questions' difficulty and the questions' ability to discriminate between high ability and low ability students. Therefore, this thesis aims to quantitatively evaluate the quality of the ontology-based auto-generated questions for the first time using two widely used statistical methods: the Classical Test Theory (CTT) (Alagumalai and Curtis, 2005; Ding and Beichner, 2009; Doran, 1980; Cohen et al., 2013; Erguven, 2014) and the Item Response Theory (IRT) (Baker, 2001; De Ayala, 2009; Hambleton and Swaminathan, 1985; Hambleton, 1991; Reckase, 2009; Schmidt and Embretson, 2003; Toland, 2014).

1.1 Motivation and Hypotheses

The work presented in this thesis is motivated by the clear limitation in existing personalised feedback frameworks which are domain dependent, i.e., the different types of feedback were either hardcoded or auto-generated from a restricted set of solutions defined by the teacher or the domain expert (Ana and Macario, 2009; Mitrovic and Martin, 2000; Pardo et al., 2017; Narciss and Huth, 2004; Narciss et al., 2014; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010). Moreover, those frameworks that adapt the different types of feedback based on the students and the task characteristics, were never evaluated on students (Mason and Bruning, 2001). Accordingly, this thesis aims to examine the following hypotheses:

- H1: Dynamically adjusting the weight cut-off point in the list of terms extracted from the educational corpus and used to evaluate the candidate domain ontology decreases the variance in the coverage and the semantic richness metrics and increases the agreement between the rankings of the candidate domain ontologies (tested in Chapter 3).

- H2: A system can be built independently of the underlying domain ontology to auto-generate different assessment questions and different types of feedback (tested in Chapter 4).

- H3: Assessment questions auto-generated from domain ontologies and tests formed from these questions have satisfactory quality (tested in Chapter 5).

- H4: The questions' quality is affected by the ontology-based generation strategies used to auto-generate questions, the level of the questions in Bloom's taxonomy which is determined by the stem template (tested in Chapter 5), and/or the type of the questions generated (true and false, multiple choice, and short answer).

- H5: Providing the appropriate type of auto-generated feedback to students after analysing the student's background knowledge, current level of knowledge, and the question's level in Bloom's taxonomy improves students' performance and learning gain (tested in Chapter 6).

- H6: A relationship exists between the student's characteristics, the task's characteristics, and the personalised feedback (tested in Chapter 6).

- H7: Students and teachers are satisfied with the different types of formative feedback auto-generated from domain ontologies (tested in Chapter 6).

1.2 Aims and Objectives

The aim of this thesis is to develop and evaluate a generic novel framework which achieves the following:

1. The framework selects a domain ontology for auto-generating the assessment questions and feedback by evaluating the candidate domain ontologies which capture the concepts in an educational course using the educational course corpus (e.g., textbooks and lecture slides) based on the coverage and semantic richness;

2. uses domain ontologies to auto-generate assessment questions and different types of feedback across different educational domains; and

3. adapts the type of feedback given to students based on students' background knowledge, students' current level of knowledge and the task's characteristics.

1.3 Research Contributions

The main contribution of this thesis is the proposal and analysis of a novel personalised formative feedback generator, OntoPeFeGe (for Ontology-based Personalised Feedback Generator), which auto-generates assessment questions and different types of formative feedback from pre-existing domain ontology, and provides students with the appropriate types of feedback immediately after answering the assessment question. This and a number of other contributions are briefly outlined in this section in the order they appear in the thesis.

Contribution 1 (Ontology Evaluator): A Terminological ONtology Evaluator (TONE) was proposed for evaluating the candidate domain ontologies used to auto-generate assessment questions and formative feedback. TONE decreases the variance and improves the agreement between the ontology evaluation metrics compared with existing terminological ontology evaluation approaches.

Contribution 2 (Ontology-based Generator): A generator which is capable of auto-generating questions and different types of formative feedback from any pre-existing domain ontology was developed. The generator integrates the state of the art ontology-based generation strategies and the stem templates, which aim to assess students at different levels in Bloom's taxonomy. The ontology-based generation strategies are then used to auto-generate questions and associate each question's option with different types of feedback.

Contribution 3 (Integrating Personalised feedback framework in Moodle): Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001) was implemented in Moodle VLE. The different types of feedback auto-generated from the domain ontology were provided to students based on the student and the task characteristics. However, before adopting Mason and Bruning's personalised feedback framework in OntoPeFeGe, a preliminary study (Study 2) was carried out and reported in Appendix A to investigate the types of feedback teachers provided to students immediately after answering an assessment question. The study results justified using the framework in OntoPeFeGe.

Contribution 4 (Assessment Questions Evaluation): Experimental work evaluation of OntoPeFeGe in a real educational environment showed that the questions auto-generated by the Ontology-based Personalised Feedback Generator (OntoPeFeGe) and the assessment tests formed from these questions had medium difficulty which is very close to the value that the test setters are advised to achieve when constructing questions and where the test has the maximum discrimination. In addition, the questions and tests have satisfactory discrimination values which indicate that the questions and tests could effectively discriminate between high ability and low ability students. The experimental work evaluation of OntoPeFeGe also revealed that the ontology-based generation strategies, the level of the question in Bloom's taxonomy (stem templates), and the type of questions affects the question's difficulty and discrimination.

Contribution 5 (Personalised Feedback Evaluation): An experiment was carried out in a real educational environment to study the effect of personalised feedback, which consists of different types of feedback auto-generated from the domain ontology (KOR, KCR, BR, TC, and RC) on students' performance. The effect of some types of feedback auto-generated in the OntoPeFeGe was not studied before. For instance, Kazi *et al.* (Kazi et al., 2013), and Sãnchez-Vera *et al.* (Frutos-Morales et al., 2010; S et al., 2012) evaluated the effect of KOR, KCR, and hint feedback on students' performance (see Section 2.3.3). Their results revealed that the feedback improved students' performance significantly. On the other hand, Duboc *et al.* (Duboc et al., 2011; Duboc, 2013) whose system auto-generated more types of formative feedback (KCR, BR, and TC) did not evaluate the effect of different types of feedback on students performance, and confined their evaluation to students' satisfaction regarding the importance and usefulness of the auto-generated types of feedback. The experiment carried out in this thesis also investigated the relationship between student's characteristics, the task's characteristics (level of the question in Bloom's taxonomy), and the personalised feedback which is auto-generated from the domain ontology. The results revealed that the personalised feedback improved the performance of students with low background knowledge more than students with high background knowledge. Furthermore, the results revealed no difference in the performance of male and female students after receiving personalised feedback. Examining the relationship between the personalised feedback and the task's characteristics revealed that the personalised feedback improved students' learning gain significantly at questions designed to assess students at high levels in Bloom's taxonomy. The experiment also investigated students (48 students) and teachers (three teachers) satisfaction with the auto-generated formative feedback in OntoPeFeGe. The results revealed that 72.92% of the students agreed that the feedback was useful, 68.75% agreed that the formative feedback helped them answer other questions in the assessment test, and 70.83% agreed that the feedback generated was easy to read. Similarly, the teachers agreed that the feedback was easy to read and that the personalised feedback generator provided students with different types of feedback. Moreover, they agreed that the feedback pedagogical content was reasonable and related to the question generated.

1.4 Publications

The findings reported in this thesis had been published in a number of research publications and one presentation listed as follows:

1. **Contributions 1, 2, 3, and 5:** M. N. Demaidi, M. M. Gaber and N. Filer, (2018). ”OntoPeFeGe: Ontology-based Personalised Feedback Generator,” in *IEEE Access. IEEE press.* doi: 10.1109/ACCESS.2018.2846398

2. **Contribution 1:** M. N. Demaidi and M. M. Gaber, (in press), *Terminological Ontology Evaluator in eLearning*. LAMBERT Academic Publishing, 2018.

3. **Contribution 1:** M. N. Demaidi, M. M. Gaber, (2018). ”TONE: A Method for Terminological Ontology Evaluation”, (under review), in *MethodsX.*, Elsevier.

4. **Contribution 4:** M. N. Demaidi, M. M. Gaber, N. Filer, (2017). ”Evaluating the quality of the ontology-based auto-generated questions,” in *Smart Learning Environments*, 4(1), 7.

5. **Introduces the thesis general idea:** M. N. Demaidi, N. Filer, (2012). ”Wireless ray tracing educational land,” in *Proceedings of INTED2012 Conference*. International Association for Technology, Education and Development (IATED), Conference Proceedings, pp. 0402−0411.

6. **Introduces the thesis general idea:** M. N. Demaidi, N. Filer, (2012). ”Teaching about Radio using an On-line Virtual Environment,” in *Proceedings of Multi-Service Networks workshop (Presentation only)*[1].

1.5 Thesis Structure

- Chapter 2 reviews the existing literature giving summary studies of personalised formative feedback in learning environments and domain ontologies and how they are used to auto-generate formative feedback and questions. In addition, Chapter 2 reviews several approaches used to evaluate the coverage and semantic richness of domain ontologies.

[1] http://www.informatics.sussex.ac.uk/research/projects/ngn/slides/msn12talks/demaidi-radio.pdf

- Chapter 3 introduces Contribution 1, which is a Terminological ONtology Evaluator (TONE) for evaluating the candidate domain ontologies used to auto-generate assessment questions and different types of formative feedback.

- Chapter 4 presents Contributions 2 and 3 by describing the design and implementation of the Ontology-based Personalised Feedback Generator (OntoPeFeGe) system. Two preliminary studies had been carried out and reported in Appendix A to guide the design and implementation of OntoPeFeGe. The first study aimed to investigate the importance of formative feedback to students in VLEs. The second study examined the types of feedback some teachers provide to students in learning environments and if the types of feedback provided to students differ across educational courses.

- Chapter 5 presents Contribution 4 by evaluating the quality of assessment tests and questions using the Classical Test Theory and the Item Response Theory statistical methods. Moreover, the chapter studied the effect of the ontology-based generation strategies, the level of the question in Bloom's taxonomy, and the type of question on the questions' quality measurement results.

- Chapter 6 presents Contribution 5 which evaluates the Ontology-based Personalised Feedback Generator (OntoPeFeGe) by studying the effect of the personalised feedback on students' performance, and investigating students and teachers' satisfaction with the different types of auto-generated feedback.

- Chapter 7 summarises the contributions and outline future avenues for research.

Chapter 2

Personalised Formative Feedback and Ontologies in Education, A Review

As aforementioned in Chapter 1, the primary aim of this thesis is to propose a novel, generic framework that can use a pre-existing domain ontology to auto-generate assessment questions and different types of feedback, provides students with personalised feedback, and evaluates the candidate ontologies using the educational corpus. Therefore, this chapter reviews the existing literature underlying this thesis with the following four aims in mind:

1. Discussing the types of formative feedback students receive in learning environments, and how the different types of feedback interact with the student and the task characteristics and affect students' learning. Moreover, this chapter discusses the limitations in the existing personalised feedback frameworks and how this thesis will address them.

2. Presenting several ontology-based feedback generators, which are domain dependent (e.g., an expert knowledge base which captures the experts' solutions to the problem scenario is used in addition to the domain ontology during the generation process), auto-generate specific types of formative feedback, and do not take into account the student or the task characteristics.

3. Describing several ontology-based question generators, which use different generation strategies to auto-generate assessment questions. The chapter will show that the evaluation of the auto-generated questions was confined to measuring the qualitative satisfaction of the domain experts and students. It will also demonstrate that none of the ontology-based question generators tested the questions

on students to analyse the questions' difficulty and the questions' ability to discriminate between high ability and low ability students.

4. Studying several ontology evaluation approaches and demonstrating the need for a new approach to evaluate the candidate domain ontologies used to auto-generate the assessment questions and the different types of feedback.

2.1 Personalised Formative Feedback in Learning Environments

Formative feedback is the feedback students receive after answering an assessment task and it is essential in formative assessment systems (Black and Wiliam, 2009; Hattie and Timperley, 2007; Black and Wiliam, 1998; Sadler, 1989; Vasilyeva et al., 2008b). The assessment tasks focused on in this thesis are the true and false, multiple choice and short answer questions. Several studies reported that students found the formative feedback they received immediately after answering assessment questions in Blackboard (Blackboard, 1997) and Moodle (Moodle, 2011) Virtual Learning Environments (VLEs) useful and important to understand the educational material (Dermo and Carpenter, 2011; Bedford and Price, 2007).

Formative feedback provided to students can be classified into several types which vary in their pedagogical content. The following are a representative set of the types of feedback (Mason and Bruning, 2001; Narciss et al., 2014; Narciss, 2013; Bangert-Drowns et al., 1991; Mory, 2004):

- Knowledge Of Results (KOR) feedback provides students with information about the correctness of their response (e.g., correct/incorrect).

- Knowledge of the Correct Response (KCR) feedback provides students with the correct answer.

- Answer Until Correct (AUC) feedback provides students with KOR feedback until they answer the question correctly.

- Elaborated feedback provides additional information besides KOR and/or the KCR feedback:

  - Bugs-Related (BR) feedback provides students with information about their specific error or misconceptions (e.g., what is wrong and why).

- Topic Contingent (TC) feedback provides students with the correct answer and why the correct answer is right.
- Response Contingent (RC) feedback provides students with information which describes why the incorrect answer is wrong and why the correct answer is right.
- Hint feedback provides students with information on what to do next as a guide towards the right direction and it avoids presenting the correct answer explicitly.

Blackboard (Blackboard, 1997) and Moodle (Moodle, 2011) VLEs provide teachers with a web-based platform to create assessment questions and associate each question with hard-coded feedback (Vasilyeva et al., 2008b). These environments adopt the 'one size fits all' approach and provide students with the same type of feedback when their answers to a specific question are identical (Vasilyeva et al., 2008b; Downes, 2007). This may negatively affect students' learning, as students in learning environments differ from each other in many ways including their background knowledge of a particular subject and current level of knowledge (Narciss et al., 2014; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010).

This thesis focuses on providing students with personalised formative feedback, which is defined as tailoring the types of feedback provided to students based on the student's characteristics and/or the task's characteristics (Narciss and Huth, 2004; Narciss et al., 2014; Narciss, 2013; Mory, 2004; Vasilyeva et al., 2008a). Student's background knowledge and current level of knowledge have been considered as core factors for adapting the types of feedback to individual students (Mason and Bruning, 2001; Mitrovic and Martin, 2000; Narciss et al., 2014; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010). In addition to the student's characteristics, researchers considered the task's characteristics as important factors in the process of personalising feedback in learning environments (Narciss et al., 2014).

The following section presents several personalised feedback frameworks which tailor the different types of feedback based on the student's and/or the task's characteristics.

Table 2.1: Summary of personalised feedback frameworks

	Adaptive framework	Context (Theoretical /Practical)	Adaption			Types of feedback content (Auto-generated /Hard-coded)	Types of feedback supported					
			Student characteristics		Task characteristics		KOR	KCR	BR	TC	RC	Hint
			Background knowledge	Current level of knowledge								
1	Mason and Bruning (Mason and Bruning, 2001)	Theoretical	✓	✓	✓	N/A			✓	✓	✓	
2	Adaptive feedback framework (AFF) (Gouli et al., 2006)	Theoretical		✓		N/A	✓	✓	✓	✓	✓	
3	Gimeno *et al.* (Ana and Macario, 2009)	Theoretical		✓		Hard-coded	✓					✓
4	Pardo *et al.* (Pardo et al., 2017)	Practical		✓		Hard-coded	✓					✓
5	SQL-Tutor (Mitrovic and Martin, 2000)	Practical		✓		Auto-generated	✓		✓			✓
6	Narciss and Huth (Narciss and Huth, 2004; Narciss and Huth, 2006)	Practical		✓		Hard-coded	✓	✓	✓			
7	Narciss *et al.* (Narciss et al., 2014)	Practical		✓		Auto-generated	✓	✓		✓		✓
8	Animal Watch (Arroyo et al., 2000; Arroyo et al., 2001)	Practical		✓		Hard-coded						✓
9	Wayang Outpost (Arroyo et al., 2011; Woolf et al., 2010)	Practical		✓		Hard-coded						✓

2.2 Personalised Feedback Frameworks

Personalised feedback frameworks define different mechanisms to tailor the types of feedback to student's characteristics and/or the task's characteristics (Narciss et al., 2014). This section presents a literature survey study that was conducted to identify the limitations in the existing personalised feedback frameworks. Table 2.1 provides a summary of the personalised feedback frameworks reviewed. The frameworks are classified into two groups according to their context:

1. Theoretical frameworks, which have not been evaluated on students.
2. Practical frameworks, which have been evaluated on students. These frameworks adapt the feedback either by increasing the amount of information provided in a particular type of feedback or by providing students with different types of feedback gradually.

Table 2.1 shows that the reviewed personalised feedback frameworks adapt the feedback based on student's current level of knowledge except Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001), which considered the student's background knowledge, the student's current level of knowledge, and the task's characteristics. Mason and Bruning (Mason and Bruning, 2001) presented a theoretical framework which aimed to help developers, instructors and designers build personalised feedback learning environments. Their framework is based on research which examined the relationships between the types of feedback, student's background knowledge, student's current level of knowledge, the task difficulty, and the timing of feedback (e.g., immediate or delayed). Mason and Bruning defined six rules in (Mason and Bruning, 2001) to provide students with personalised BR, TC, and RC feedback. For example, one of Mason and Bruning's rules is that if a student with high background knowledge fails to answer (student's current level of knowledge is low) a difficult question, the student receives TC feedback.

Gouli *et al.* (Gouli et al., 2006) also presented a theoretical Adaptive Feedback Framework (AFF), which was designed to provide the students with more types of feedback (i.e., KOR, KCR, BR, TC, and RC) compared to Mason and Bruning's framework (Mason and Bruning, 2001). The AFF provides the students with different types of feedback gradually based on their current level of knowledge. The framework starts by providing the student with the KOR feedback and if the student fails to answer the question, the framework provides him/her with more detailed feedback such as the RC feedback. Gouli *et al.* focused on adapting the types of feedback based on student's current level of knowledge and did not consider student's background knowledge and the task's characteristics.

Gimeno *et al.* (Ana and Macario, 2009) developed a system to teach students foreign languages and provide them with personalised KOR and hint feedback based on their current level of knowledge. Their system provided teachers with a template to create assessment questions and associate each expected answer with hint feedback. Gimeno *et al.* did not carry out any empirical evaluation to assess the effect of hint feedback with different information on students' performance. Similarly, Pardo *et al.* developed a system which provides students enrolled in the computer systems course with personalised KOR and hint feedback based on their current level of knowledge. The feedback messages were hard-coded by the teacher for each assessment question. Unlike Gimeno *et al.* (Ana and Macario, 2009), Pardo *et al.* carried out an empirical evaluation to assess the effect of personalised feedback on students' performance. Their results revealed that the personalised feedback significantly improved students' performance (Pardo et al., 2017).

Mitrovic and Martin (Mitrovic and Martin, 2000) developed SQL-Tutor, which aims to teach the SQL database language to university students. SQL-Tutor auto-generates KOR, BR, and hint feedback from a constraint-based model, which represents the knowledge in the SQL database domain as a set of constraints on the correct solutions (Mitrovic and Ohlsson, 1999). SQL-Tutor starts by providing the student with KOR feedback and if the student fails to answer the question he or she will receive BR feedback and then hint feedback. SQL-Tutor evaluation results revealed that students who received personalised feedback had higher performance compared to students who received no feedback (Mitrovic and Ohlsson, 1999).

Narciss and Huth (Narciss and Huth, 2004; Narciss and Huth, 2006) provided students with KOR, KCR, and BR personalised types of feedback based on their current level of knowledge. Their framework starts by providing students with KOR feedback and if the student fails to answer the question he or she is provided with BR feedback and then both KCR and BR feedback. Narciss and Huth evaluation results revealed that students' performance improved by 35.3% after receiving personalised feedback compared to 15% improvement when students received non-personalised KOR and KCR feedback (Narciss and Huth, 2006).

Narciss *et al.* (Narciss et al., 2014) in a different study provided students with personalised KOR, KCR, TC, and hint feedback based on the student's current level of knowledge. If a student fails to answer a question he or she is provided with KOR and hint feedback, and if the student fails again to answer the same question Narciss *et al.* framework provides the student with TC feedback and then KCR feedback. Narciss *et al.* evaluated the personalised feedback framework in ActiveMath, which is a web-based learning environment for mathematics. ActiveMath captures the mathematics exercises, students' interactions with the system, and the feedback in a finite state machine. States in the finite state machine represents an assessment task, an interaction in which the student provides the solution to the task and submits it into the system, or feedback. ActiveMath uses the finite state machine to auto-generate different types of feedback based on students' interaction (Goguadze, 2009). Narciss *et al.* (Narciss et al., 2014) evaluated the personalised feedback framework on students and the results revealed that students who received personalised feedback had more improvement in their performance compared to students who received non-personalised KOR and KCR feedback (Narciss et al., 2014). Narciss *et al.* (Narciss et al., 2014) also examined the relationship between students gender and students' performance after receiving personalised feedback, and their results revealed that female students had higher performance compared to male students.

Arroyo *et al.* (Arroyo et al., 2000; Arroyo et al., 2001) developed a system called Animal Watch which provides students with different types of hint feedback in the mathematical domain. The types of hint feedback content increased progressively based on the correctness of student's answers. Animal Watch starts by providing the student with little information, but if the student keeps entering wrong answers, Animal Watch increases the amount of information which will ultimately guide the student towards the correct answer. Arroyo *et al.* studied the relationship between students' gender and performance after receiving different types of hint feedback. The results again revealed that female students outperformed male students after receiving personalised feedback. Arroyo *et al.* carried out another study in (Arroyo et al., 2011; Woolf et al., 2010) using Wayang Outpost, an intelligent tutor for teaching students geometry in mathematics. They studied the relationship between students' background knowledge, gender, and performance after receiving different types of hint feedback. Student's background knowledge was determined by an assessment test. Students with high background knowledge had performance above 50% while students with low background knowledge had performance below 50%. The evaluation results revealed that students' performance increased after receiving personalised feedback. Moreover, the results revealed the students with low background knowledge had more improvement in their performance compared to students with high background knowledge after receiving personalised feedback. However, no statistically significant difference was found between male and female students.

In summary, surveying the personalised feedback frameworks revealed the following:

1. The personalised feedback improved students' performance (Narciss et al., 2014; Arroyo et al., 2011; Woolf et al., 2010; Mitrovic and Ohlsson, 1999). However, the personalised feedback frameworks, which were evaluated on students adapted the types of feedback based on student's characteristics and only Mason and Bruning's theoretical framework (the framework was not evaluated on students) considered the task's characteristics in the adaptation process. This issue has also been addressed by Narciss et al. in (Narciss et al., 2014) who considered the task characteristics as important factors in the process of personalising feedback in learning environments.

2. The evaluation results obtained from different frameworks are not consistent. Narciss *et al.* (Narciss et al., 2014) showed that female students had higher performance than male students after receiving personalised feedback. Arroyo *et al.*, had similar results in (Arroyo et al., 2000; Arroyo et al., 2001), however in another study they carried out in (Arroyo et al., 2011; Woolf et al., 2010) no statistically significant difference in performance was found between male and female students, and both genders had an increase in their performance after receiving personalised feedback. This suggests that there is still no clear definition regarding the relationship between student's characteristics, task's characteristics and personalised feedback (Mason and Bruning, 2001; Narciss et al., 2014; Arroyo et al., 2011; Shute, 2008).

3. The types of feedback in the frameworks are domain dependent as they are only applicable for the educational domain the personalised feedback framework is designed for (Cocea, 2011). The feedback content is either hard-coded in the system (Ana and Macario, 2009; Narciss and Huth, 2004; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010) or auto-generated from a restricted set of solutions defined by the teacher or a domain expert such as in SQL-Tutor (Mitrovic and Martin, 2000) and Narciss *et al.* (Narciss et al., 2014) personalised feedback frameworks.

4. Creating different types of feedback in personalised feedback frameworks is a time consuming process (Mitrovi, 1998; Passier and Jeuring, 2004), as teachers have to either hard-code the different types of feedback in the system (Ana and Macario, 2009; Narciss and Huth, 2004; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010) or create a restricted set of solutions from which the feedback is auto-generated (Mitrovic and Martin, 2000; Narciss et al., 2014; Kazi et al., 2013).

This thesis addresses the issues mentioned above by developing a novel personalised feedback framework in Chapter 4 which achieves the following:

1. Auto-generates domain independent types of feedback so that teachers do not need to intervene with the system and create feedback for each question's option (e.g., a multiple choice question has more than one option).

2. Provides students with personalised feedback by adopting Mason and Bruning's approach (see Section 4.2). This is used to study the relationship between the student's characteristics, the task's characteristics and the personalised feedback in Chapter 6.

Auto-generating different types of formative feedback in learning environments require a knowledge representation of the educational domain (e.g., medicine or computer networks), which captures the concepts in an educational course. As mentioned above, the SQL-Tutor (Mitrovic and Martin, 2000) and Narciss *et al.* (Narciss et al., 2014) personalised feedback frameworks used the constraint-based model and the finite state machine respectively as a knowledge representation of the educational domain. They encoded all possible solutions into the system to auto-generate feedback to students. Their approaches are time consuming and lead to the knowledge acquisition bottleneck (Kazi et al., 2013). To overcome these drawbacks, feedback could be generated using a pre-existing knowledge base which captures the educational domain and is available for reuse and sharing. Ontologies represent a feasible alternative to constraint-based models and finite state machines in reducing the burden of knowledge acquisition, as they present a high potential for reuse (Kazi et al., 2013). In addition, high quality ontologies already exist in several domains (e.g., medicine) (Bodenreider, 2004). More details about ontologies and how they are used in learning environments are provided in Section 2.3.

2.3 Ontology-based Formative Feedback Generators

This section presents an overview of ontologies and explains how ontologies are used in learning environments to capture the concepts of an educational course and auto-generate formative feedback and assessment questions.

2.3.1 An Overview of Ontologies

An ontology is defined as a formal and explicit specification of a shared conceptualisation (Uschold and Gruninger, 1996; Studer et al., 1998; Borst, 1997). It is a knowledge representation structure which models a specific domain of interest by providing a formal machine readable representation of entities in the domain. Entities include classes, individuals, and properties. Classes represent sets of individuals, individuals represent actual objects in the domain, and properties represent relationships in the domain between individuals. In this thesis, the word individual is used to represent an instance in the domain ontology (Group, 2004).

The World Wide Web Consortium (W3C), an international organisation supporting the development of standards for the World Wide Web, proposes several formalisms to encode the ontology such as Resource Description Framework (RDF) (W3C, 2014), Resource Description Framework Schema (RDFS) (W3C, 2014) and Web Ontology Language (OWL) (Group, 2004). The formalisms vary in their expressive power (W3C, 2004a). RDF is considered the basis for encoding ontologies and it is usually interpreted as a set of statements (triples) about ontology resources in the form of:

[subject property object]

The subject identifies the resource described in the statement, the property describes the property of the resource, and the object defines the value of the property which could be either a resource or a literal which is a representation of data values such as strings or integers. For instance, RDF could be used to define that the Knowledge Of Results (KOR) feedback is a type of feedback using the following statement:

$$\overbrace{KOR}^{Subject}\ is\ a\ \overbrace{type\ of}^{Property}\ \overbrace{Feedback}^{Object}$$

RDF resources and properties are named using Uniform Resource Identifier (URI) references. URIs are short strings that identify resources and a URI reference is a URI with an optional fragment identifier at the end (Mealling and Denenberg, 2002). For example, the URI reference *‘http://example.com#KOR’* consists of the URI *‘http://example.com’* and the fragment identifier KOR separated by a ‘#’ character.

RDF represents a set of statements as a directed labelled graph where subjects and objects are nodes connected using a property. Any additional statements could be made about KOR or Feedback, resulting in a growing graph of interconnected data. For instance, the following statements could be added about the KOR Key, which is the KOR feedback associated with the question's correct answer (Key):

1. 'KOR Key is a KOR feedback'. The subject is *KOR Key*, the object is *KOR*, and the *'is a'* is the property.

2. 'KOR Key has a pedagogical content which is your answer is right'. The subject is *KOR Key*, the object is *your answer is right*, and the property is *pedagogical content*.

The statements are represented as a graph in Fig 2.1. RDF is useful to define simple statements about resources using properties and values. However, it lacks the ability to define the resources and properties, and the relationships between them (Brickley and Guha, 2000). For example, a real-world resource such as 'Feedback' in the example above could be described in different ways, as RDF does not support a common language or vocabulary, where classes, subclasses, properties, and also relationships between these classes and properties are defined.

This issue had been addressed by RDFS, which extended the RDF vocabulary by allowing RDF resources to be presented in classes and properties hierarchies. RDFS does not provide application specific vocabulary for defining the class *Feedback* or the property *pedagogical content*. Instead, it provides the facilities needed to define these classes and properties and determine which classes and properties could be used together. RDFS uses rdfs:Class, rdfs:Resource, and rdf:Property to define classes (concepts), resources, and properties respectively. Moreover, RDFS defines the following properties which are used to create ontologies (W3C, 2014):

- rdf:type: Defines a property which describes the relationship between the class and the individual.

- rdfs:subClassOf: A property that models the hierarchy between classes.

- rdfs:subPropertyOf: A property that models the hierarchy between properties.

- rdfs:domain: A property which constraints all individuals of a property to describe individuals of a particular class.

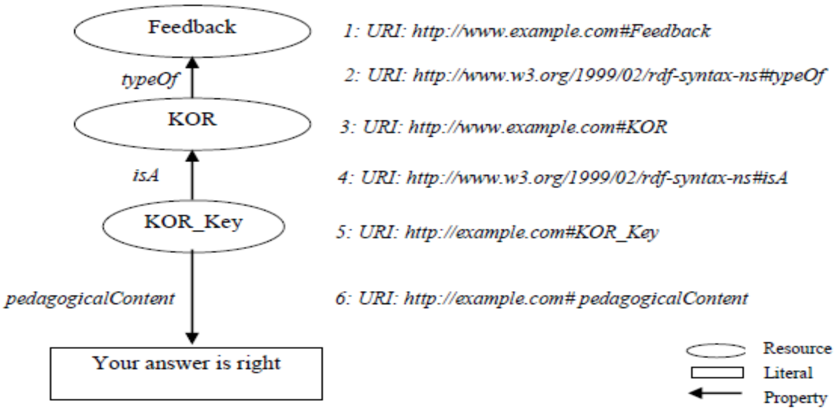

Feedback
typeOf
KOR
isA
KOR_Key
pedagogicalContent
Your answer is right
1: URI: http://www.example.com#Feedback
2: URI: http://www.w3.org/1999/02/rdf-syntax-ns#typeOf
3: URI: http://www.example.com#KOR
4: URI: http://www.w3.org/1999/02/rdf-syntax-ns#isA
5: URI: http://example.com#KOR_Key
6: URI: http://example.com# pedagogicalContent
Resource
Literal
Property

Figure 2.1: Three statements in an RDF graph

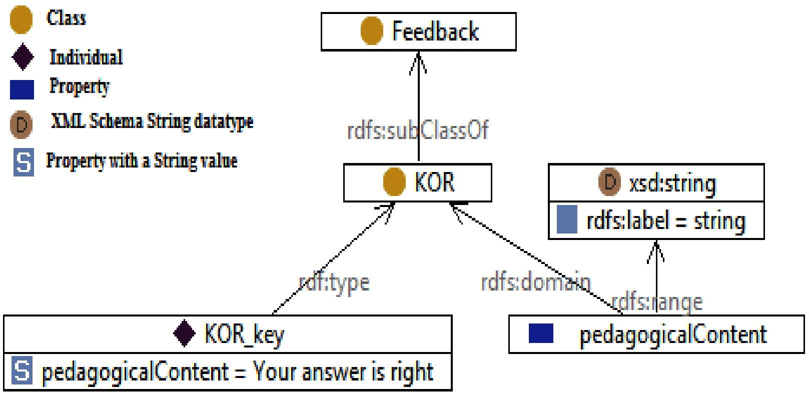


Figure 2.2: Feedback ontology in RDFS

- rdfs:range: A property which constraints all individuals of a property to have values of a particular class.
- rdfs:comment: A property which provides human readable descriptions.
- rdfs:label: A property used to provide a name for the class or property.

Fig 2.2 illustrates the feedback ontology presented in RDFS. The Feedback and KOR are defined as classes (rdfs:Class), KOR is defined as a subClass of Feedback (rdfs:subClass), the pedagogical content is defined as a property which has the domain KOR and a string range value, and the KOR_Key is defined as an individual of KOR class which has the pedagogical content property value your answer is right.

RDFS is a simple encoding formalism which supports classes and properties hierarchies, as well as domain and range restrictions on properties. OWL adds more vocabulary to RDFS, which allows restricting classes and properties, and thus allowing the design of more expressive domain ontologies (Pahl and Holohan, 2009). Classes in OWL are created using owl:Class and the properties are slightly different from the properties in RDFS. While RDFS use rdf:Property to connect two resources or to connect a resource with a literal, OWL distinguishes between these two connections using two different classes:

- owl:ObjectProperty: Connects two resources together.
- owl:DatatypeProperty: It is used to connect a resource to an RDFS:Literal or to an XML schema built-in datatype value (W3C, 2012). For example, the pedagogical content is a datatype property which connects the resource KOR to the string datatype value.

In addition to the object and datatype properties, OWL defines the annotation property which includes rdfs:comment and rdfs:label.

Properties in OWL have much richer features compared to RDF and RDFS. For example, a property could be defined as an inverse of another property (i.e., if a resource R1 is connected to resource R2 using property P, then R2 is connected to R1 by the inverse property of P). In addition to the properties definition features, OWL allows object properties and datatype properties to have value restrictions and cardinality restrictions by defining the following classes:

- owl:allValuesFrom: The values of the restricted property must all come from the specified class or data range.
- owl:someValuesFrom: At least one of the values of the restricted property has to come from the specified class or data range.
- owl:hasValue: At least one of the property values is equal to the specified value.
- owl:Cardinality: To constrain the number of property values. For example, a person has only one biological mother and one biological father.
- owl:minCardinality and owl:maxCardinality: They can be used to specify a property value range.

A restriction in OWL is a class defined by describing the individuals it contains (Allemang and Hendler, 2011). Fig 2.3 illustrates the value and cardinality restrictions applied to the *pedagogical content* property in the feedback ontology. The figure shows

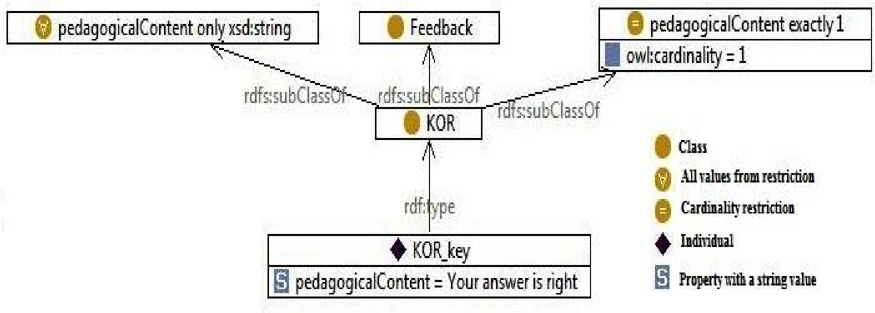

pedagogicalContent only xsd:string
Feedback
pedagogicalContent exactly 1
owl:cardinality = 1
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
KOR
rdf:type
KOR_key
pedagogicalContent = Your answer is right
Class
All values from restriction
Cardinality restriction
Individual
Property with a string value

Figure 2.3: Value and Cardinality restrictions in OWL

that all values of the pedagogical content property must be a *string* (All values from restriction). In addition, the figure shows that the *pedagogical content* property must have one value (Cardinality restriction).

OWL provides richer vocabulary compared to RDFS and RDF. For this reason, the ontologies used in this thesis to auto-generate the assessment questions and the different types of formative feedback are created in OWL.

2.3.2 Ontologies in Virtual Learning Environments

Ontologies have been used in Virtual Learning Environments (VLEs) to capture the concepts in an educational course (Al-Yahya et al., 2015). Sakathi (Murugan.R et al., 2013) developed an ontology which captures concepts in the computer networks domain such as the network topology, the communication's medium, and the Open Systems Interconnection (OSI) model (Kurose and Ross, 2013). Part of Sakathi's Computer Networks ontology is shown graphically in Fig 2.4. Lee (Lee et al., 2005), Kouneli (Kouneli et al., 2012), and Ganapathi (Ganapathi et al., 2011) developed ontologies which capture the educational concepts in the Java language introductory courses (Arnold et al., 1996). The ontologies aimed to teach students the fundamental concepts of programming in Java. Part of Lee's Java ontology which captures the Java expressions and statements is shown graphically in Fig 2.5 (Lee et al., 2005).

Domain ontologies had also been developed in the medical domain such as the Unified Medical Language System (UMLS) ontology which contains two million medical concepts (Bodenreider, 2004), and the OpenGALEN (Rector et al.,) ontology which captures concepts about drugs, anatomy, oncology, and many other concepts in the medical domain. On the other hand, some ontologies were not developed to

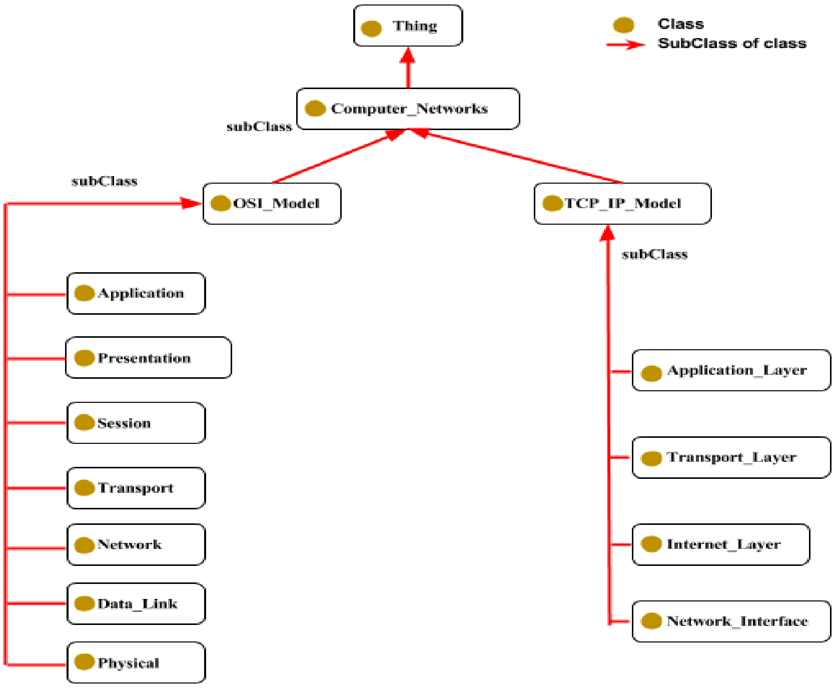


Figure 2.4: Part of a Computer Networks ontology that illustrates the class hierarchy adopted from (Murugan.R et al., 2013)

capture particular domains. Instead, they aimed to have the world's largest and complete knowledge base that covers different domains. Among these ontologies is the OpenCyc ontology (OpenCyc, 2011). The word OpenCyc consists of two words, Cyc, which is taken from the word 'encyclopaedia' and Open which is used to indicate that the ontology is open source. OpenCyc covers several domains such as mathematics, physics, medicine, computer networks and many others, and it consists of hundreds of thousands of concepts and properties (OpenCyc, 2011).

The experiments in Chapters 3, 5, and 6 of this thesis were carried out on a sample of undergraduate university students registered in the Data Networking and Computer Networks courses at the University of Manchester (of Manchester, 2013d; of Manchester, 2013a). Therefore, the examples in Chapter 4 and the studies in Chapters 3, 5, and 6 focused on ontologies developed to capture concepts from the computer networks field. Due to the lack of ontologies available in this field, the pre-existing Sakathi's Computer Networks ontology (Murugan.R et al., 2013) and the OpenCyc ontology (OpenCyc, 2011) were used, and no changes have been applied to the ontologies.

Section 2.3.3 explains how the domain ontologies have been used to auto-generate

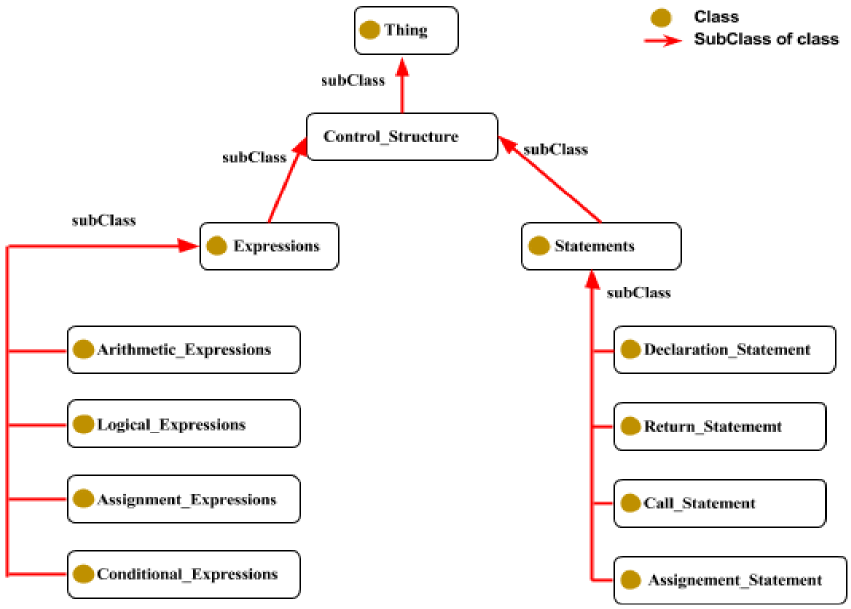


Figure 2.5: Part of a Java ontology adopted from (Lee et al., 2005)

different types of feedback and assessment questions.

2.3.3 A Review of Existing Ontology-based Generators

Ontologies have been used in learning environments to generate formative feedback and questions in different educational domains. This section reviews and highlights the limitations in existing ontology-based feedback generators.

Ontology-based Formative Feedback Generators

Several generators were found in the literature and reviewed in this thesis (See Table 2.2). The generators differ in the types of feedback generated, the capability of the generator to auto-generate feedback across different educational domains (domain dependent or domain independent), and the underlying domain ontology used in the generation process.

Kazi *et al.* developed a system in the medical domain, which auto-generates hint feedback using the Unified Medical Language System (UMLS)[1] ontology and an expert knowledge base that captures the experts' solutions to a problem scenario. The expert solutions represent the cause-effect relationships between the medical concepts

[1] UMLS and OpenGALEN are medical ontologies. They were briefly introduced on page 39.

Table 2.2: Ontology-based formative feedback generators

Formative feedback generator	**Types of feedback generated**	**Domain dependency**	**Underlying domain ontology**
Kazi *et al.* generator	Hint	Dependent (medicine)	UMLS
Sãnchez-Vera *et al.* generator	KOR KCR	Dependent (Design and Production of Educational Materials)	Domain ontology created by teachers
Duboc *et al.* generator	KCR BR TC	Dependent (medicine)	OpenGALEN and Freebase

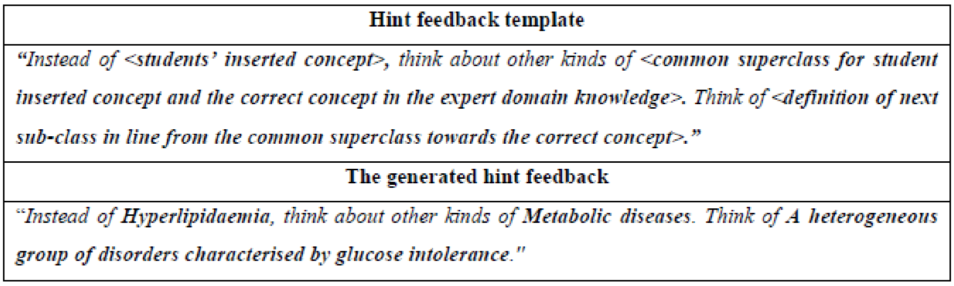
(Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013). Kazi *et al.* system provides a group of six to eight students with a medical problem scenario. Students solve the problem scenario in the form of a causal graph, where the graph nodes represent the medical concepts, and the directed edges are the casual links which represent the cause-effect relationships between the concepts. Each medical concept in the graph represents one of the following (Kazi et al., 2010):

1. The enabling conditions, which are factors that trigger the onset of a medical condition (e.g., ageing and smoking).

2. Faults, which are the bodily malfunctions that result in various signs and symptoms (e.g., pneumonia, diabetes).

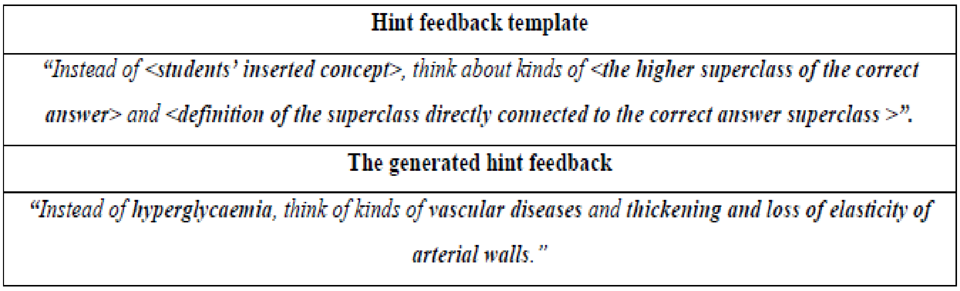
3. Consequences, which are the signs and symptoms that occur as a result of the diseases or disorders (e.g., fatigue and numbness).

Kazi *et al.* system auto-generates hint feedback after evaluating the casual links which the students create between the medical concepts. If the casual link exists in the expert knowledge base, the system accepts and draws the casual link. However, if the casual link does not exist in the expert knowledge base, the casual link is not drawn and the hint feedback is generated to guide students towards the correct answer. The hint feedback is auto-generated in Kazi *et al.* system by searching the UMLS domain ontology for the lowest common superclass between the students' answer and the correct answer in the expert knowledge base. If a common superclass is found, the system uses the hint feedback template shown in Fig 2.6. However, if no common superclass is found, the system uses the hint feedback template shown in Fig 2.7. For example, if the students draw the following casual link: $Hyperlipidemia \Rightarrow$ *Diabetic neuropathy* which does not exist in the expert knowledge base, the system will recognise that the students have mistaken *Hyperglycemia* by *Hyperlipidemia*. After that, the system will



Hint feedback template
“Instead of <students’ inserted concept>, think about other kinds of <common superclass for student inserted concept and the correct concept in the expert domain knowledge>. Think of <definition of next sub-class in line from the common superclass towards the correct concept>.”
The generated hint feedback
“Instead of Hyperlipidaemia, think about other kinds of Metabolic diseases. Think of A heterogeneous group of disorders characterised by glucose intolerance.”

Figure 2.6: Feedback template and the feedback generated (common superclass)



Hint feedback template
“Instead of <students’ inserted concept>, think about kinds of <the higher superclass of the correct answer> and <definition of the superclass directly connected to the correct answer superclass >”.
The generated hint feedback
“Instead of hyperglycaemia, think of kinds of vascular diseases and thickening and loss of elasticity of arterial walls.”

Figure 2.7: Feedback template and the feedback generated (no common superclass)

check if there is a common superclass between the hint feedback using the template shown in Fig 2.6 which is fulfilled by:

1. Replacing ‘students’ inserted concept’ with *Hyperlipidaemia*.

2. Searching for the lowest common superclass in the UMLS domain ontology between *Hyperglycaemia* and *Hyperlipidemia*, which is the *Metabolic Disease* class.

3. Retrieving the definition of the next subclass in line from the common superclass towards the correct answer, which is the *Glucose Metabolism Disorders* class and it is defined as ‘A heterogeneous group of disorders characterized by glucose intolerance’.

If no common superclass is found in the UMLS domain ontology between students’ answer and the correct answer in the expert knowledge base, the system will use the hint feedback template shown in Fig 2.7. The template is fulfilled using the superclass of the correct answer. For example, if the students draw the following causal link: *Hyperlipidaemia* ⇒ *Hyperglycaemia* which does not exist in the expert knowledge base, the system will recognise that the students have mistaken *Hyperglycemia*

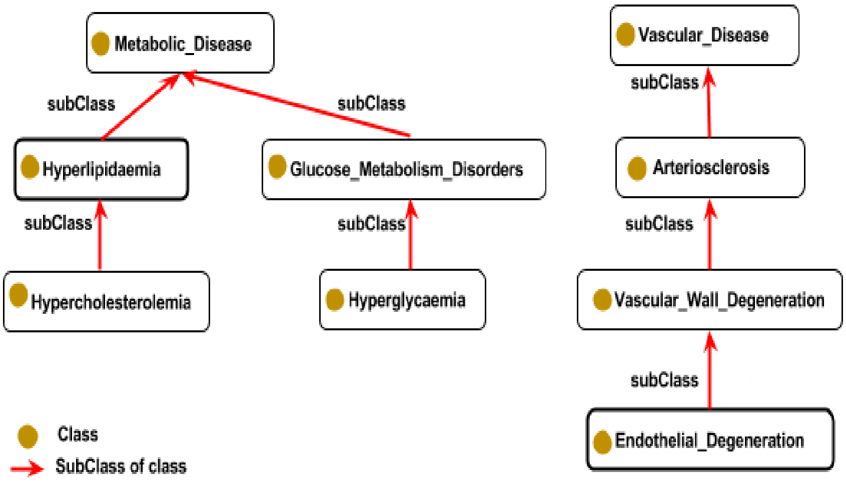


Figure 2.8: UMLS class hierarchy example (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013)

by *Endothelial Degeneration*. After that, the system will check if there is a common superclass in the UMLS domain ontology between the *Hyperglycaemia* and the *Endothelial Degeneration* concepts. In this example, no common superclass is found (see Fig 2.8). Therefore, the hint feedback is generated using the template shown in Fig 2.7. The higher superclass of the *Endothelial Degeneration* concept is *Vascular Disease* (see Fig 2.8), and the superclass which is directly connected to the *Vascular Disease* concept is *Arteriosclerosis* and it has the following definition: 'thickening and loss of elasticity of arterial walls'.

Kazi *et al.* evaluated the auto-generated hint feedback using five domain experts and ten students (Kazi et al., 2012; Kazi et al., 2010). Thirty samples of auto-generated hint feedback were selected randomly from students' log files. The experts and students were asked to rate their acceptance of the samples of auto-generated hint feedback on a 5-point Likert scale: 1(strongly disagree) to 5 (strongly agree). For each sample, the experts and students were provided with the causal link created by the student, the causal link in the expert knowledge base, and the auto-generated hint feedback. The experts had an average rating score of 3.58 that the auto-generated hint feedback is close to what themselves would have provided to students, and the students had an average rating score of 3.78 that the hint feedback is what they would have expected to receive (Kazi et al., 2012; Kazi et al., 2010)[103]. This indicates that both experts and students accepted the auto-generated hint feedback (Kazi et al., 2012; Kazi et al., 2010).

Kazi *et al.* also evaluated the effect of the auto-generated feedback on the performance of thirty second-year medical students and thirty general physicians (Kazi et al., 2013). The students and physicians were asked to do a pre-test to assess their performance before receiving the ontology-based auto-generated hint feedback and a post-test to assess their performance after receiving the feedback. The results revealed that the students had a significant increase in their performance, while physician had no statistically significant difference between their pre-test and post-test performance (Kazi et al., 2013).

S$\tilde{a}$nchez-Vera *et al.* (Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011) also developed an ontology-based formative feedback generator. The generator is called the Ontology eLEarning (OeLE) system and it auto-generates the Knowledge Of Results (KOR) and the Knowledge of Correct Response (KCR) to students after they answer essay questions. The feedback is auto-generated using:(1) Domain ontologies which capture the concepts in an educational course; and (2) the annotations which are created by the teachers and used to identify the expected answer for each assessment question.

Teachers in OeLE create the course ontologies and the essay questions. The course ontologies in OeLE capture the course's educational concepts and they are used by teachers to annotate the expected answer associated with each question, and by the algorithm introduced in (S et al., 2012) to extract a set of annotations which capture the student's answer. OeLE auto-generates the feedback using a similarity algorithm, which compares the expected answer annotations with the annotations extracted from student's answer. If the similarity value between the annotations is above a threshold point which is set by the teachers using the assessment parameters, the students receive the KOR 'correct' feedback otherwise they receive the KOR 'incorrect' feedback. The similarity algorithm used in OeLE is not explained here, as it is out of the thesis scope. More details about the similarity algorithm can be found in (S et al., 2012; Castellanos-Nieves et al., 2011).

The OeLE system also auto-generates the KCR feedback by providing students with the expected answer annotations which are identified by the teachers and not included in the student's answer.

The formative feedback auto-generated in the OeLE system was evaluated by twenty-one students registered in the Design and Production of Educational Materials' course. The teachers in OeLE designed two assessment tests each consisting of five essay questions. There was a one week gap between the assessment tests. Students who performed the tests received both KOR and KCR feedback. After answering the assessment tests, students were provided with a questionnaire to assess their opinion regarding the feedback provided. 77.3% of the students agreed that the feedback allowed them to have a better understanding of the course content, 90% of the students agreed that the feedback is useful, and 86.4% of the students found that the feedback provided was not a waste of time (Frutos-Morales et al., 2010; S et al., 2012).

Sãnchez-Vera *et al.* also evaluated the effect of feedback on students' performance by comparing students' performance (scores) in both tests. The results revealed that students' performance increased significantly in the second test.

The last generator reviewed is Duboc *et al.* (Duboc et al., 2011; Duboc, 2013) formative feedback generator. They developed a virtual patient system which provides medical students with interactive computer simulations of real life clinical scenarios, and auto-generates feedback to help students solve the patients' cases. Duboc *et al.* created a virtual patient ontology which captures the following:

1. The virtual patient details using concepts from the openGALEN[2] domain ontology (e.g., body parts and gender).

2. The patient's symptoms (e.g., fever) using concepts from the Freebase[3], which is a large collaborative knowledge base (Bollacker et al., 2008).

3. Student's interactions with the system such as asking questions related to the patient or applying medical tests to solve the patient case.

4. The relationships between the medical conditions, symptoms and diagnosis which are inserted manually by the author of the virtual patient case.

The system provides the student with a virtual case of a patient. The student solve the case by applying medical examinations and asking the patient about the symptoms. The system analyses the student's interactions and provides him/her with the following types of formative feedback:

1. KCR feedback: A list of all diagnoses the student should have tested in the virtual patient case.

2. BR feedback: A list of interview questions and examinations the student should not have chosen, and the justification (e.g., irrelevant).

[2] OpenGALEN is a medical ontology which is briefly introduced on page 39.

[3] http://www.freebase.com/

3. TC feedback: A list of the interview questions and examinations students should have chosen, and the justification (e.g., a type of disease to consider and a related symptom to check).

The types of feedback are auto-generated by querying two types of data: the virtual patient data which is entered mainly by the virtual case author and the interaction data which is entered by each student trying to solve the case.

Duboc *et al.* evaluated the auto-generated feedback on 36 medical students. Students were asked to rank the importance and usefulness of each type of feedback on a 5-point Likert scale: 1(strongly disagree) to 5 (strongly agree). The results revealed that students had an average rating score above 4.0 for the KCR, BR, and TC types of feedback (Duboc, 2013). Moreover, Students agreed that the feedback in the virtual patient system improved their understanding of the patients' cases (average ranking score = 4.17) and helped them understand their errors and misconceptions (average ranking score = 4.03).

In summary, the ontology-based feedback generators discussed above have the following limitations:

1. Auto-generate limited types of feedback: Kazi *et al.* focused on auto-generating hint feedback, while S*ã*nchez-Vera *et al.* auto-generated KOR and KCR feedback. Duboc *et al.* system auto-generated more types of feedback. However, their system only generated KCR, BR, and TC feedback. See Table 2.2.

2. Domain dependent: Kazi *et al.* auto-generated the feedback using the expert knowledge base in addition to the domain ontology. S*ã*nchez-Vera *et al.* generated feedback based on both the course domain ontology and the annotations used to identify the expected answer for each assessment question. Duboc *et al.* generated feedback using the links between the conditions, symptoms and diagnosis classes which are created by the patients' case author.

3. The assessment questions are hard-coded by the teachers or the domain experts: The feedback generators hard-coded the assessment questions, therefore, the questions are only valid in the educational domain they are created in. In addition, the feedback generators did not specify the question's characteristics (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Duboc et al., 2011; Duboc, 2013; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011).

The limitations mentioned above hinder providing students with personalised feedback in a generic framework. Therefore this thesis presents an Ontology-based Personalised Feedback Generator (OntoPeFeGe) which achieves the following:

1. Auto-generates different types of formative feedback: The different types of feedback are determined after a preliminary study carried out and reported in Appendix A to investigate the types of feedback teachers provide to students in Virtual Learning Environments (VLEs).

2. Auto-generates domain independent feedback: The feedback generator is domain independent and no expert knowledge base or human intervention (teacher or domain expert) is needed.

3. Auto-generates assessment questions: The feedback generator aims to auto-generate assessment questions with different characteristics from a domain ontology.

4. Personalises the feedback provided to students: The feedback is personalised based on the student and the question characteristics.

Providing students with personalised feedback requires information about the assessment question characteristics (e.g., the question difficulty). Therefore, this next section reviews several domain independent question generators, which use the domain ontology to auto-generate questions with different characteristics.

Ontology-based Question Generators

Ontologies have been used by several ontology-based question generators to auto-generate several types of questions (true or false, multiple choice, and short answer questions) with different characteristics. The question generators used several ontology-based generation strategies, which exploit the ontology classes, properties and individuals. According to the literature, the ontology-based generation strategies could be categorised into the following three main strategies (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2014):

1. The class-based strategy, which uses the relationship between the class and individuals in the ontology.

2. The terminology-based strategy, which uses the relationship between the class and its subclasses in the ontology.

3. The property-based strategy, which uses the object, datatype, and annotation properties in the domain ontology (see Section 2.3.1, page 23 for more details on the object, datatype, and annotation properties).

Table 2.3 illustrates the ontology-based question generators, which auto-generate different types of questions using several ontology-based generation strategies.

Papasalouros (Papasalouros et al., 2008; Papasalouros et al., 2011) defined class-based, terminology-based, and property-based generation strategies, which traverse the domain ontology and auto-generate Multiple Choice Questions (MCQs) consisting of a correct answer (key) and incorrect answers (distractors). The three main strategies contain several sub-strategies, see Section 4.1.4 for more details.

Table 2.4 illustrates a MCQ generated using Papasalouros's terminology-based strategy. The question was generated from Sakathi's Computer network ontology which is shown in Fig 2.4. The question's key is a subclass of the concept OSI model and the question's distractors are sibling classes of the OSI model class. Table 2.4 also shows that the question had the Choose the correct sentence text, which is called the question's stem and it is used in all the questions generated using Papasalouros's question generator.

Cubric and Tosic (Cubric and Tosic, 2011) built a question generator, which used the ontology-based generation strategies defined by Papasalouros to auto-generate MCQs. However, they extended the property-based strategies to include more sub-strategies, which used the rdfs:label (a property used to provide a name for the class) and the rdfs:comment (a property which provides human readable descriptions) annotation properties (see Table 4.6 in Chapter 4, page 115 for an example). Moreover, instead of using Papasalouros's stem template which is not related to an educational theory, Cubric and Tosic defined a set of stem templates which aimed to assess students' cognition at different levels in Bloom's taxonomy (Bloom et al., 1956; Krathwohl, 2002). See Section 4.1.3 in Chapter 4, page 97 for more details about Cubric and Tosic's stem templates.

Bloom's taxonomy is widely used in the educational research (Krathwohl, 2002; Anderson and Sosniak, 1994). The taxonomy is used to categorise the assessment questions into the following six major levels which are arranged in a hierarchical order according to the complexity of the cognitive process involved (Bloom et al., 1956; Krathwohl, 2002; CAA, 2002):

Table 2.3: Ontology-based question generators

Question generators	Type of question			Ontology-based generation strategies				
	True and false	Multiple choice	Short answer	Class	Terminology	Property		
						Object	Datatype	Annotation
Papasalouros (Papasalouros et al., 2008; Papasalouros et al., 2011)		✓		✓	✓	✓	✓	
Cubric and Tosic (Cubric and Tosic, 2011)		✓		✓	✓	✓	✓	✓
Grubisic (Grubisic, 2012; Grubisic et al., 2013)	✓	✓	✓		✓	✓	✓	
Al-Yahya (Al-Yahya, 2011; Al-Yahya, 2014)	✓	✓	✓	✓		✓	✓	✓

Table 2.4: A MCQ generated using Papasalouros's terminology-based strategy (Price et al., 2010)

Stem:	Choose the correct sentence
Options:	(1) Transport layer is part of the OSI model (2) Network operating system is part of the OSI model (3) TCP/IP suite is part of the OSI model
Key:	(1) Transport layer is part of the OSI model

1. Knowledge: At this level, the students need only to recall certain concepts in the domain. For example, students need to list, define, and describe specific concepts in the domain without understanding how they are related to other concepts.

2. Comprehension: At this level, the students need to start thinking about the meaning of the concepts in terms of their relationship with other concepts in the domain.

3. Application: At this level, the students need to demonstrate their ability to use the concepts they have learned in real situations. For example, the students need to provide and show examples that prove their understanding of the domain concepts.

4. Analysis: At this level, the students need to understand the domain terminology structure. For example, the students need to have a good overview of the concepts in the domain by analysing how they are classified and related to each other.

5. Synthesis: At this level, the students should be able to relate concepts from different domains to create and develop new ideas.

6. Evaluation: At this level, the students need to make judgements, assess and compare ideas, and evaluate the data.

Each level in Bloom's taxonomy is subsumed by the higher levels. For example, a student functioning at the application level has mastered the educational concepts in the knowledge and comprehension levels (Bloom et al., 1956). Bloom's associated the levels hierarchical order with the question's difficulty (Bloom et al., 1956). For example, the knowledge level questions are easier than questions which assess other levels in Bloom's taxonomy, and the synthesis and evaluation questions are harder than the comprehension level questions (Bloom et al., 1956).

Cubric and Tosic (Cubric and Tosic, 2011) generated questions which assess the students at the knowledge, comprehension, application and analysis levels only. Grubisic (Grubisic, 2012; Grubisic et al., 2013) followed a similar approach to Cubric and Tosic by defining a set of stem templates which assess students' cognition at the knowledge, comprehension, application and analysis levels (see Section 4.1.3 in Chapter 4, page 97 for more details about Grubisic's stem templates). However, unlike the previous work, Grubisic generated different types of questions (true and false, multiple choice, and short answer) from domain ontologies. Moreover, she ignored the class-based strategies, and only used the terminology-based and the property-based strategies excluding the annotation properties (see Table 2.3), to traverse the domain ontologies and generate assessment questions.

Grubisic used ontology-based generation strategies, which are similar to the strategies defined by Papasalouros. However, fewer restrictions were applied for selecting the distractors in the generated question. For example, if a question is generated to assess concept *C*, Papasalouros defined that the distractor should be one of *Class C's* siblings while Grubisic allowed selecting any class randomly from the domain ontology as long as it has no relationship with *C*.

Al-Yahya (Al-Yahya, 2011; Al-Yahya, 2014) also built a question generator for auto-generating true and false, multiple choice, and short answer questions using class-based and property-based strategies. She defined the following two stem templates, one for each generation strategy:

1. The questions generated using the class-based strategy have the following stem template: *'Individual is a?'*. For example *'User Datagram Protocol is a?'*, generated by replacing *Individual* with the *User Datagram Protocol*, which is an individual in the OpenCyc ontology.

2. The questions generated using the property-based strategy have the following stem template: *'Individual Property?'*. For example, *'KOR Key pedagogical content is?'*, generated by replacing *Individual* with *KOR Key* and the *Property* with *pedagogical content*. See Fig 2.3.

Al-Yahya followed Grubisic's steps in allowing the distractors to be randomly selected from the domain ontology, however, the questions generated were aimed only to assess students' cognition at the knowledge level in Bloom's taxonomy (Al-Yahya, 2011; Al-Yahya, 2014).

The questions auto-generated using the question generators discussed above were evaluated either by domain experts to assess if the questions are syntactically correct and could be used in an assessment test, or students to assess if the auto-generated questions are comprehensible.

Cubric and Tosic (Cubric and Tosic, 2011) developed an online environment[4] where users could upload their domain ontologies, auto-generate the MCQs, and evaluate the questions created by them or other users in the environment. The users evaluate the auto-generated questions by determining the question quality (the question is easy to understand and the grammar is correct), and the question usability (the question could be used in an assessment test). Cubric and Tosic did not publish any evaluation results.

Grubisic (Grubisic, 2012; Grubisic et al., 2013) evaluated the questions auto-generated from the 'computer as system' domain ontology using two groups of students. The first group consisted of fourteen students who had good prior knowledge in the 'computer as system' domain. However, the students had no experience working with VLEs. The second group consisted of sixteen students who had learned about the 'computer as system' domain three years before the experimental study was carried out and had a good knowledge of different VLEs. 21% of the students in the first group found the questions comprehensible, 29% had a neutral opinion, and 50% found the questions incomprehensible (Grubisic et al., 2013). On the other hand, 38% of the students in the second group found the questions comprehensible, 38% had a neutral opinion, and 24% found the questions incomprehensible (Grubisic et al., 2013). Grubisic concluded that the students in the second group who were more mature (students who took the 'computer as system' course three years before the experiment was carried out) and had more experience working with different VLEs were more satisfied in terms of understanding the ontology-based generated questions [107].

Al-Yahya (Al-Yahya, 2011; Al-Yahya, 2014) auto-generated true and false, multiple choice, and short answer questions from several domain ontologies such as the travel ontology, which captures information about travel destinations and hotels (Library, 2013). She evaluated the auto-generated questions by assessing if the questions are syntactically correct and whether the questions were suitable to be used in an assessment test. Al-Yahya's evaluation results revealed that 90% of the questions generated were syntactically correct and could be used as assessment questions (Al-Yahya, 2011). Al-Yahya carried out further evaluation to assess if the auto-generated MCQs

[4] http://www.opensemcq.org/

were syntactically correct and could be used as assessment questions using three domain experts. The experts had experience in formulating MCQs and were asked to assess the MCQs generated from two domain ontologies (an ontology which captures the Arabic vocabulary (AlYahya et al., 2010) and a history ontology in Arabic which captures the historical concepts taught to students in the 8th grade (Al-Yahya, 2011; Al-Yahya, 2014; AlYahya et al., 2010)). The experts agreed that 82% of the MCQs generated from the Arabic vocabulary were syntactically correct and could be used as assessment questions, while 60% of the MCQs generated from the history ontology were syntactically correct and could be used as assessment questions (Al-Yahya, 2014). Al-Yahya stated that the difference in the evaluation results was due to the content of the domain ontologies, as the MCQs, which were classified as unacceptable in the history ontology, were dealing with common sense or general knowledge. This was not the case in the Arabic vocabulary ontology (Al-Yahya, 2014).

The review of the ontology-based question generators revealed that the evaluation of the auto-generated questions was confined to measuring the qualitative satisfaction of the domain experts and students. None of the ontology-based question generators mentioned above tested the questions on the students to analyse the quality of auto-generated questions by examining the question's difficulty, and the question's ability to discriminate between high ability and low ability students. The lack of quantitative analysis results in having no evidence on how the ontology-based generation strategies, the level of the question in Bloom's taxonomy (determined by the stem templates), and the type of question affect the questions' quality and produce more difficult or more discriminating questions. In addition, the domain experts and students' opinions are subjective. For example, the domain experts may agree that the questions could be used in an assessment test. However, there is no evidence that the question could discriminate between high ability and low ability students.

This thesis addresses the drawbacks mentioned above by integrating the ontology-based generation strategies used by the question generators and the stem templates into OntoPeFeGe (see Chapter 4) to achieve the following:

1. Auto-generate questions, which assess students at different levels in Bloom's taxonomy, and associate each question with different types of feedback. This helps provide students with personalised feedback in a generic framework where the types of feedback are adapted based on the student and the question characteristics.

2. Analyse the question's quality (see Section 5.3 in Chapter 5). More details on

the statistical methods used to assess the question's quality will be presented in Section 5.2.

3. Study the effect of the ontology-based generation strategy, the level of the question in Bloom's taxonomy, and the type of question (true and false, multiple choice, and short answer) on the quality of the question generated. This aims to provide guidance for developers and researchers working in the field of ontology-based question generators. See Section 5.3 in Chapter 5.

2.4 Feasibility of Ontology Evaluation Approaches

Assessing how suitable the ontology is for generating questions and feedback, as well as how it represents the subject domain of interest, is a necessary preface to using the ontology. However, this is not an easy task to carry out. As an example, suppose that a teacher needs to generate a question to assess students on the computer networks concept *Transport layer* (see Fig 2.4) or generate formative feedback related to that concept. The first step would be to search for a subset of computer networks ontologies that contain the concept. The teacher would find several candidate ontologies developed for different purposes with different sizes and levels of detail (e.g., the Computer Networks ontology (Murugan.R et al., 2013) and the OpenCyc ontology (OpenCyc, 2011)). Then, the next step would be to choose the most appropriate ontology from the set of candidate ontologies to auto-generate the questions and formative feedback. Achieving this process by hand is a difficult task and requires both deep subject knowledge and deep knowledge of domain ontologies.

As another example, suppose that the teacher needs to find an ontology containing not just a simple concept, but a set of concepts which are all taught in an educational course. Ideally, the teacher would expect to find a single ontology containing all the concepts. However, different ontologies capture different sets of concepts, and even if they do capture the same educational concepts that the teacher is searching for, the ontologies may capture them at a different level of detail giving too little or too much information.

This section presents several ontology evaluation approaches and assesses their feasibility for evaluating and selecting a domain ontology which could be used to auto-generate questions and formative feedback. According to Brank (Brank et al., 2005), the ontology evaluation approaches fall into one of the following categories:

1. Gold-standard evaluation: Compare the candidate ontology to a gold-standard ontology. Maedche and Staab (Maedche and Staab, 2002) evaluated the ontology by comparing it to a gold-standard ontology created by domain experts. They applied string matching techniques to measure the similarity between terms and properties in each candidate ontology and the gold-standard ontology and their approach was very useful for comparing and combining ontologies (Maedche et al., 2002). Using the gold-standard evaluation approach to evaluate the candidate domain ontologies used to auto-generate questions and feedback is infeasible due to the absence of a gold-standard domain ontology covering specific fields in the educational domain such as the computer networks field (Alsubait et al., 2014). Moreover, building a gold-standard ontology is an expensive and time-consuming process (Brank et al., 2007).

2. Assessment by humans: Evaluates the candidate ontology manually using domain experts to assess how well the ontology meets a set of pre-defined standards and requirements. Several tools such as the OntoMetric have been developed to help domain experts decide how well the ontology matches the application standards and requirements (Lozano-Tello and Gmez-Prez, 2004). OntoMetric provides domain experts with information about the candidate domain ontologies such as the content of the ontology, the ontology implementation language, and the ontology cost. Using the assessment by humans' evaluation approach to evaluate the candidate domain ontologies used to auto-generate questions and feedback is infeasible as it has several drawbacks. First, it is time consuming and sensitive to the subjective nature of human judgement. Second, the limited research time made it difficult to find domain experts trained to assess and evaluate the candidate domain ontologies.

3. Application-based evaluation: Uses the candidate ontology in an application and evaluate the application results obtained using the ontology. Porzel and Malaka (Porzel and Malaka, 2004) created an ontology and applied it to a speech recognition problem and then compared the results obtained using the ontology with gold-standard results provided by domain experts. Using this approach to evaluate the candidate domain ontologies used to auto-generate questions and feedback is infeasible because there are no gold-standard results available to compare with the candidate domain ontologies. For example, no information is available on what educational concepts the domain ontology should or should not cover

or what concepts are missing in the ontology.

4. Data-driven evaluation: Compare the candidate ontology with a source of data such as a collection of documents. Brewster (Brewster et al., 2004) evaluated ontologies by comparing them to a collection of textual documents (corpus) which captured concepts in a specific domain. Term extraction tools were used to extract terms from the corpus, and ontologies were evaluated by counting the number of these terms found in both the list of extracted terms and each candidate domain ontology. See Section 3.1 for more details about term extraction tools. The data-driven evaluation approach has been adopted in this thesis as it can utilise the materials used to teach the educational courses such as the textbooks and lecture slides, and there is no need for a gold-standard domain ontology or domain experts.

Using the data-driven ontology evaluation approach, the ontology could be evaluated on different levels (Brank et al., 2005):

1. Terminological level: Focuses on the concepts and individuals covered by the domain ontology. Evaluation at this level involves comparisons with various sources of data which represent the domain of interest (e.g., educational text corpora).

2. Syntactic level: At this level the ontology is evaluated to assess if it matches the syntactic requirements of the formal language adopted. For example, the syntax in the ontology definition is compared with the syntax specification of the formal language in which the ontology is written (e.g., RDF or OWL) (Brank et al., 2005).

3. Hierarchical or taxonomical level: This level evaluates the structural fit between the domain ontology and the corpus of documents.

4. Semantic level: This level evaluates if the ontology underlying semantic model represents the domain of interest correctly; i.e., it evaluates the correctness and completeness of the domain ontology concepts and properties used to capture the domain of interest (Rospocher et al., 2012).

Contribution 1 presented in this thesis concerns the terminological level, since it aims at assessing whether an ontology adequately covers the educational domain of interest, i.e., whether the concepts used in the domain ontology comprehensively represent the relevant terms of a domain. Table 2.5 illustrates the state of the art data-driven approaches used to evaluate the candidate domain ontologies at the terminological level. The approaches have been reviewed in this thesis according to nine criteria to demonstrate the need for a new approach to evaluate the candidate domain ontologies used to auto-generate assessment questions and feedback. The criteria are grouped into five categories which are discussed below.

General criteria

Among the evaluation approaches, only Jonquet[5], Mart$\tilde{i}$nez-Romero[6], and Rospocher's[7] data-driven approaches are available to end users. However, none of the approaches are open-source.

Candidate ontology

Jonquet and Rospocher's approaches allow users to upload their candidate domain ontologies. On the other hand, Mart$\tilde{i}$nez-Romero's approach is restricted to a set of built-in candidate ontologies in the medical domain and users cannot upload and evaluate their domain ontologies.

Input

The terminological data-driven approaches which are shown in Table 2.5 capture the domain of interest using terms extracted from a corpus and/or a set of terms defined by the user. Most approaches ignored the term's weight (reflects how important the term is to the domain of interest). This could negatively affect the ontology evaluation results as the ontologies may be penalised for missing some terms which are neither important to the user nor central to the domain of interest.

Table 2.5 shows that only Rospocher (Rospocher et al., 2012), and Jones and Alani's (Jones and Alani, 2006) terminological data-driven approaches considered the term's weight. Rospocher (Rospocher et al., 2012) extracted terms from the corpus

[5] http://bioportal.bioontology.org/recommender

[6] http://bioss.ontologyselection.com/Default.aspx

[7] https://moki.fbk.eu/moki/MoKiX/

using the *N*-gram extraction tool which extracts sequences of N words from the text (see Section 3.1 for more details), and assigned weight to each extracted term using the 'term frequency-inverse document frequency' recognition algorithm which reflects how important the term is to a document in a corpus (Salton and Buckley, 1988). Only terms with weight above an 0.01 arbitrary cut-off point were considered representative of the domain of interest and were used to evaluate the candidate domain ontologies (Rospocher et al., 2012). The choice of the 0.01 cut-off point in the list of extracted terms was not justified. Similar to Rospocher, Jones and Alani (Jones and Alani, 2006) considered the terms' weight. They selected the corpus based on a Google query inserted by the user and extended using WordNet terms (Fellbaum, 1998; Miller, 1995). Terms were extracted from the corpus and again assigned weight using the 'term frequency-inverse document frequency' recognition algorithm (Salton and Buckley, 1988). The top fifty terms with the highest weights were selected to evaluate the candidate domain ontologies. The choice of the top fifty terms was not justified and Jones and Alani did not reveal the term extraction tool used to extract terms from the corpus. Moreover, their approach only takes into account terms expressed by single words, which limits the efficiency of ontology evaluation, given that domain-specific concepts could be expressed by multiple word terms (Rospocher et al., 2012; Jones and Alani, 2006).

Pre-processing

The terms extracted from the corpus or defined by users could contain mistakes (e.g., spelling errors) which must be corrected to guarantee a proper ontology evaluation process. Moreover, the candidate ontology may not capture the exact same terms extracted from the corpus or inserted by the user which may also affect the ontology evaluation process. Therefore, to achieve better matching between terms and candidate domain ontologies, the input terms are expanded by finding their synonyms using WordNet (Fellbaum, 1998; Miller, 1995). Table 2.5 shows that only Mart$\tilde{i}$nez-Romero and Rospocher's evaluation approaches applied spell checking and expansion to the set of input terms. On the other hand, Jones and Alani's evaluation approach did not use the spell checking and only expanded the input terms.

Selection criteria

Table 2.5 shows that most of the terminological data-driven approaches focused on selecting the candidate domain ontology by assessing the ontology coverage of terms extracted from the corpus or defined by the user. As shown in Table 2.6 these approaches used different metrics to measure the ontology coverage. Brewster (Brewster et al., 2004) evaluated the candidate domain ontologies using Recall and Precision information retrieval metrics. Recall is the percentage of terms in the extracted list that also appear as concepts in the ontology, relative to the total number of terms in the list of terms extracted from the corpus (Powers, 2011; Manning et al., 2008). Precision is the percentage of the ontology concepts that also appear in the list of terms extracted from the corpus, relative to the total number of ontology concepts (Powers, 2011; Manning et al., 2008). Other terminological data-driven approaches evaluated the candidate domain ontologies using either Precision such as in Jonquet's approach (Jonquet et al., 2010) or Recall such as in Martĩnez-Romero's approach (Martnez-Romero et al., 2012; Martnez-Romero et al., 2014).

Rospocher (Rospocher et al., 2012) used similar metrics to Brewster, however, he also evaluated the candidate domain ontologies coverage using the F-measure metric which is a trade-off between Recall and Precision and it is measured using the harmonic mean of Recall and Precision (Powers, 2011; Manning et al., 2008). See Chapter 3, Section 3.1, page 57 for more details on Recall, Precision, and F-measure metrics.

In addition to the information retrieval metrics, some researchers such as Jonquet (Jonquet et al., 2010) and Jones and Alani (Jones and Alani, 2006) used the number of matching terms' between the list of terms extracted from the corpus and the candidate domain ontologies to measure the ontology coverage.

This evaluation metric is focused only on ranking the candidate domain ontologies and no attempts are made to find a relevance score that represents in absolute terms the quality of the ontology with respect to the domain of interest (Rospocher et al., 2012). In other words, if the same candidate domain ontology has the same number of matching terms with two different domains of interest, the user cannot judge which domain of interest is represented more precisely in the ontology.

Table 2.5: Terminological data-driven ontology evaluation approaches

Criteria		**Data-driven approaches**				
		Brewster	**Jonquet**	**Mart*ĩ*nez-Romero**	**Rospocher**	**Jones and Alani**
General	**Availability**	No	Tool, service	Tool, service	Tool, service	No
	Open-source	No	No	No	No	No
Candidate Ontology	**External Ontology**	N/A	Accept external ontologies	Does not accept external ontologies	Accept external ontologies	N/A
Input	**Types of input**	Corpus	Corpus/terms	Terms	Corpus	Corpus
	Weighted terms	No	No	No	Yes	Yes
Pre-processing	**Spell checking**	No	No	Yes	Yes	No
	Terms expansion	No	No	Yes	Yes	Yes
Selection Criteria	**Domain Coverage**	Yes	Yes	Yes	Yes	Yes
	Semantic Richness	No	No	Yes	No	No

Table 2.6: Ontology coverage metrics in terminological data-driven approaches

Terminological data-driven ontology evaluation approaches	**Ontology coverage metrics**			
	Recall	**Precision**	**F-measure**	**Number of matching terms**
Brewster (Brewster et al., 2004)	✓	✓		
Jonquet (Jonquet et al., 2010)		✓		✓
Martĩnez-Romero (Martnez-Romero et al., 2012; Martnez-Romero et al., 2014)	✓			
Rospocher (Rospocher et al., 2012)	✓	✓	✓	
Jones and Alani (Jones and Alani, 2006)				✓

The candidate domain ontologies could also be evaluated by measuring the semantic richness for each matching term (concept) between the candidate domain ontology and the lists of terms extracted from the corpus or defined by the user. The semantic richness is measured by summing the number of parent concepts, the number of children concepts, and the additional information contained within each concept such as the object, datatype, and annotation properties (Martnez-Romero et al., 2012; Martnez-Romero et al., 2014). Table 2.5 shows that only Martĩnez-Romero *et al.* terminological data-driven approach (Martnez-Romero et al., 2012; Martnez-Romero et al., 2014) evaluated the candidate ontologies by assessing their coverage and semantic richness based on a set of input terms. They stated that Ontologies that offer more detail for a given domain can be considered potentially more useful than less detailed ontologies (Martnez-Romero et al., 2014).

See Chapter 3, Section 3.1, page 57 for more details on the semantic richness metric.

Summary

The review of the terminological data-driven approaches revealed that only Martĩnez-Romero *et al.* evaluation approach evaluates the candidate domain ontologies coverage and semantic richness. The review also revealed that their approach evaluated the ontology coverage using the Recall metric which usually provides high Recall scores for large ontologies with broad scope that contains a particular domain of interest (e.g., OpenCyc (Matuszek et al., 2006)) compared to domain ontologies specifically developed to capture a particular domain of interest (e.g., Sakathi's Computer Networks ontology (Murugan.R et al., 2013)). Therefore, using only the Recall metric to evaluate the coverage of the candidate domain ontologies may result in selecting an ontology which has a broad scope, and consequently generate questions and feedback which are not related to the domain of interest. Other terminological data-driven approaches such as Jonquet's approach (Jonquet et al., 2010) measured the ontology coverage using the Precision metric which is sensitive to the ontology size, as small ontologies which are not related to the domain of interest may have higher Precision scores compared to large ontologies related to the domain of interest. This may again result in auto-generating questions and feedback, which are not related to the domain of interest. To provide balance between Recall and Precision, Rospocher (Rospocher et al., 2012) measured the ontology coverage using the F-measure metric, however, he did not measure the candidate domain ontologies' semantic richness which select the candidate domain ontology that offer more detail about the educational concepts. Both metrics are essential to assess the candidate domain ontologies used to auto-generate questions and feedback. This is due to the fact that the generators use a set of predefined generation strategies which exploit the relationships between concepts and the additional information contained within each concept such as the object, datatype, and annotation properties to generate assessment questions (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014) and formative feedback (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011; Duboc et al., 2011; Duboc, 2013). The generation strategies used by the generators could be categorised into the following: (a) the class-based strategies, which use the relationship between the ontology classes and individuals, (b) the terminology-based strategies, which use the relationship between the class and sub-class in ontologies, and (c) the property-based strategies, which use the object, datatype, and annotation properties in the ontologies. For example, Table 2.7 shows a multiple choice question generated using Cubric and Tosic generator (Cubric and Tosic, 2011). The question is generated using the annotation property in the OpenCyc ontology (Matuszek et al., 2006) which provides human readable descriptions to concepts in the domain ontology. Formative feedback generators also use the annotation property to provide students with information related to their answer. For example, if the student answered the multiple choice question by choosing `Network layer protocol`, the feedback generator will use the annotation property associated with the concept and provide the student with the following: 1) the correct answer is Transport layer protocol which splits a stream of data into datagrams. 2) Network layer protocol is responsible for routing data to its destination.

The review of the data-driven approaches also revealed that only Rospocher (Rospocher et al., 2012) and Jones and Alani's (Jones and Alani, 2006) terminological data-driven approaches considered the weight associated with terms extracted from the corpus. However, they both applied an arbitrary cut-off point to the list of terms extracted from

Table 2.7: Question generated using Cubric and Tosic question generator

Multiple Choice Question	
Stem	Read the paragraph and decide which one of the following concepts it defines? A _____ splits a stream of data into datagrams
Options	Transport layer protocol
	Network layer protocol
	Presentation layer protool

the corpus without any justification. This may affect the ontology coverage and semantic richness evaluation results, as terms used to represent the domain of interest could be extracted from the corpus using several term extraction tools such as the *N*-gram, and assigned weight using several term recognition algorithms such as the ‘term frequency-inverse document frequency’ recognition algorithm. Applying the same arbitrary cut-off to the lists of terms extracted using different term extraction tools and assigned weight using several term recognition algorithms has the following drawbacks:

1. High variance in the ontology coverage and semantic richness evaluation results: The same candidate domain ontology could have different coverage of the corpus when the lists of terms are extracted using different term extraction tools and assigned weight using several term recognition algorithms.

2. No agreement between the rankings of the candidate ontologies evaluation results: The terminological data-driven approaches evaluate the candidate domain ontologies coverage and semantic richness and rank the ontologies based on their coverage and semantic richness scores. Using the same arbitrary cut-off point in the lists of terms extracted from the same corpus may result in having the same candidate domain ontology being ranked the first (highest coverage score) in one list of terms, and ranked the last (lowest coverage score) when another list of terms is used in the ontology evaluation process.

To close the gaps in existing terminological data-driven approaches, this thesis presents a novel Terminological ONtology Evaluator (TONE) in Chapter 3 which is a generic approach that can be applied in learning environments. TONE achieves the following: 1) Assesses the candidate ontologies’ coverage using the Recall, Precision, and F-measure metrics. This will overcome the problems associated with the Recall and Precision metrics and help the user choose the candidate domain ontology which has the best representation of the subject domain of interest. 2) Measures the semantic richness for each candidate domain ontology. This metric will help teachers select the best candidate domain ontology which could be used to generate a wide range of assessment questions (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014) and formative feedback (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011; Duboc et al., 2011; Duboc, 2013) in existing generators. 3) Supports the ontology evaluation against both the input terms defined by a user and the corpus. Teachers will be able to use the educational course material (e.g., textbooks and lecture slides) to evaluate the candidate domain ontologies. 4) Dynamically select a cut-off point in the list of terms extracted from the corpus to reduce the number of irrelevant terms used to evaluate the candidate domain ontologies when different term extraction tools and terms recognition algorithms are used. This aims to reduce the variance in the ontology coverage and semantic richness results and improve the agreement between the rankings of the candidate domain ontologies evaluation results.

TONE will be used to select a candidate domain ontology which has high coverage and semantic richness of the educational domain. The selected ontology will be used after that by the Ontology-based Personalised Feedback Generator (OntoPeFeGe) to auto-generate assessment questions and feedback.

2.5 Chapter Summary

The purpose of this review was to identify the need for a novel, generic framework that can use pre-existing domain ontology to auto-generate assessment questions and different types of feedback, provides students with personalised feedback, and evaluates the candidate ontologies using the educational corpus.

Several personalised feedback frameworks have been surveyed (see Table 2.1) and the results revealed that most frameworks adapt the types of feedback towards student's characteristics and only Mason and Bruning's theoretical framework considered both the student and the task characteristics. The survey results also revealed that the types of feedback in the personalised feedback frameworks are either hard-coded in the system (Ana and Macario, 2009; Narciss and Huth, 2004; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010) or auto-generated from a restricted set of solutions created by the teacher or a domain expert (Mitrovic and Martin, 2000; Narciss et al., 2014). This has two main disadvantages: firstly, creating the different types of feedback in these frameworks is a time consuming process;

secondly, the frameworks are domain dependent and cannot be used to auto-generate feedback across different educational domains. Therefore, the thesis presents a novel generic framework which auto-generates several types of feedback across different educational domains using a broad knowledge base called ontology.

Providing students with personalised feedback after auto-generating different types of feedback requires information about the assessment question characteristics. Therefore, this thesis investigated several domain independent question generators (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014) which use an ontology to auto-generate several types of questions with different characteristics. In particular, questions aimed to assess students' cognition at different levels in Bloom's taxonomy (knowledge, comprehension, application, and analysis) (Bloom et al., 1956; Krathwohl, 2002). The ontology-based generation strategies used by the question generators and the stem templates were integrated into one system called Ontology-based Personalised Feedback Generator (OntoPeFeGe) and presented in Chapter 4. OntoPeFeGe auto-generates various types of assessment questions and associates each question with different types of feedback auto-generated from ontology. Moreover, OntoPeFeGe provides students with personalised feedback immediately after answering an assessment question by adopting Mason and Bruning's personalised feedback framework (see Section 4.2).

OntoPeFeGe could be applied to any educational domain where the concepts and properties are captured in an ontology. However, assessing how suitable the ontology is for generating questions and feedback, as well as how it represents the subject domain of interest, is a necessary preface to using the ontology in OntoPeFeGe. Several approaches have been proposed in the literature for evaluating ontologies and selecting the appropriate ontology for specific applications (Brank et al., 2005; Maedche and Staab, 2002; Lozano-Tello and Gmez-Prez, 2004; Porzel and Malaka, 2004; Brewster et al., 2004; Jonquet et al., 2010; Martnez-Romero et al., 2012; Martnez-Romero et al., 2014; Rospocher et al., 2012; Jones and Alani, 2006). These approaches focused on the ontology coverage of concepts in the domain of interest and ignored the semantic richness associated with each concept. Martĩnez-Romero *et al.* (Martnez-Romero et al., 2012; Martnez-Romero et al., 2014) addressed this issue by developing a system which evaluates ontologies by assessing their coverage and semantic richness based on a set of input terms. However, their system had several drawbacks which have been addressed in this thesis by presenting a Terminological ONtology Evaluator (TONE) in Chapter 3.

In addition to the importance of evaluating the candidate domain ontology used by the OntoPeFeGe, it is essential to ensure the quality of assessment questions generated using OntoPeFeGe. Previous evaluations of the auto-generated questions were confined to measuring the qualitative satisfaction of domain experts and students (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014). Therefore, this thesis evaluated the quality of the ontology-based auto-generated questions for the first time in Chapter 5 using two statistical methods: the Classical Test Theory (CTT) (Alagumalai and Curtis, 2005; Ding and Beichner, 2009; Doran, 1980; Cohen et al., 2013; Erguven, 2014) and the Item Response Theory (IRT) (Baker, 2001; De Ayala, 2009; Hambleton and Swaminathan, 1985; Hambleton, 1991; Reckase, 2009; Schmidt and Embretson, 2003; Toland, 2014).

Prior to the design and implementation of OntoPeFeGe, two preliminary studies are carried out and reported in Appendix A. The first study aims to examine the importance of building OntoPeFeGe, which focuses on the formative feedback students receive immediately after answering an assessment question. The second study aims to justify using Mason and Bruning's personalised feedback framework in OntoPeFeGe.

Chapter 3

Terminological ONtology Evaluator (TONE)

Selecting the best candidate ontology for question and feedback generation

Different domain ontologies may capture different concepts in the educational domain at different levels of detail. Thus, selecting the most appropriate candidate domain ontology is necessary to ensure that the questions and feedback auto-generated using the Ontology-based Personalised Feedback Generator (OntoPeFeGe) system covers the educational domain at a reasonable level of detail. None of the ontology based question and feedback generators explained in Section 2.3.3 evaluated the domain ontology used to auto-generate the assessment questions and feedback, and they only focused on the generation process.

This chapter presents Contribution 1: a Terminological ONtology Evaluator (TONE) for selecting the candidate domain ontology by evaluating the ontology coverage and the level of detail captured about the concepts in the educational domain (semantic richness). TONE extends the ontology evaluation approaches discussed in Section 2.4 by supporting the following:

1. Evaluate the candidate domain ontologies' coverage using the Recall, Precision, and F-measure metrics.

2. Measure the semantic richness for each candidate domain ontology.

3. Evaluate the candidate domain ontologies against the user defined input terms

and a corpus of text documents (e.g., textbooks and lecture slides).

4. Consider the input terms' weight and select a cut-off dynamically in the list of terms extracted from the educational corpus: The domain ontology evaluation metrics (e.g., coverage and semantic richness) highly depend on the list of terms extracted from the educational corpus, i.e., an increase in the number of irrelevant terms may results in selecting the inappropriate candidate domain ontology and consequently generating questions and feedback which are not related to the domain of interest. Therefore, it is essential to consider the weight (reflects how important the term is to the domain of interest) associated with terms extracted from the educational corpus. Most of the terminological data-driven ontology evaluation approaches in the literature ignored the terms' weight except Rospocher (Rospocher et al., 2012), and Jones and Alani's approaches (Jones and Alani, 2006). However, both approaches used an arbitrary cut-off point in the list of terms extracted from the corpus. TONE aims to select dynamically a cut-off point in the list of terms extracted from the educational corpus using several term extraction tools and assigned weight using different term recognition algorithms (e.g., Term Frequency). Selecting the cut-off dynamically aims to reduce the variance of the ontology evaluation coverage and semantic richness when different term extraction tools and term recognition algorithms are used. Moreover, it aims to improve the agreement between the rankings of the candidate domain ontologies evaluation results (coverage and semantic richness) when different term extraction tools and term recognition algorithms are used. Reducing the variance and improving the agreement help teachers and domain experts receive the same ranking results of the candidate ontologies regardless the term extraction tools and the term recognition algorithms used.

The chapter is structured as follows: Section 3.1 illustrates TONE, Section 3.2 explains the experiments carried out to evaluate TONE, and Section 3.3 concludes the chapter.

3.1 Domain Ontology Evaluation Approach

This section presents TONE which is used to select a candidate domain ontology for auto-generating questions and different types of feedback. Fig 3.1 is a graphical representation of TONE which is described in detail in the following steps all of which

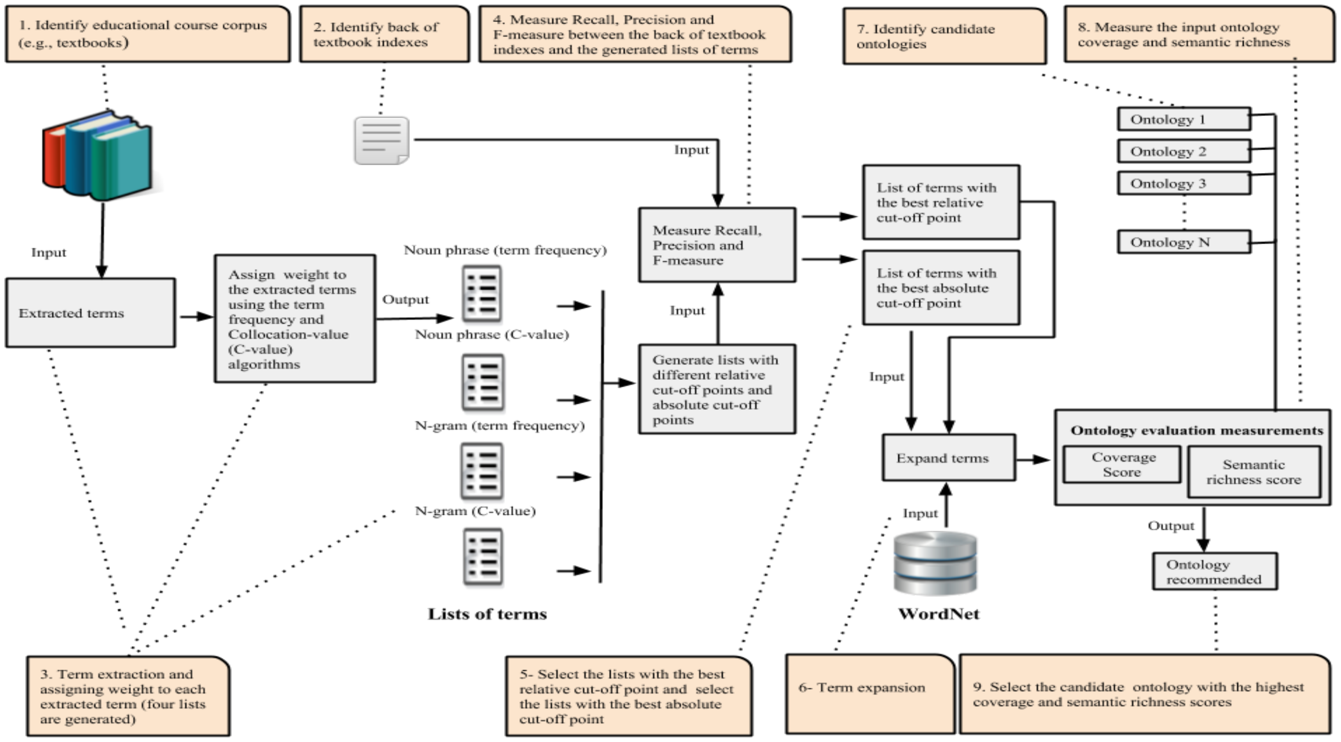

1. Identify educational course corpus (e.g., textbooks)
2. Identify back of textbook indexes
4. Measure Recall, Precision and F-measure between the back of textbook indexes and the generated lists of terms
7. Identify candidate ontologies
8. Measure the input ontology coverage and semantic richness
Input
Input
Extracted terms
Assign weight to the extracted terms using the term frequency and Collocation-value (C-value) algorithms
Output
Noun phrase (term frequency)
Noun phrase (C-value)
N-gram (term frequency)
N-gram (C-value)
Lists of terms
Measure Recall, Precision and F-measure
Input
Generate lists with different relative cut-off points and absolute cut-off points
List of terms with the best relative cut-off point
List of terms with the best absolute cut-off point
Input
Expand terms
Input
WordNet
Ontology 1
Ontology 2
Ontology 3
Ontology N
Ontology evaluation measurements
Coverage Score
Semantic richness score
Output
Ontology recommended
3. Term extraction and assigning weight to each extracted term (four lists are generated)
5- Select the lists with the best relative cut-off point and select the lists with the best absolute cut-off point
6- Term expansion
9. Select the candidate ontology with the highest coverage and semantic richness scores

Figure 3.1: Terminological ONtology Evaluator

are shown in the figure:

1) Identify Corpus

The corpus consists of the educational material used in teaching the educational course such as textbooks, lecture slides, and any trending topics related to the subject domain of interest. In addition, video and audio transcribed into textual format could be used.

2) Identify Back of Textbook Indexes

The back of textbook indexes contain the concepts in a document or a collection of documents (Csomai and Mihalcea, 2006), and they are usually created by domain experts (e.g., book's authors) who highlight the important terms in the educational domain (Huggett and Rasmussen, 2013). In case of a new emerging topic, teachers could add the new concepts to the back of textbook indexes.

3) Term Extraction

Term extraction is the process of extracting terms (lexical units) from text which is used in TONE to identify terms that characterise a specific domain of interest (Wong et al., 2008; Zhang et al., 2008). Term extraction techniques use linguistic, statistical or hybrid (combination of linguistic and statistical) approaches to extract simple (single) and complex (multiple words) terms from text (Wong et al., 2008; Pazienza et al., 2005). The linguistic approach uses the following (Pazienza et al., 2005):

1. Part of speech tagging: This is used to assign a grammatical tag (e.g., noun, adjective, and verb) to each word in the corpus.
2. Linguistic filter: It uses the part of speech tagging and permits the extraction of words with specific tags.
3. The stop-list: Is a list of words which are not accepted as a term in a specific domain (e.g., 'the' and 'a').

The statistical approach assigns a value to the extracted candidate terms using the 'unithood' and 'termhood' factors (Zhang et al., 2008; Pazienza et al., 2005). Unithood determines whether or not a sequence of words should be combined to form a stable term. On the other hand, termhood measures the degree to which these stable terms are related to domain specific concepts (Zhang et al., 2008). Unithood is only relevant to complex (multiple words) terms while termhood deals with both simple and complex terms. Term recognition algorithms such as the 'Term Frequency' (TF) and the Collocation value (C-value) use the 'termhood' and 'unithood' values to assign weight to the extracted terms (Zhang et al., 2008; Frantzi et al., 2000; Nakagawa, 2001).

TONE uses the hybrid approach and extracts terms from the educational corpus using the Java Automatic Term Extraction toolkit (jatetoolkit, 2011) (JATE) which contains several open Natural Language Processing[1] (openNLP) libraries for term extraction tools and term recognition algorithms (Foundation, 2010). The JATE toolkit processes the input corpus by applying the following steps:

1. Extracts the candidate terms from the educational corpus using two linguistic approaches: noun phrase (NP) extractor and *N*-gram (NG) extractor. The NP extractor applies heuristics to clean a candidate noun phrase and return it to a normalised root (e.g., the word 'networking' is normalised to become 'network'). The heuristics include the following:

    (a) The stop words are trimmed from the head and tail of a phrase. For example, 'the cat on the mat' becomes 'cat on the mat'.

    (b) Split phrases containing 'and' or 'or'. For example, ”Tom and Jerry” becomes 'Tom' 'Jerry'.

    (c) Terms must have letters and must have at least two characters.

    The NG extractor extracts sequences of N words from the text and uses statistical measurements to evaluate the probability that each of the sequences can be classified as a term. For example, the probability that two words can be considered bigrams increases when they appear more frequently together (Lopes et al., 2010). The NP extraction tool is linguistically more complex compared to the NG extraction tool; NP extraction tool extracts fewer terms from the corpus compared to the NG extraction tool (Lopes et al., 2010).

2. The candidate terms extracted using the NP and NG extractors were assigned weights and ranked in the list of extracted terms using the following two term recognition algorithms:

    (a) The Term Frequency (TF) algorithm: The candidate term weight is simply the number of times the candidate term appears in the corpus.

[1] openNLP is a ”machine learning based toolkit for the processing of natural language text. It supports the most common tasks, such as tokenization” (Foundation, 2010).

(b) The Collocation value (C-value) algorithm which measures the candidate term weight using the following formula (Frantzi et al., 2000):

$$C - value(a) = \begin{cases} \log_2 |a|.f(a) & \text{if `a' is not a substring} \\ \log_2 |a| - (\frac{1}{P(T_a)} \sum_{b \varepsilon T_a} f(b)) & otherwise \end{cases} \tag{3.1}$$

Where:

a: The candidate string.

$|a|$: The length of the candidate string (the number of words).

$f(a)$: The frequency of the candidate string ‘a’ in the corpus.

T_a: The set of candidate terms containing ‘a’.

$P(T_a)$: The number of candidate terms in the set T_a. It is used to measure the independence of term ‘a’ from the longer candidate terms in T_a. The higher the number of candidate terms in the set T_a, the bigger its independence.

$f(b)$: The frequency of the candidate term ‘b’ that contains ‘a’.

$\sum_{b \varepsilon T_a} f(b)$: The total frequency in which the candidate term ‘a’ appears in longer strings (b).

The weight assigned to the extracted terms were normalised (from zero to one). TF and C-value term recognition algorithms were chosen to investigate the effect of term recognition algorithms on the ontology evaluation metrics. While TF is a simple algorithm which assigns weight to each candidate term by counting the frequency of the term appearances in the corpus, C-value aims to improve the extraction of nested terms (terms which are substrings in other terms) (Frantzi et al., 2000). The C-value formula is computed based on three principles: extracting the most frequent terms, penalising nested terms that occur as a substring of longer candidate terms, and considering the length of the candidate terms (the number of words they consist of) (Knoth et al., 2009).

3. The extracted terms are normalised to a standard format: All upper case letters are converted to lower case letters, and the special characters such as ‘_’ were removed. After that, a spell checker is used to correct possible typographical errors.

The term extraction step outputs the following four lists of terms (see Fig 3.1):

1. NP (TF): A list of terms generated using the NP extraction tool and assigned weight using the Term Frequency (TF) algorithm.

2. NP (C-value): A list of terms generated using the NP extraction tool and assigned weight using the C-value algorithm.

3. NG (TF): A List of terms generated using the NG extraction tool and assigned weight using the TF algorithm.

4. NG (C-value): A list of terms generated using the NG extraction tool and assigned weight using the C-value algorithm.

The different term extraction tools and term recognition algorithms were used in TONE to investigate the following:

1. The effect of term extraction tools and term recognition algorithms on the ontology evaluation metrics (e.g., coverage and semantic richness).

2. The variance of the ontology evaluation metrics and the agreement between the rankings of the candidate domain ontologies evaluation results when an arbitrary cut-off point (Rospocher (Rospocher et al., 2012), and Jones and Alani's approaches (Jones and Alani, 2006)) or a dynamically derived cut-off point (TONE) is set in the list of terms extracted from the corpus.

4) Measure Recall, Precision and F-measure

The candidate terms in each of the extracted lists (the four lists of terms shown in Fig 3.1) are ordered according to their weights. Terms with low weight may be irrelevant to the domain of interest and if further used this could affect the ontology evaluation results. Therefore, TONE has the following two approaches, which aim to reduce the number of irrelevant terms: 1) TONE-R which aims to select the best relative cut-off point from a set of relative cut-off points. 2) TONE-A which aims to select the best absolute cut-off point from a set of absolute cut-off points.

The relative cut-off points are defined in each list of extracted terms by choosing the percentage of terms to maintain. For example, choose the top 10%, 20% and 30% of terms in the list. While, the absolute cut-off points depend on the extracted terms' weight assigned by the TF or the C-value algorithms. For example, choose terms with weight above 0.01.

The relative and absolute cut-off points were applied to each of the lists shown in Fig 3.1 which results in the following:

1. Forty-eight lists in TONE-R[2]: Twelve sub-lists were generated from NP (TF), NP (C-value), NG (TF), and NG (C-value) lists of terms by applying the following relative cut-off points: 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100%. 100% means all terms in the list were kept.

2. Forty lists in TONE-A[3]: Ten sub-lists were generated from NP (TF), NP (C-value), NG (TF), and NG (C-value) lists of terms by applying the following absolute cut-off points: 0.0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, and 0.09. A zero cut-off point means all terms in the list were kept.

To select the best relative cut-off point and the best absolute cut-off point in each of the four main lists (NP (TF), NP (C-value), NG (TF), and NG (C-value)), TONE compared each sub-list to the back of textbook indexes. The back of textbook indexes was used in TONE as a gold-standard reference list because they are usually created by a domain expert (book's author) who highlights the important terms in the educational domain. The Recall, Precision, and F-measure information retrieval metrics were used to evaluate each sub-list. Even though TONE used a gold-standard list, it is more feasible compared with existing gold-standard ontology evaluation approaches. The back of text book indexes are readily available and usually associated with the educational textbooks. However, building a gold-standard ontology is an expensive and time-consuming process (Brank et al., 2007), and the number of educational ontologies which capture concepts in educational courses is comparably low compared with back of text book indexes (Al-Yahya et al., 2015; Boyce and Pahl, 2007). The Recall is defined as the number of correctly extracted terms (matching terms between the extracted list and the gold-standard list) divided by the total number of terms in the gold-standard list (Powers, 2011; Manning et al., 2008):

$$Recall = \frac{\text{Extracted and correct terms}}{\text{Total number of terms in the gold-standard list}} \tag{3.2}$$

Precision is defined as the number of matching terms between the extracted list and the gold-standard list divided by the total number of terms in the extracted list (Powers,

[2] The four lists used in TONE are: NP (TF), NP (C-value), NG (TF), and NG (C-value). Each list was split into twelve sub-lists using several relative cut-off points.

[3] The four lists used in TONE are: NP (TF), NP (C-value), NG (TF), and NG (C-value). Each list was split into ten sub-lists using several absolute cut-off points.

Table 3.1: Example of terms expanded using WordNet

Terms	**Expansion**
FTP	file transfer protocol
Hypertext transfer protocol	HTTP
TCP	transmission control protocol
Network	Web, net, mesh, electronic network

2011; Manning et al., 2008):

$$Precision = \frac{\text{Extracted and correct terms}}{\text{Total number terms in the extracted list}} \tag{3.3}$$

There is a trade-off between Precision and Recall. An increase in Precision may result in a decrease in Recall. To be able to combine both the Precision and Recall in a single measure, the harmonic mean of both measures is used as an evaluation measure, and it is called the F-measure (Powers, 2011; Manning et al., 2008):

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.4}$$

5) Select the best cut-off point

The Recall, Precision and F-measure are measured between the sub-lists of extracted terms obtained in TONE-R and TONE-A and the back of textbook indexes (gold-standard reference list). After that, the best relative cut-off point and the best absolute cut-off point are determined by selecting a sub-list in each of the four main lists, which has the highest F-measure score.

6) Term expansion

The terms extracted are expanded using WordNet (Fellbaum, 1998; Miller, 1995) to find other terms with the same meaning (synonyms). The synonyms will be useful during the ontology coverage evaluation, to assess to what extent the terms or their synonyms are covered by the candidate domain ontologies. Table 3.1 shows an example of terms expanded using WordNet. The corpus could also be enriched with new terms added by domain experts, and synonyms found using other dictionaries such as Wiktionary (Meyer and Gurevych, 2012; Tarp, 2009).

7) Identify candidate ontologies

Identify the candidate domain ontologies which capture the concepts in a specific educational domain. For example, if the teacher is interested in auto-generating assessment questions and formative feedback in the computer networks domain, Sakathi's Computer Networks ontology could be used as a candidate domain ontology (Murugan.R et al., 2013). Candidate ontologies could be identified using several search engines such as; Swoogle (Ding et al., 2004).

8) Measure the coverage and semantic richness

After determining the candidate domain ontologies and the lists of terms that capture the educational domain, TONE uses several ontology evaluation metrics to recommend to the user the best candidate domain ontology for auto-generating assessment questions and formative feedback. This section presents the domain ontology evaluation metrics adopted by TONE.

Ontology coverage metrics

The ontology coverage measures how well the candidate ontology covers the terms extracted from the educational corpus by measuring the Recall, Precision, and F-measure between the list of terms and the concepts (classes) in the candidate domain ontology. The following illustrates the metrics adopted in TONE to measure the ontology coverage:

1. The Ontology Coverage Recall Score: It counts the number of matching terms between the list of extracted terms and the concepts (classes) in the domain ontology divided by the total number of terms extracted from the corpus (Brewster et al., 2004; Jonquet et al., 2010; Martnez-Romero et al., 2012; Martnez-Romero et al., 2014; Rospocher et al., 2012).

$$\text{Ontology Coverage Recall Score}\,(O,T) = \frac{\sum_{i=1}^{n} Matching(t_i,O)}{n} \tag{3.5}$$

   Where:

   O: Set of concepts in the candidate domain ontology

   T: Is a set of terms extracted from the corpus and their synonyms obtained using WordNet.

t_i: Is a term in the set of terms T

$Matching(t_i, O)$: Is a function that returns zero if there is no match between a term (t_i) in the list of extracted terms and O (set of concepts in the candidate domain ontology) and returns one otherwise

n: Is the total number of terms in the set of terms T

2. The Ontology Coverage Precision Score: It counts the number of matching terms between the list of extracted terms and the concepts (classes) in the domain ontology divided by the total number of concepts in the ontology (Brewster et al., 2004; Jonquet et al., 2010; Rospocher et al., 2012).

$$\text{Ontology Coverage Precision Score}\,(O,T) = \frac{\sum_{i=1}^{n} Matching(t_i, O)}{C_n} \tag{3.6}$$

Where:

O: Set of concepts in the candidate domain ontology

T: Is a set of terms extracted from the corpus and their synonyms obtained using WordNet.

t_i: Is a term in the set of terms T

$Matching(t_i, O)$: Is a function that returns zero if there is no match between a term (t_i) in the list of extracted terms and O (set of concepts in the candidate domain ontology) and returns one otherwise

C_n: Is the total number of concepts in O

Large candidate domain ontologies with broad scope (e.g., OpenCyc (Matuszek et al., 2006)) are expected to have higher Ontology Coverage Recall Score compared to small ontologies developed to capture a specific domain (e.g., Sakathi's Computer Networks ontology (Murugan.R et al., 2013)). On the other hand, large candidate domain ontologies are expected to have lower Ontology Coverage Precision Score compared to small candidate domain ontologies. Therefore, the F-measure is used to measure the harmonic mean between the Ontology Coverage Recall Score and the Ontology Coverage Precision Score.

Ontology semantic richness metric

Semantic richness is the level of details an ontology captures about a concept in the domain ontology. TONE evaluates the semantic richness for each concept in the candidate domain ontology which matches a term in the list of extracted terms by measuring the following (Martnez-Romero et al., 2012; Martnez-Romero et al., 2014):

1. Concept relatives (R_i): It is measured by summing the number of concept *i*'s superclasses, subclasses and sibling classes.
2. Concept additional information (A_i): It is measured by summing the number of object properties, datatype properties, and annotation properties associated with the concept *i*.
3. Similar concepts (S_i) in the domain ontology: It is the number of concepts in the candidate domain ontology that have the same meaning (synonymous) or have a name that contains concept *i*'s name.

The semantic richness is measured using the following formula (Martnez-Romero et al., 2012; Martnez-Romero et al., 2014):

$$Semantic Richness = \frac{\sum_{i=1}^{mt}(R_i + A_i + S_i)}{mt} \tag{3.7}$$

Where:

mt: Is the total number of matched terms between the concepts in the candidate domain ontology and the list of extracted terms

The Semantic Richness Score was normalised to transform the values from [0, + ∞) to [0, 1] and then multiplied by 100%. The score was divided by the maximum Semantic Richness Score obtained across the candidate domain ontologies (Romero et al., 2010).

9) Select the candidate ontology

TONE measures the ontology coverage and semantic richness for each candidate domain ontology. After that, TONE combines the coverage and semantic richness metrics using the following equation:

$$Score(O,T) = w_c \times \frac{F(O,T)}{max(F(O,T))} + (1 - w_c) \times \frac{SR(O,T)}{max(SR(O,T))} \tag{3.8}$$

Where:

O: Set of concepts in the candidate domain ontology.

T: Is a set of terms extracted from the corpus and their synonyms obtained using WordNet.

$F(O,T)$: Is the F-measure Score of the candidate domain ontology.

$SR(O,T)$: Is the Semantic Richness Score of the candidate domain ontology.

w_c: Is the weight assigned by the teacher to the F-measure coverage score and it has a value between 0 and 1.

Values of F-measure and semantic richness are normalised to be in the range [0, 1] by dividing them by the maximum value of the measure for all candidate ontologies. Finally, ontologies are ranked according to their score. TONE is an essential preface to the question and feedback generators as it helps the user select the most suitable candidate domain ontology for the generation process.

Having presented our TONE which could be used to select the best candidate domain ontology, the next section illustrates the experimental study used to evaluate TONE and compare it with existing ontology evaluation approaches.

3.2 Experimental Study

Two experiments are carried out to evaluate TONE. The first experiment aims to investigate the validity of TONE for selecting the best candidate domain ontology for generating questions and feedback. The second experiment aims to compare the variance of the ontology evaluation metrics and the agreement between the rankings of the candidate domain ontologies evaluation results in TONE-R and TONE-A with Rospocher (Rospocher et al., 2012) and Jones and Alani's (Jones and Alani, 2006) terminological data-driven approaches.

The following describes the corpus, the back of textbook index and the candidate domain ontologies used in the experiments.

Corpus

The corpus used in both experiments consists of the engineering educational textbooks and lecture slides used in teaching the Data Networking (of Manchester, 2013c) and the Computer Networks (of Manchester, 2013a) courses at the University of Manchester.

Table 3.2: Educational Materials

Educational Domain	**Corpus**	**Document Name**
The Data Networking and the Computer Networks courses	One textbook	Computer networking: a top-down approach (Kurose and Ross, 2013)
	Twenty-seven lecture slides	Introduction Application layer Link layer Transport layer Network layer Network security Performance

Table 3.2 illustrates the educational materials used in teaching both courses merged together. This corpus was chosen because the Ontology-based Personalised Feedback Generator (OntoPeFeGe) was used to auto-generate questions and formative feedback in the domain of computer networks (see Chapters 5 and 6).

Back of textbook indexes

The 'Computer networking: a top-down approach, 6^{th} edition' back of textbook index was used as a gold-standard reference list and it consists of 2442 terms.

Candidate domain ontologies

The literature was reviewed to find the candidate domain ontologies that capture concepts in the computer networks domain. Unfortunately, the number of educational ontologies which capture concepts in educational courses is low (Al-Yahya et al., 2015; Boyce and Pahl, 2007). Therefore, only two candidate domain ontologies were found which capture concepts in the computer networks domain: the Computer Networks ontology (Murugan.R et al., 2013) and the OpenCyc ontology (Matuszek et al., 2006).

The Computer Networks ontology consists of 537 classes and it has been developed to cover different fields in the computer networks domain such the network topology, the communication's medium, and the Open Systems Interconnection model (Murugan.R et al., 2013). The second ontology was OpenCyc, a large scale general purpose ontology which covers several domains and consists of approximately 239,000 terms (Matuszek et al., 2006). Both ontologies capture concepts related to the computer networks domain. However, they capture different concepts at a different level of detail. In addition to the Computer Networks and OpenCyc domain ontologies, the following two domain ontologies which do not capture the computer networks domain were also used in the evaluation to investigate the behaviour of TONE for evaluating the candidate domain ontologies:1) Pizza ontology: Describing the domain of pizza, including pizza types, toppings, and bases. The ontology consists of ninety-seven classes

(Stanford, 2012). 2) C programming language ontology: Captures general concept in C programming and consists of forty-four classes (lorecommender, 2013). Having candidate ontologies with similar specifications will result in having similar results. Therefore, the choice of the ontologies was done deliberately to evaluate the behavior of TONE when ontologies with different specifications are used as candidates. Both the Computer Networks ontology and the OpenCyc are related to the corpus. However, they differ in the size (number of classes) and how precisely they cover the corpus. On the other hand, The Pizza and C programming were chosen as they are not related at all to the computer networks domain.

3.2.1 Validity of TONE

This experiment aims to achieve the following three goals: 1) Investigate the capability of TONE to evaluate the candidate domain ontologies by assessing their terminological coverage and semantic richness with respect to the corpus illustrated in Table 3.2. 2) Examine the relationship between the ontology evaluation metrics and the lists of terms extracted using different term extraction tools and assigned weights using the TF and C-value algorithms. 3) Select the best candidate domain ontology for generating assessment questions and formative feedback from the domain ontology. The best candidate domain ontology is the ontology which has the highest coverage and semantic richness.

The experiment was carried out after determining the best relative cut-off point and the best absolute cut-off point for each of the NP (TF), NP(C-value), NG (TF) and NG (C-value) lists of terms which are shown in Fig 3.1. Table 3.3 illustrates the best relative cut-off point and the best absolute cut-off point for each list which were determined using the back of textbook index.

TONE-R and TONE-A were used to evaluate the Pizza, C programming language, CN, and OpenCyc domain ontologies against the lists of extracted terms. The ontologies were chosen to examine the capability of TONE to distinguish between domain ontologies which capture concepts in the computer networks domain (CN and OpenCyc) and the domain ontologies which are not related to the computer networks domain (Pizza and C programming language). The coverage and semantic richness scores were measured for the candidate domain ontologies and the results can be summarised as follows:

Table 3.3: The best relative and absolute cut-off points for different lists of terms

List of Terms	Best Relative Cut-off	Best Absolute Cut-off
NP(TF)	10%	0.01
NP(C-value)	20%	0.04
NG(TF)	1%	0.01
NG(C-value)	1%	0.04

Coverage Score

The Recall, Precision and, F-measure Scores are measured for each candidate domain ontology using TONE-R and TONE-A.

1) Coverage Recall Score

The Coverage Recall Scores obtained using TONE-R are shown in Fig 3.2. The results revealed that OpenCyc ontology had the highest scores followed by the CN, Pizza, and C Programming Language ontologies regardless of the term extraction tools and the term recognition algorithms used. For OpenCyc, the Coverage Recall Scores were approximately 100 times larger than the Pizza and C Programming Language ontologies. Moreover, the CN's Coverage Recall Scores were approximately ten times larger than the scores for the Pizza and C Programming Language ontologies.

Comparing the Coverage Recall Scores of the OpenCyc and the CN ontologies revealed that the OpenCyc's Coverage Recall Scores were approximately eight times larger than the CN. The results are expected, as OpenCyc is a large ontology with a broad scope compared to CN, which was intentionally developed to capture the computer networks domain.

The Coverage Recall Scores obtained using TONE-A are approximately similar to the scores measured using TONE-R (see Fig 3.3). OpenCyc ontology had the highest scores followed by CN, Pizza, and C Programming language ontologies regardless of the term extraction tools and the term recognition algorithms used. OpenCyc had Coverage Recall Scores approximately 100 times higher than the Pizza and C programming language ontologies, and approximately eight times higher than the CN ontology.

Fig 3.2 and Fig 3.3 also show that using the NP (C-Value) and NG (C-value) lists of terms to evaluate the candidate domain ontologies results in low Coverage Recall Scores. This is due to the fact that the C-value improves the extraction of multiple word terms from the corpus compared to the TF term recognition algorithm which only assigns a frequency to each extracted term. Therefore, more multiple word terms were at the top of the NP(C-value) and NG (C-value) lists and fewer terms were matched between the lists of extracted terms and the candidate domain ontologies.

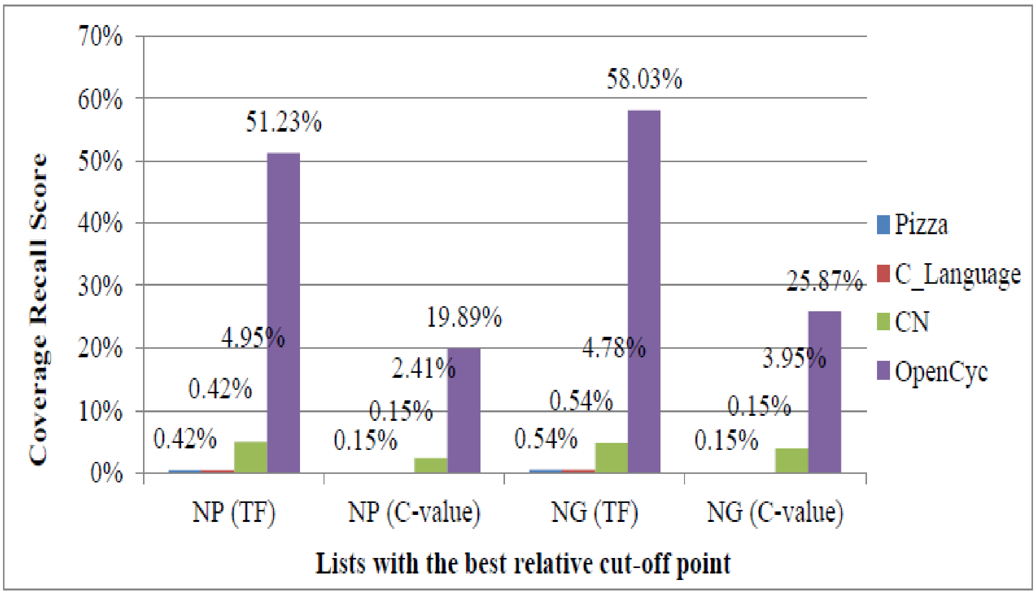


Figure 3.2: Coverage Recall Score obtained using TONE-R

The Coverage Recall Score in TONE-R and TONE-A was able to distinguish between the candidate domain ontologies related to the computer networks domain (OpenCyc and CN) and the candidate domain ontologies which capture concepts unrelated to the domain (Pizza and C Programming Language). However, the ontology Coverage Recall Score revealed that OpenCyc, which is a large ontology with broad scope had higher scores compared to the CN ontology which was specifically developed to capture the computer networks domain. This suggests that relying only on the Coverage Recall Score in evaluating the candidate domain ontologies such as in Mart$\tilde{i}$nez-Romero's approach (Martnez-Romero et al., 2012; Martnez-Romero et al., 2014) may result in generating questions and feedback which are not related to the domain of interest.

2) Coverage Precision Score

The Coverage Precision Score was also measured for the candidate domain ontologies using TONE-R and TONE-A. Fig 3.4 illustrates the Coverage Precision Score obtained using TONE-R. OpenCyc had very low Coverage Precision Scores compared to the CN, Pizza, and C Programming Language ontologies due to the ontology size (239,000 terms). Moreover, CN ontology had lower Coverage Precision Score compared to the C Programming Language ontology when NP (TF) and NG (TF) lists of terms were used. The results suggest that Precision is sensitive to the ontology size as the number of concepts in the C Programming language ontology (44 classes) is very low

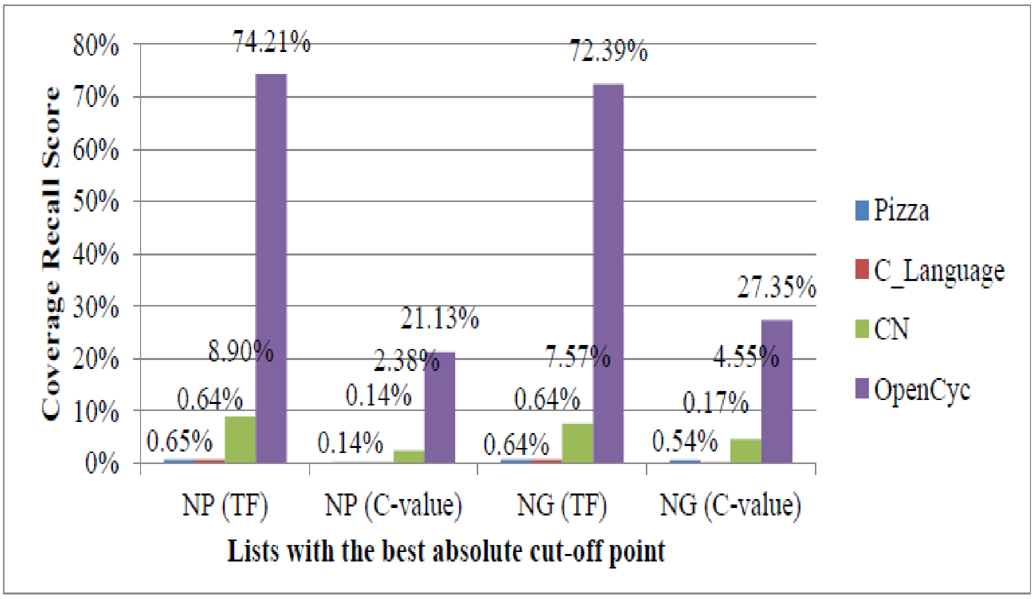


Figure 3.3: Coverage Recall Score obtained using TONE-A

compared to the CN ontology (537 classes) and OpenCyc ontology. Moreover, Fig 3.4 shows that using the TF algorithm to assign weight to the terms extracted from the corpus increased the chance of having more matching terms with the candidate domain ontologies (Pizza and C Programming language).

On the other hand, the CN ontology had higher precision compared to the Pizza and C Programming language ontologies when NP (C-value) and NG (C-value) lists of terms were used. This is due to the fact that more multiple word terms were at the top of the list of extracted terms, and less matching terms occurred between the lists and the candidate domain ontologies.

The Coverage Precision Score obtained using TONE-A also shows that the CN ontology had lower Coverage Precision Score compared to the C Programming Language when NG (TF) list of terms were used (see Fig 3.5). Moreover, the CN ontology had very close Coverage Precision scores to the C Programming Language ontology when NP (TF) list is used.

The results suggest that using the C-value algorithm may be a better approach than TF to distinguish between the candidate domain ontologies related to the domain of interest (CN ontology) and the candidate domain ontologies unrelated to the domain of interest (Pizza and C Programming Language). This is due to the fact that the C-value algorithm reduces the number of matching terms between the list of terms and the candidate domain ontologies unrelated to the domain of interest. Moreover, the results

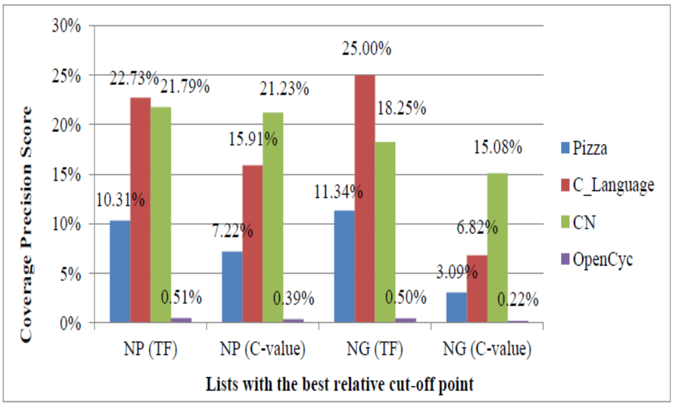


Figure 3.4: Coverage Precision Score obtained using TONE-R

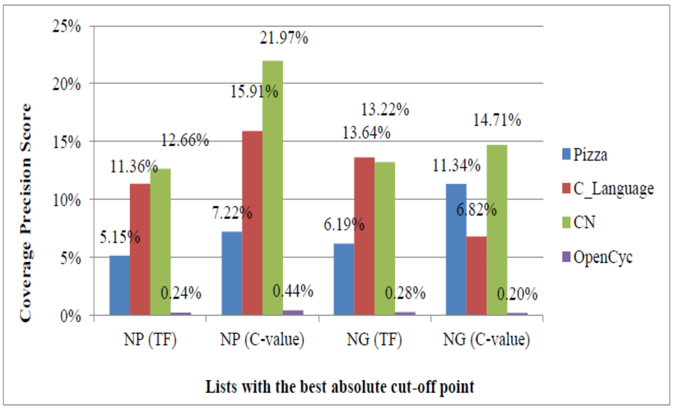


Figure 3.5: Coverage Precision Score obtained using TONE-A

show that the Coverage Precision Score is sensitive to the ontology size as OpenCyc which has more matching terms compared to the CN, Pizza and C Programming Language ontologies had the lowest Coverage Precision Score due to the ontology large size (239,000 terms).

3) Coverage F-measure Score

The Coverage F-measure Score is used to provide a balance between the Coverage Recall Score and the Coverage Precision Score. Fig 3.6 shows the Coverage F-measure Scores obtained in TONE-R. The results show that the CN ontology which was intentionally developed to capture concepts in the computer networks domain had the highest score regardless of the term extraction tools and the term recognition algorithms used. However, OpenCyc ontology which captures concepts related to the computer networks domain had close Coverage F-measure scores to the Pizza and C-Programming Language ontologies which are not related to the computer networks

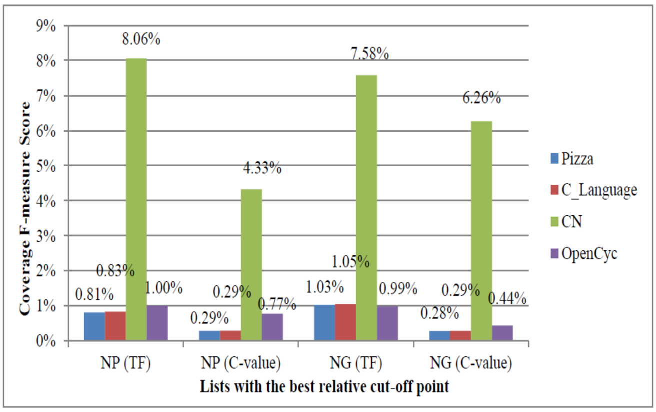


Figure 3.6: F-measure Score using TONE-R

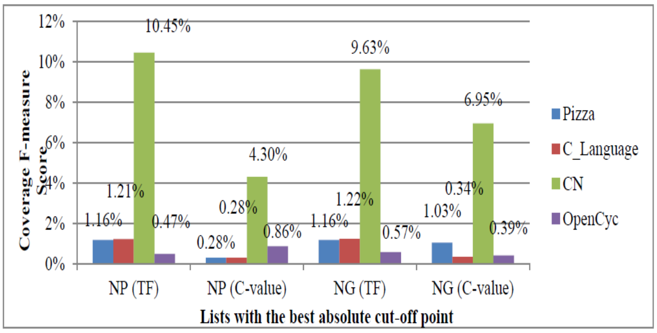


Figure 3.7: F-measure Score using TONE-A

domain.

The Coverage F-measure Score was also obtained using TONE-A (see Fig 3.7). Similar to TONE-R, CN had the highest Coverage F-measure Score regardless of the term extraction tools and the term recognitions algorithms used. Fig 3.6 and Fig 3.7 also show that similar to the Coverage Recall Scores, the Coverage F-measure Scores were higher for the lists of terms assigned weight using the TF algorithm.

Semantic Richness Score

The Semantic Richness Score was measured for each matching term (concept) between the list of extracted terms and the candidate domain ontology by summing the number of concept's relatives, concept's additional information, and the number of similar concepts in the domain ontology (see Equation (3.7)). The Semantic Richness Scores ranges between 0% and 100%, where 100% represents the highest Semantic Richness Score obtained among the candidate domain ontologies. In this experiment, OpenCyc had the highest Semantic Richness Score due to the following:

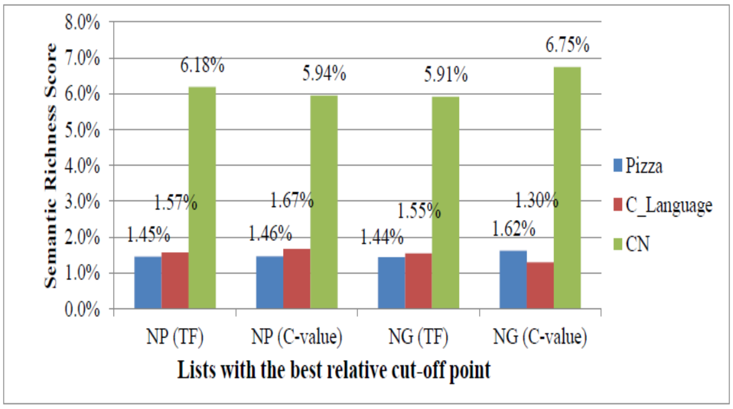


Figure 3.8: Semantic Richness Score obtained using TONE-R

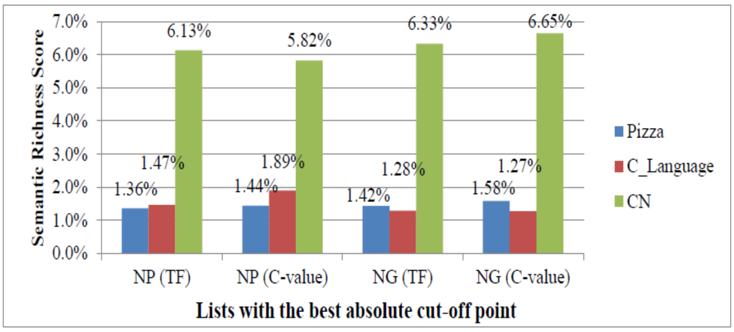


Figure 3.9: Semantic Richness Score obtained using TONE-A

1. The high number of matching terms between the lists of terms extracted from the corpus and OpenCyc.

2. The high number of the concept's relatives, the concept's additional information and the similar concepts in OpenCyc.

Therefore, OpenCyc was used to normalise the Semantic Richness Scores obtained in the Pizza, C Programming Language and the CN ontologies. Fig 3.8 and Fig 3.9 show the Semantic Richness Scores for the candidate ontologies obtained using TONE-R and TONE-A. The Semantic Richness Score for CN ontology in TONE-R and TONE-A is four times higher than the Pizza and C Programming Language ontologies. Fig 3.8 and Fig 3.9 also show that different Semantic Richness Scores were obtained in TONE-R and TONE-A when different term extraction tools and term recognition algorithms were used:

1. The CN ontology Semantic Richness Scores in TONE-R did not have any relationship with the term extraction tools and the term recognition algorithms.

On the other hand, the CN ontology Semantic Richness Scores in TONE-A had higher scores when the NG extraction tool was used compared to the Semantic Richness Scores obtained when terms were extracted using NP regardless of the terms recognition algorithm used.

2. The Pizza Language ontology in TONE-R and TONE-A had higher Semantic Richness Scores when terms were assigned weight using the C-value algorithm compared to the Semantic Richness Scores obtained when terms were assigned weight using the TF algorithm regardless of the term extraction tool used.

3. The C Programming Language ontology in TONE-R and TONE-A had higher Semantic Richness Scores when the terms were extracted using NG compared to the Semantic Richness Scores obtained when terms were extracted using NP regardless of the term recognition algorithm used.

Summary

TONE-R and TONE-A are able to distinguish between domain ontologies related to the domain of interest (CN ontology) and domain ontologies which cover different domains (Pizza and C Programming Language). However, similar to other terminological data-driven approaches (Rospocher et al., 2012; Jones and Alani, 2006) TONE-R and TONE-A are not able to differentiate between domain ontologies with broad scope which capture concepts related to the domain (OpenCyc) and domain ontologies which are not related to the domain of interest (Pizza and C Programming Language). Fig 3.6 and Fig 3.7 show that OpenCyc ontology had close Coverage F-measure Scores to the Pizza and C Programming Language ontologies.

The results also revealed that the CN ontology which was intentionally developed to capture concepts in the computer networks domain had the highest Coverage F-measure Score regardless of the list of terms used to represent the corpus. On the other hand, the Semantic Richness Score revealed that OpenCyc had the highest scores due to the high number of matching terms between the ontology and the lists of extracted terms.

TONE provides Teachers with the best candidate ontology which they could use later on to auto-generate questions and feedback. Fig 3.10 and Fig 3.11 illustrate that TONE selected the CN and OpenCyc ontologies as they had the highest scores using Equation 3.8. The w_c was assigned a 0.5 value.

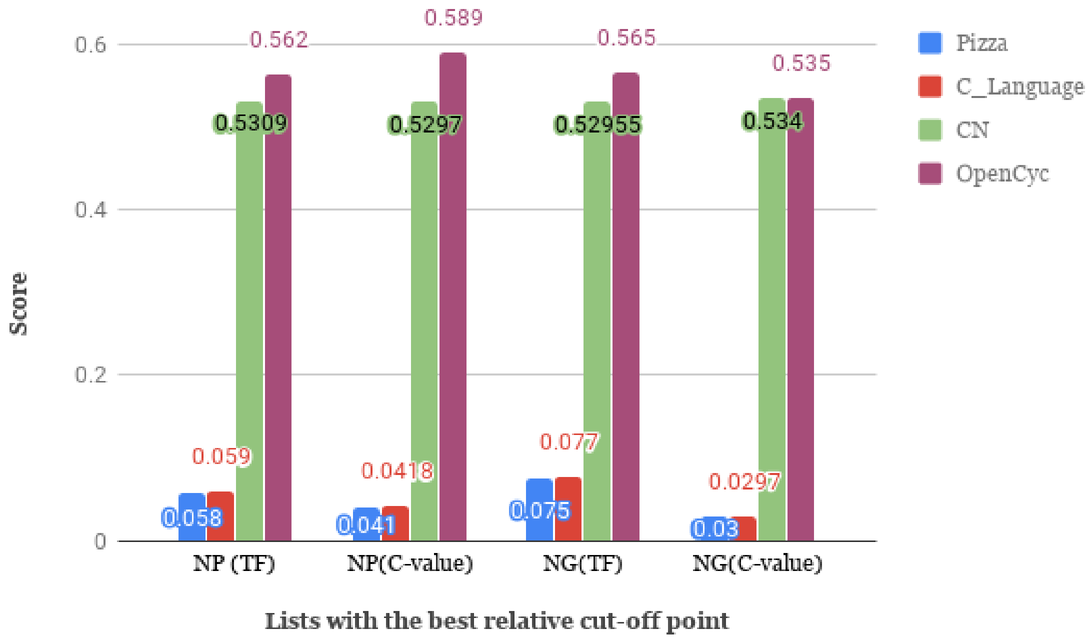


Figure 3.10: Score obtained using TONE-R

The experiment also investigated the relationship between the term extraction tools, the term recognition algorithms and the ontology evaluation metrics. The results revealed that using different term extraction tools and term recognition algorithms affects the ontology evaluation metrics.

3.2.2 TONE in comparison with other terminological data-driven approaches

The second experiment aims to compare the variance of the ontology evaluation metrics and the agreement between the rankings of the candidate domain ontologies evaluation results (coverage and semantic richness) in TONE-R and TONE-A with Rospocher (Rospocher et al., 2012) and Jones and Alani's (Jones and Alani, 2006) terminological data-driven approaches.

The ontology evaluation metrics used by the terminological data-driven approaches depend on the input terms extracted from the corpus. Rospocher's data-driven approach extracted terms from the corpus using the NG extraction tool and used a 0.01 absolute cut-off point in the list of extracted terms (Rospocher et al., 2012). Jones and Alani (Jones and Alani, 2006) followed a different approach as they extracted terms from the corpus, ranked them according to their weight, and then used the top fifty

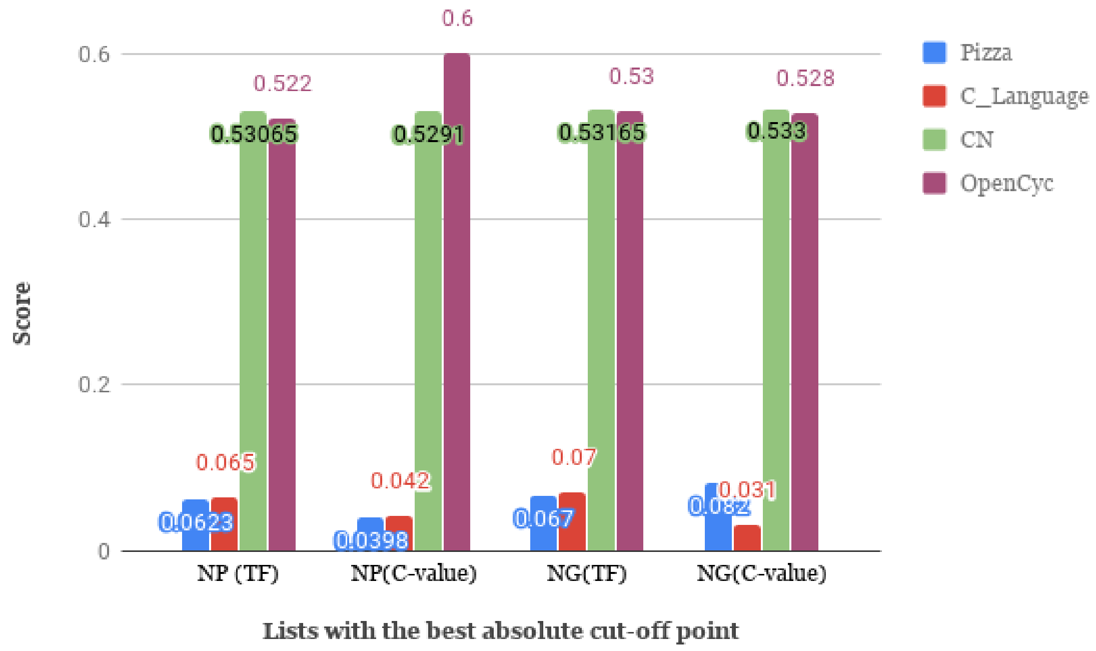


Figure 3.11: Score obtained using TONE-A

terms with the highest weights to evaluate the candidate domain ontologies. Using the arbitrary cut-off points defined by both approaches may result in the following drawbacks:

1. Having high variance in the candidate domain ontology metrics when terms used to evaluate the candidate domain ontologies are extracted from the corpus using different term extraction tools and assigned weight using several term recognition algorithms. For example, the same candidate domain ontology may have high coverage of the educational domain when the list of terms used is NP (TF) with a 0.01 absolute cut-off point, and low coverage of the same educational domain when the list of terms used is NG (TF) with a 0.01 absolute cut-off point.

2. Having no agreement between the rankings of the candidate domain ontologies evaluation results: The terminological data-driven approaches evaluate the candidate domain ontologies coverage and semantic richness and rank the ontologies based on their coverage and semantic richness scores. Using the same arbitrary cut-off point in the lists of terms extracted from the corpus may result in having the same candidate domain ontology being ranked the first among the candidate domain ontologies (e.g., the ontology has the highest coverage score) when one list of terms is used and ranked the last (lowest coverage score) when

another list of terms is used even if both lists are extracted from the same educational corpus.

The drawbacks mentioned above may result in choosing an unsuitable candidate domain ontology for auto-generating questions and feedback. Therefore, TONE-R and TONE-A set the best cut-off point in the list of terms extracted from the corpus dynamically by choosing the cut-off point which has the highest F-measure score between the list of terms and the back of textbook index. This aims to reduce the number of irrelevant terms in the list of extracted terms used to evaluate the candidate domain ontology, and consequently reduce the variance of the ontology evaluation metrics and improve the agreement between the ontologies rankings when different term extraction tools and term recognition algorithms are used.

The experiment presented in this section evaluates the variance and agreement of the ontology evaluation metrics (Coverage Recall Score, Coverage Precision Score, Coverage F-measure Score, and Semantic Richness Score) obtained using TONE-R and TONE-A and compare them to Rospocher (Rospocher et al., 2012), and Jones and Alani's terminological data-driven approaches (Jones and Alani, 2006). The variance in each approach was measured by averaging the variance of the ontology evaluation metrics obtained in each candidate domain ontology across the lists of terms extracted using several term extraction tools and assigned weight using different term recognition algorithms.

The agreement between the rankings of the candidate domain ontologies evaluation results in each approach was measured using the percentage agreement (Cooper et al., 2009):

$$\text{Percentage agreement} = \frac{\text{Number of observations agreed upon}}{\text{Total number of observations}} \tag{3.9}$$

The percentage agreement in each approach was averaged across the lists of terms extracted using several term extraction tools and assigned weight using different term recognition algorithms. The variance and percentage agreement results for TONE-R, TONE-A, Rospocher, and Jones and Alani's terminological data-driven approaches could be summarised as follows:

Coverage score

The Coverage Recall Score, the Coverage Precision Score, and the Coverage F-measure score were measured for the Pizza, C Programming Language, CN, and OpenCyc

Table 3.4: Variance and percentage agreement in the Coverage Recall Score

Terminological data-driven approach	Variance in the Coverage Recall Score	Percentage agreement in the Coverage Recall Score
TONE-R	87.8	75%
TONE-A	204.5	100%
Rospocher's approach (Rospocher et al., 2012)	613.9	75%
Jones and Alani's approach (Jones and Alani, 2006)	105.3	100%

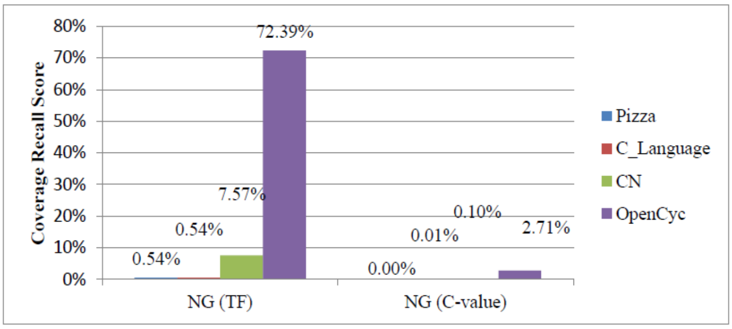


Figure 3.12: Coverage Recall Scores obtained using Rospocher's approach

ontologies using TONE-R, TONE-A, Rospocher's data-driven approach (Rospocher et al., 2012), and Jones and Alani's data-driven approach (Jones and Alani, 2006).

1) Coverage Recall Score

The variance in the Coverage Recall Scores between the terminological data-driven approaches is shown in Table 3.4. TONE-R had the lowest variance compared to the other terminological data-driven approaches (see Fig 3.2). Fig 3.12 illustrates the Coverage Recall Score obtained using Rospocher's data-driven approach. Similar to TONE-R and TONE-A, the OpenCyc domain ontology had the highest Coverage Recall Score. However, the figure shows that there is a high variance in the Coverage Recall Scores between NG (TF) and NG (C-value) for the Pizza, C Programming Language, CN and OpenCyc ontologies. Fig 3.12 also shows that the OpenCyc ontology had 72.39% Coverage Recall Score of the educational corpus when NG (TF) list was used and 2.71% Coverage Recall Score for the same corpus when NG (C-value) was used. This huge difference in the Coverage Recall Scores is due to the fact that a 0.01 absolute cut-off point is reasonable for the NG (TF) list of terms (Table 3.3 shows that using TONE-A the best absolute cut-off point in the NG (TF) list is 0.01). However, using a 0.01 absolute cut-off point in the NG (C-value) list results in having more irrelevant terms to the domain of interest, and consequently lower Coverage Recall Scores and higher variance.

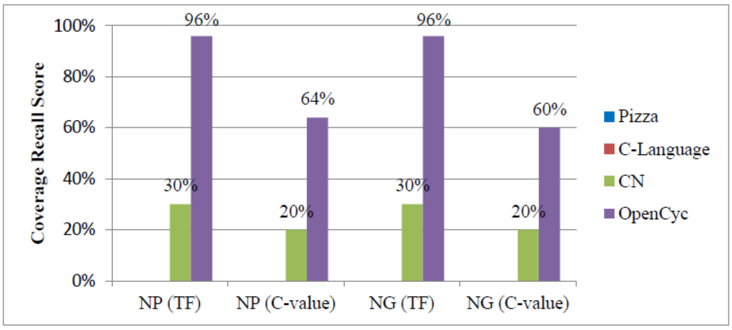


Figure 3.13: Coverage Recall Scores obtained using Jones and Alani's approach

Fig 3.13 shows that Jones and Alani's approach also revealed that OpenCyc had the highest Coverage Recall Score. The variance in the Coverage Recall Scores (105.3) is less than Rospocher's data-driven approach (613.9). The figure also shows that the Pizza and C Programming language ontologies had 0% Coverage Recall Score. TONE-A (see Fig 3.2) had similar behaviour to Jones and Alani's approach, as the terms assigned weight using TF had close Coverage Recall Scores regardless of the term extraction tool used. The same applied to terms assigned weight using C-value. However, in Jones and Alani's approach the lists of terms assigned weight using the same term recognition algorithm had closer Coverage Recall Scores and thus the Coverage Recall Score is less variant compared to TONE-A.

The percentage agreement between the rankings of the Coverage Recall Score in the four terminological data-driven evaluation approaches was also examined (see Table 3.4) and the results revealed that all approaches had a percentage agreement equal or above 75%. Table 3.4 also shows that TONE-A had 100% percentage agreement compared to Rospocher's approach which suggests that setting the cut-off point dynamically in the list of extracted terms improves the percentage agreement.

2) Coverage Precision Score

Table 3.5 shows that Jones and Alani's approach had the lowest variance in the Coverage Precision Score, followed by TONE-A, TONE-R and Rospocher's approach. Fig 3.14 shows that Rospocher's approach has high variance in the Coverage Precision Score. While Fig 3.15 shows that using Jones and Alani's approach the candidate domain ontologies had close Coverage Precision Scores when different lists of terms were used.

Table 3.5: Variance and percentage agreement in the Coverage Precision Score

Terminological data-driven approach	Variance in the Coverage Precision Score	Percentage agreement in the Coverage Precision Score
TONE-R	22.5	54%
TONE-A	10.2	67%
Rospocher's approach (Rospocher et al., 2012)	121.7	100%
Jones and Alani's approach (Jones and Alani, 2006)	0.1	100%

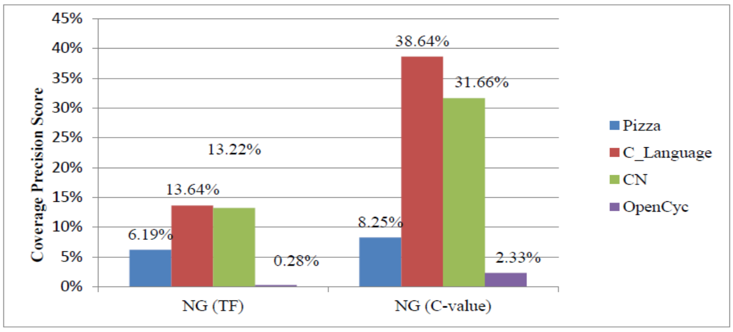


Figure 3.14: Coverage Precision Scores obtained using Rospocher's approach

Table 3.5 shows that even though Rospocher's approach had high variance compared to TONE-R and TONE-A. His approach had higher percentage agreement in the rankings of the Coverage Precision Scores (100%).

3) Coverage F-measure Score

Table 3.6 shows that Jones and Alani's approach had the lowest variance in the Coverage F-measure Score (see Fig 3.16), followed by TONE-R, TONE-A and Rospocher's approach. Fig 3.17 shows that similar to the Coverage Recall and Precision Scores, the Coverage F-measure Score had high variance in Rospocher's approach.

The percentage agreement between the rankings of the candidate domain ontologies was also measured and the results revealed that TONE-R and TONE-A had higher percentage agreement compared to Rospocher's approach (see Table 3.6).

Table 3.6: Variance and percentage agreement in the Coverage F-measure Score

Terminological data-driven approach	Variance in the Coverage F-measure Score	Percentage agreement in the Coverage F-measure Score
TONE-R	0.8	38%
TONE-A	2.1	63%
Rospocher's approach (Rospocher et al., 2012)	11.9	0%
Jones and Alani's approach (Jones and Alani, 2006)	0.2	100%

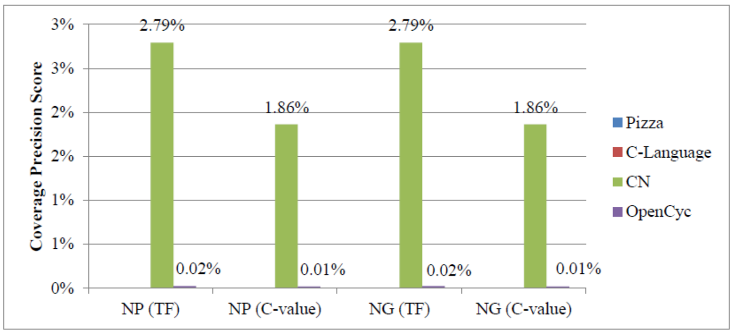


Figure 3.15: Coverage Precision Scores obtained using Jones and Alani's approach

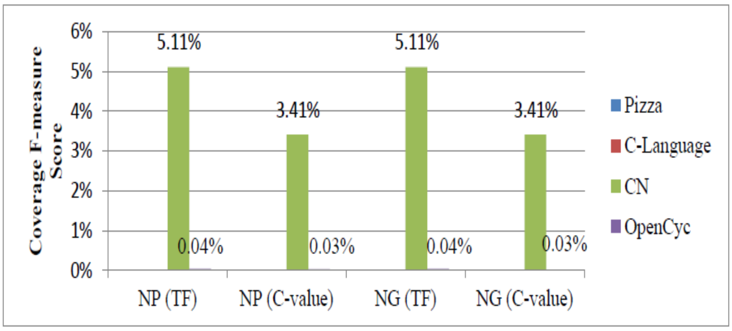


Figure 3.16: Coverage F-measure Scores obtained using Jones and Alani's approach

Semantic Richness Score

Fig 3.18 shows the Semantic Richness Score of the CN candidate domain ontology obtained using TONE-R, TONE-A, Rospocher, and Jones and Alani's approaches. The figure shows that TONE-R and TONE-A had the lowest variance across the lists of extracted terms. The semantic Richness was also obtained for the Pizza and C Programming Language ontologies and the average variance for each data-driven approach across the candidate domain ontologies is shown in Table 3.7. Table 3.7 also shows that TONE-R and TONE-A had higher percentage agreement compared to Rospocher's approach.

Summary

The experiment revealed that setting the cut-off point dynamically in the lists of extracted terms using TONE-R and TONE-A results in a less variant Semantic Richness Score compared to other terminological data-driven approaches and higher percentage agreement compared to Rospocher's approach. Moreover, TONE-R and TONE-A had lower variance and higher percentage agreement in the F-measure Scores compared

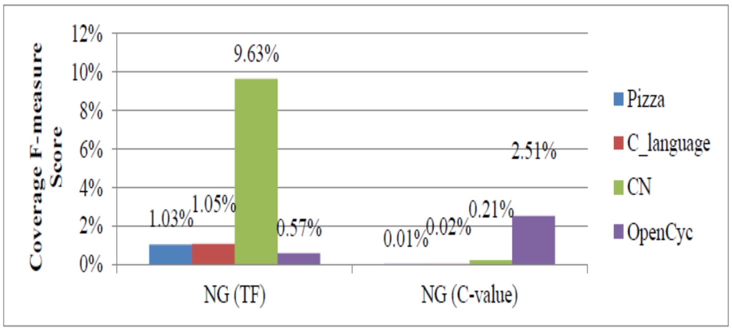


Figure 3.17: Coverage F-measure Scores obtained using Rospocher's approach

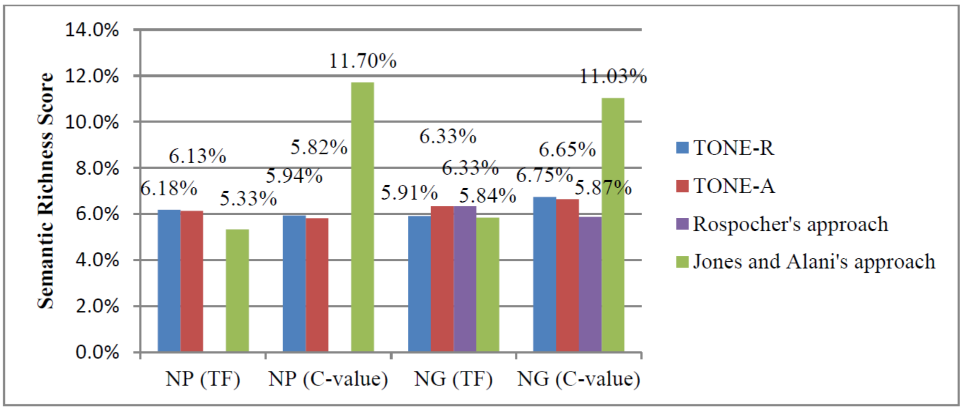


Figure 3.18: Semantic Richness Score in different terminological data-driven approaches

to Rospocher's approach. This helps the user obtain approximately similar ontology evaluation metrics regardless of the list of terms (NP (TF), NP (Cvalue), NG (TF), NG (C-value)) used during the ontology evaluation process.

On the other hand, Jones and Alani's approach which only considered the top fifty terms in the lists of terms extracted from the corpus had lower variance in the Coverage Precision and F-measure Scores compared to the other terminological data-driven approaches and 100% percentage agreement in the coverage and semantic richness rankings. However, using TONE-R and TONE-A to select the best candidate domain ontology for auto-generating questions and feedback is more reasonable than using Jones and Alani's approach due to the following:

1. Jones and Alani only used the top fifty terms expressed by single words to capture the domain of interest (e.g., computer networks). This limits the efficiency of ontology evaluation, given that domain-specific concepts could be expressed

Table 3.7: Variance and percentage agreement in the Semantic Richness Score

Terminological data-driven approach	Variance in the Semantic Richness Score	Percentage agreement in the Semantic Richness Score
TONE-R	0.1	56%
TONE-A	0.1	67%
Rospocher's approach (Rospocher et al., 2012)	6	0%
Jones and Alani's approach (Jones and Alani, 2006)	3.7	100%

by multiple word terms (Rospocher et al., 2012; Jones and Alani, 2006). For example, in the Computer Networks domain ontology 50% of the concepts (269 out of 537) were expressed by multiple word terms. Similarly, 57% of the concepts (55 out of 97) in the Pizza domain ontology, and 23% of the concepts (10 out of 44) in the C Programming Language ontology were expressed by multiple word terms.

2. Jones and Alani's approach was applied to a corpus formed from the first 100 pages returned by Google search engine after querying a term which captures the domain of interest. This will result in having documents in the corpus which are not related to the course material. For example, querying the computer networks term using Google search engine returns many news articles.

3.3 Chapter Summary

This chapter presents Contribution 1: a Terminological ONtology Evaluator (TONE) for selecting the candidate domain ontology which covers the educational domain at a reasonable level of detail. TONE evaluates the candidate domain ontologies coverage and semantic richness and extends the ontology evaluation approaches discussed in Section 2.4 by:

1. Evaluating the candidate domain ontologies coverage and semantic richness against a corpus of text documents (e.g., textbooks and lecture slides).

2. Considering the weight associated with each extracted term during the ontology evaluation process.

Rospocher (Rospocher et al., 2012), and Jones and Alani's (Jones and Alani, 2006) terminological data-driven ontology evaluation approaches used the term's weight during the ontology evaluation process. Rospocher (Rospocher et al., 2012) extracted terms from the corpus using the *N*-gram extraction tool and only terms with weight above an 0.01 absolute cut-off point were considered representative of the domain of interest, and were used to evaluate the candidate domain ontologies (Rospocher et al., 2012). The choice of the 0.01 cut-off point in the list of extracted terms was not justified. Similar to Rospocher, Jones and Alani's (Jones and Alani, 2006) considered the terms' weight and the top fifty terms with the highest weights were selected to evaluate the candidate domain ontologies. The choice of the top fifty terms was not justified. Moreover, their approach only takes into account terms expressed by single words, which limits the efficiency of ontology evaluation given that domain-specific concepts could be expressed by multiple word terms (Rospocher et al., 2012; Jones and Alani, 2006). TONE selected dynamically the best relative cut-off point (TONE-R) and the best absolute cut-off point (TONE-A) in the list of extracted terms to reduce the number of irrelevant terms used to evaluate the candidate domain ontologies when different term extraction tools and terms recognition algorithms are used. Moreover, TONE supports several term extraction tools which extract single and multiple words and assigns weight to each extracted term using several term recognition algorithms (e.g., Term Frequency).

TONE-R and TONE-A were examined on several candidate domain ontologies, and the results revealed that TONE-R and TONE-A are able to distinguish between domain ontologies related to the domain of interest (CN ontology) and domain ontologies which cover different domains (Pizza and C Programming Language). However, similar to other terminological data-driven ontology evaluation approaches discussed in (Rospocher et al., 2012; Jones and Alani, 2006), TONE-R and TONE-A are not able to differentiate between the domain ontologies with broad scope (e.g., OpenCyc) which capture concepts related to the domain of interest and domain ontologies which are not related to the domain of interest (e.g., Pizza and C Programming Language ontologies).

The variance of the ontology evaluation metrics adopted in TONE-R and TONE-A were compared against Rospocher (Rospocher et al., 2012), and Jones and Alani's (Jones and Alani, 2006) terminological data-driven approaches. The results revealed that TONE-R and TONE-A had less variance in the Semantic Richness Score compared to the other terminological data-driven approaches. Moreover, TONE-R and TONE-A had lower variance in the Coverage Recall, Precision and F-measure Scores compared to Rospocher's approach. The percentage agreements between the rankings of the ontology evaluation results were also examined and the results revealed that

TONE-R and TONE-A had higher percentage agreement in the F-measure Score and Semantic Richness Score compared to Rospocher's approach. This helps the user obtain approximately similar ontology evaluation metrics regardless of the list of terms (NP (TF), NP (C-value), NG (TF), NG (C-value)) used during the ontology evaluation process.

TONE-R and TONE-A were used to select the best candidate domain ontology for auto-generating questions and feedback in the computer networks domain. The evaluation results in this chapter revealed that the CN ontology had the highest coverage scores compared to the other candidate domain ontologies, while OpenCyc ontology had the highest semantic richness scores. Therefore, both ontologies were used to auto-generate the questions and feedback which were analysed in Chapters 5 and 6 respectively.

After presenting TONE, the next chapter introduces OntoPeFeGe (Contribution 2) which will use the candidate ontology selected by TONE to auto-generate assessment questions and feedback, and provide students with personalised feedback (Contribution 3).

Chapter 4

Ontology-based Personalised Feedback Generator (OntoPeFeGe)

This chapter presents a generic personalised feedback framework which auto-generates different types of feedback from the domain ontology and provides students with personalised feedback. The chapter is divided into two main Sections. Section 4.1 introduces Contribution 2 which is a feedback generator prototype that aims to overcome drawbacks in the existing feedback generators discussed in Section 2.3.3 by achieving the following:

1. **Auto-generate different types of formative feedback:** The preliminary study carried out and reported in Appendix A to investigate the different types of feedback teachers provide to students immediately after answering an assessment question revealed that the teachers provided students with the following types of feedback: Knowledge Of Results (KOR), Knowledge of Correct Response (KCR), Bugs-Related (BR), Topic Contingent (TC), and Response Contingent (RC). These types of feedback were either neglected (Kazi *et al.* (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013) focused on auto-generating hint feedback) or partially supported (Snchez-Vera *et al.* (Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011) focused on auto-generating KOR and KCR feedback) by the feedback generators introduced in Section 2.3.3. Therefore, the feedback generator presented in this chapter aims to support the generation of the KOR, KCR, BR, TC, and RC feedback.

2. **Auto-generate domain independent feedback using pre-existing domain ontology:** The feedback generators introduced in Section 2.3.3 were domain dependent, as Kazi *et al.* (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013) auto-generated the hint feedback using the UMLS domain ontology in addition to an expert knowledge base which captures the experts' solutions to the problem scenario. The expert solution represents the cause-effect relationships between the medical concepts, and it is time consuming to construct as each solution requires about three to four hours. S*ã*nchez-Vera *et al.* (S et al., 2012; Castellanos-Nieves et al., 2011) generated feedback based on both the course domain ontology, and the annotations created by the teacher and used to identify the expected answer for each assessment question. Duboc *et al.* (Duboc et al., 2011; Duboc, 2013) generated feedback using the links between the conditions, symptoms and diagnosis classes created by the author of example patient cases.

3. **Auto-generate assessment questions using pre-existing domain ontology:** The feedback generators discussed in Section 2.3.3 hard-coded the assessment questions, which mean that the questions are only valid in the educational domain they are created in. In addition, the feedback generators did not specify the question's characteristics (e.g., the level of the question in Bloom's taxonomy) (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Duboc et al., 2011; Duboc, 2013; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011). These limitations hinder providing students with personalised feedback in a generic framework where the types of feedback are adapted based on the student's and the question's characteristics.

Section 4.2 in this chapter presents Contribution 3, which is a personalised formative feedback algorithm that adopts Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001), and adapts the feedback based on student's background knowledge about a specific educational topic, student's current level of knowledge while the student is answering one question after another, and the question's level in Bloom's taxonomy. The algorithm aims to overcome the drawbacks in the feedback generators discussed in Section 2.3.3, which follow the 'one size fits all' approach and provide students with the same type of feedback when their answers to a specific question are identical regardless of the differences in student's characteristics and the question's characteristics.

The feedback generator and the personalised feedback algorithm are introduced in one system called the Ontology-based Personalised Feedback Generator (OntoPeFeGe).

4.1 Generator

Fig 4.1 shows the feedback generator prototype which auto-generates questions and associates each question with different types of formative feedback. The feedback generator takes the domain ontology which captures the concepts in an educational course as an input and outputs Q questions associated with T types of feedback.

The feedback generator generates different types of questions (true and false, multiple choice, and short-answer) and different types of feedback using the ontology-based generation strategies defined by Papasalouros (Papasalouros et al., 2008; Papasalouros et al., 2011), Grubisic (Grubisic, 2012; Grubisic et al., 2013), Al-Yahya (Al-Yahya, 2011; Al-Yahya, 2014), Cubric, and Tosic (Cubric and Tosic, 2011) (see Section 2.3.3). The ontology-based generation strategies traverse the domain ontology to achieve the following:

1. Determine the classes, individuals (in this thesis the word individual is used to represent an instance in the domain ontology (Group, 2004)), or properties in the domain ontology, which are used to auto-generate the question's correct answer (Key) and the question's wrong answers (Distractors).

2. Instantiate a set of stem templates (the text stating the question) similar to those identified by Grubisic, Cubric, and Tosic which assess student's cognition at different levels in Bloom's taxonomy.

3. Instantiate a set of feedback templates, which represent the KOR, KCR, BR, TC, and RC types of feedback.

4.1.1 Technical Design

The feedback generator was implemented in Java using the Jena API (Foundation,). The main classes used to manage the generation of assessment questions and the different types of formative feedback are depicted in the Unified Modelling Language (UML) diagram shown in Fig 4.2 and are described as follows:

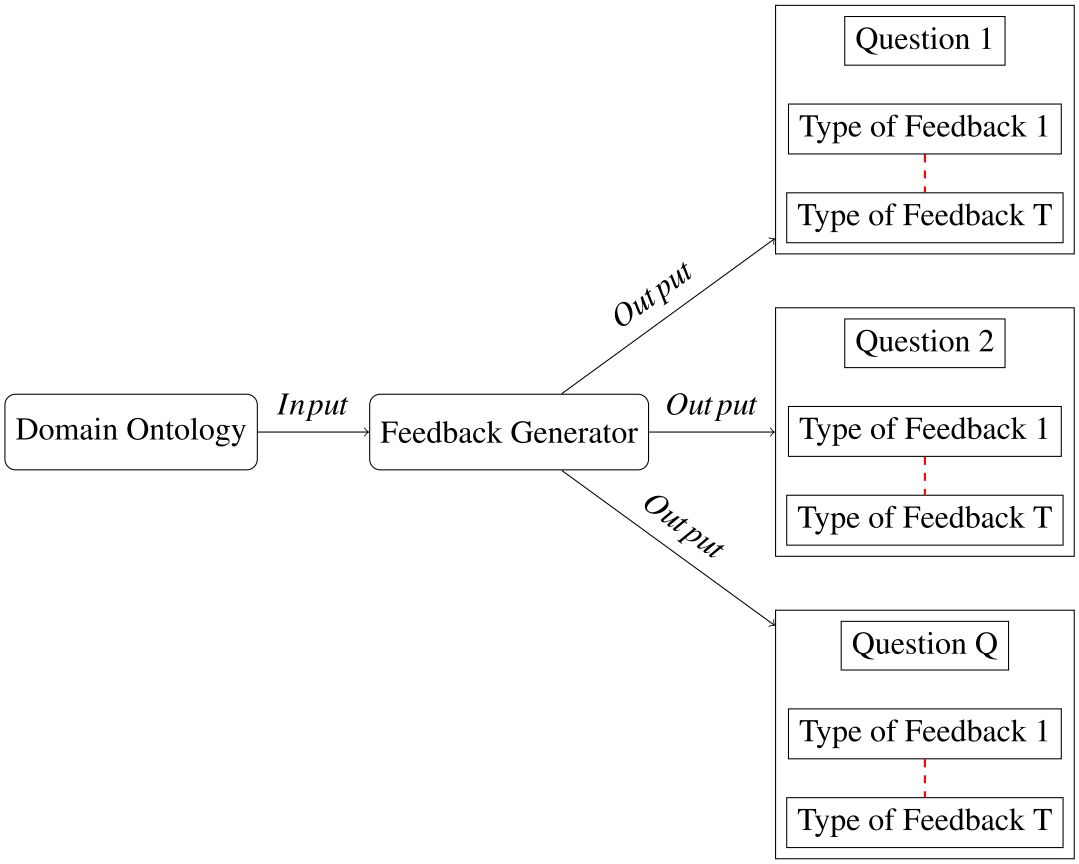


Figure 4.1: Formative feedback generator

- Ontology-based generation strategy factory: It creates strategies that are used to generate questions and feedback from the knowledge represented in domain ontologies. The class takes two ontologies as an input. The first ontology is the domain ontology which captures the educational concepts of a specific course (e.g., the Computer Networks ontology created by Sakathi (Murugan.R et al., 2013)) and it is used to auto-generate the assessment questions and the pedagogical content of the different types of formative feedback. The second ontology is called the OntoPeFeGe question and feedback ontology and it is developed in the OntoPeFeGe to structure the auto-generated questions and their feedback, so that they can be mapped into several standard formats supported by different Virtual Learning Environments (VLEs). Examples of the standardised formats used in VLEs are Instructional Management System Questions and Test Interoperability (IMS QTI) format (GLOBAL, 2012), and Moodle XML format (Moodle, 2012). Currently, only the mapping to Moodle XML format has been implemented, since it was needed for the experimental purposes of this research presented in Chapters 5 and 6. More details about the OntoPeFeGe question and feedback ontology are presented in Section 4.1.2.

- Ontology-based generation strategy: There are different possible strategies for generating questions from domain ontologies (see Section 2.3.3). This class is an abstract class which is inherited by each class implementing a specific ontology-based generation strategy. The diagram also shows that there are several class-based strategies, terminology-based strategies and property-based strategies. More details about the strategies are provided in Section 4.1.4.

- Question: This class is initiated in each ontology-based generation strategy class using the create question function (see the createQuestion function in Fig 4.2). Each question has a Bloom's taxonomy level, stem, options and formative feedback.

- Formative feedback: This class is an abstract class for the different types of feedback (KOR, KCR, BR, TC, and RC) generated in OntoPeFeGe.

4.1.2 The OntoPeFeGe Question and Feedback Ontology

This section illustrates the OntoPeFeGe question and feedback domain ontology, which is created in OntoPeFeGe using the Web Ontology Language (OWL) to structure the

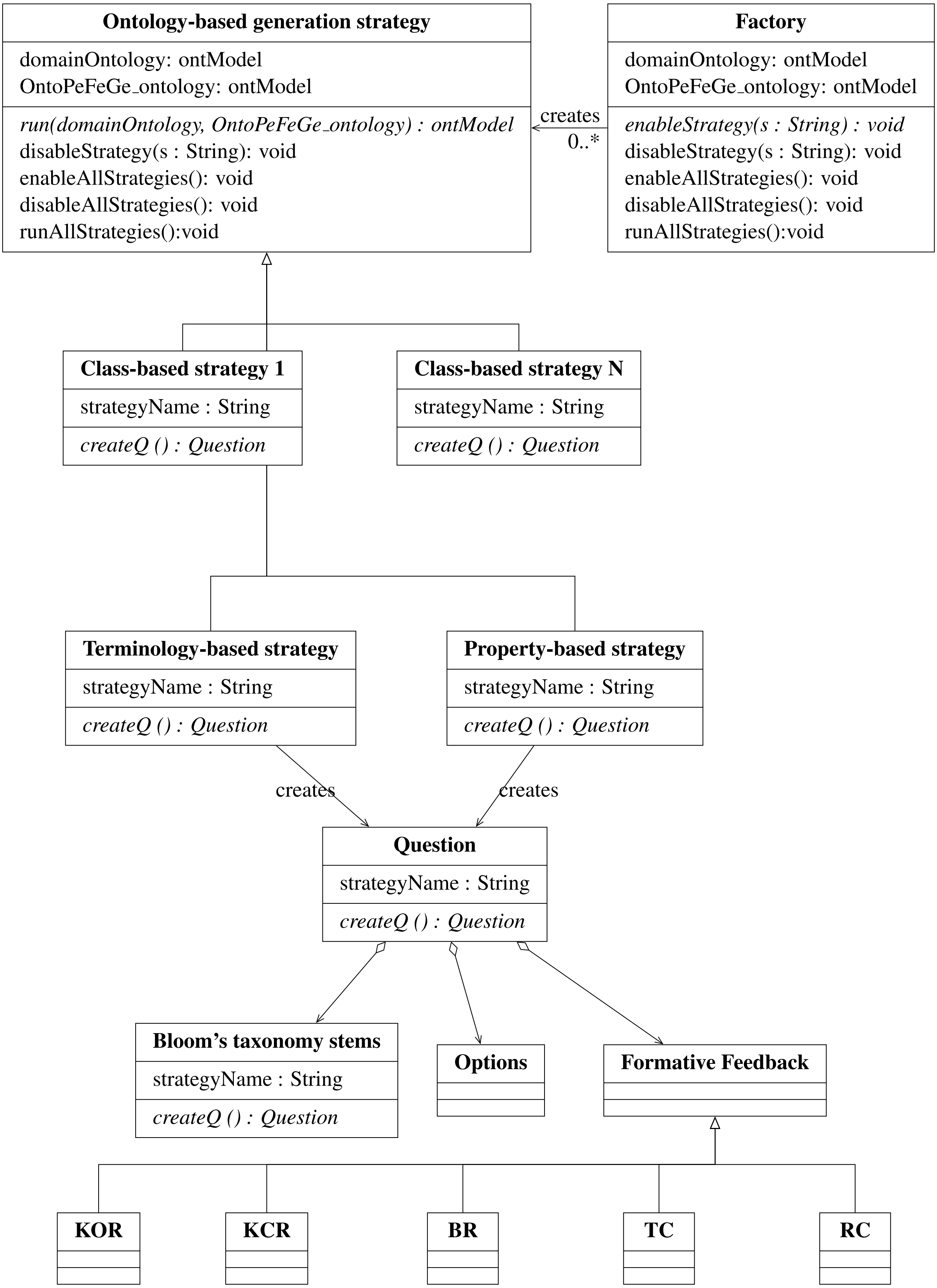


Figure 4.2: UML class diagram of the feedback generator

auto-generated questions and feedback (see Fig 4.3 and Fig 4.4). The ontology consists of classes, individuals, and properties. It is developed to capture the relationships between the assessment questions, Bloom's taxonomy levels, the ontology-based generation strategies, and the different types of formative feedback. During the question and feedback generation process, the ontology is populated with individuals of questions each of which is associated with individuals of different types of feedback (KOR, KCR, BR, TC and RC).

Fig 4.3 shows the three types of questions (true and false, multiple choice and short answer) which are auto-generated in OntoPeFeGe and captured in the OntoPeFeGe question and feedback ontology. The ontology shows that the Question is part of a Test and it has two subclasses: the Multiple choice question and the Short answer question. In addition, the ontology shows that the True and false question is a subclass of the Multiple choice question.

Questions are composed of several components. For example, a Multiple choice question consists of three main components:

1. The Stem, which is the text stating the question.

2. Options, which are a set of possible answers and contains both the Key which is the correct answer and the Distractors which are the wrong answers.

3. Feedback, which has different types and it is associated with the question's Key and the question's Distractors in the multiple choice question example.

The ontology also shows that the Feedback class has several subclasses that represent the different types of feedback (KOR, KCR, BR, TC, and RC) auto-generated in OntoPeFeGe, and a pedagogical content property with a string value which is populated with the feedback pedagogical content during the generation process. The ontology also captures the ontology-based generation strategies used to auto-generate the questions and the different types of feedback. OntoPeFeGe implements several ontology-based generation strategies derived from the literature (class-based, terminology-based, and property-based) (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Hardouin and Mesbah, 2004; Al-Yahya, 2011; Al-Yahya, 2014). Fig 4.4 shows that each question and feedback is generated using an ontology-based generation strategy. The class, terminology, and property-based generation strategies consist of several substrategies. For instance, the class-based strategy is currently associated with five sub-strategies, which traverse the domain ontology to

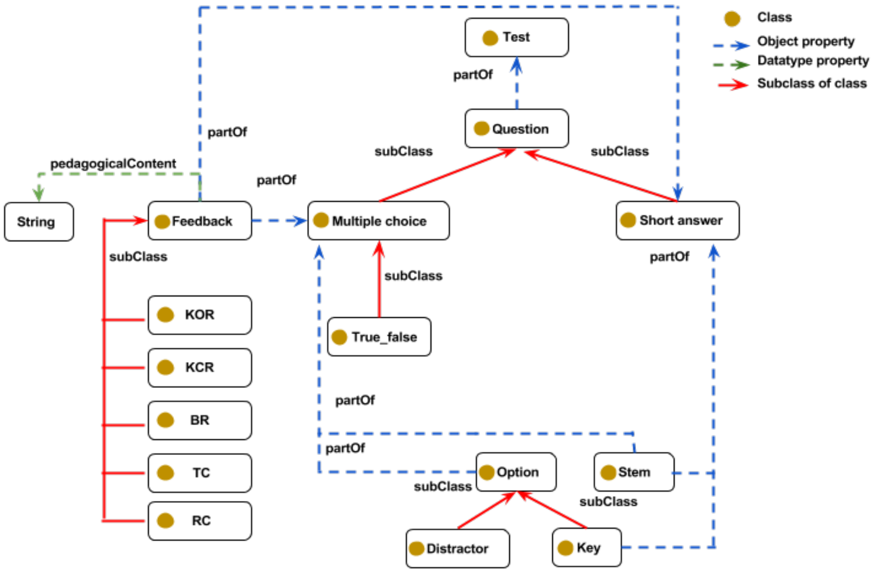


Figure 4.3: Questions and feedback represented in OntoPeFeGe domain ontology

auto-generate the question's Key and Distractors based on a predefined set of rules. For example, Strategy 2 which is a class-based strategy and is shown in Fig 4.4 auto-generates the question's Key from Class A in a domain ontology and auto-generates the Distractors from one of Class A's subclasses. The terminology and property-based strategies also consist of sub-strategies. For example, the terminology-based strategy consists of two sub-strategies, while the property-based strategy consists of thirteen sub-strategies in total, which falls into one of the following categories: Object, Datatype, or Annotation. More details about the ontology-based generation strategies are presented in Section 4.1.4.

In addition to the ontology-based generation strategies, the ontology captures the level of the question in Bloom's taxonomy. Fig 4.4 illustrates that each assessment question has a Bloom's taxonomy level (Knowledge, Comprehension, Application, Analysis) which is determined by Grubisic (Grubisic, 2012; Grubisic et al., 2013), and Cubric and Tosic's (Cubric and Tosic, 2011) stem templates (see Section 4.1.3).

At runtime each of the class, terminology and property-based strategies traverses the input domain ontology (e.g., Sakathi's Computer Networks ontology (Murugan.R et al., 2013)) to generate individuals of questions and formative feedback. Fig 4.5 illustrates an example of a multiple choice question and the different types of feedback individuals auto-generated in OntoPeFeGe.

The generated questions and the different types of formative feedback individuals

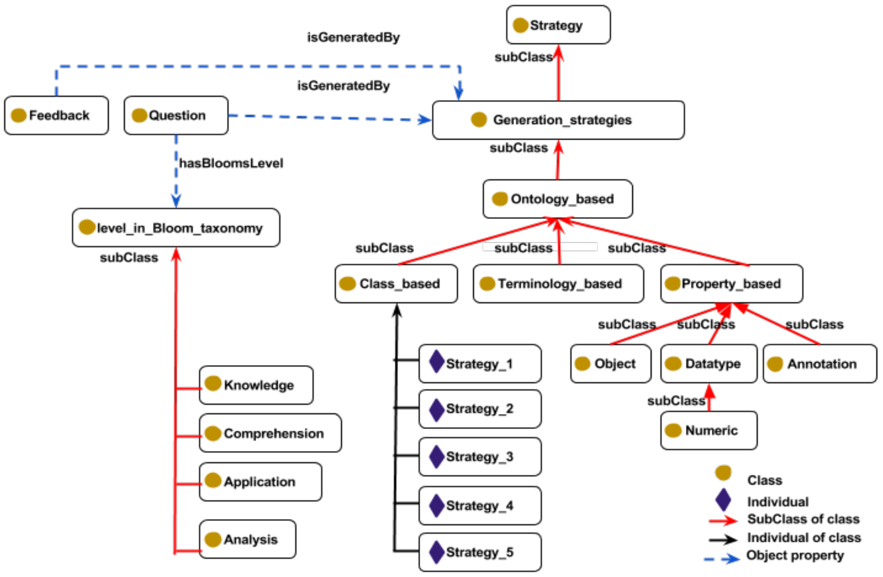


Figure 4.4: Generation strategies and Bloom's levels represented in OntoPeFeGe domain ontology

are structured using the OntoPeFeGe question and feedback domain ontology (see Appendix B for an example). This allows the questions and feedback to be mapped to Moodle XML format and then imported to Moodle VLE to run the evaluation experiments described in Chapters 5 and 6.

4.1.3 Bloom's Taxonomy Stem Templates

Providing students with personalised feedback after auto-generating different types of feedback requires information about the assessment questions characteristics. As mentioned earlier in Section 2.3.3, the feedback generators hard-coded the assessment questions and did not specify the question's characteristics (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Duboc et al., 2011; Duboc, 2013; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011). Both drawbacks hinder providing students with personalised feedback in a generic framework. To address this issue, the OntoPeFeGe integrated twenty stem templates (see Appendix C) defined by Grubisic (Grubisic, 2012; Grubisic et al., 2013), Cubric, and Tosic (Cubric and Tosic, 2011) which aimed to assess students' cognition at different levels in Bloom's taxonomy. Table 4.1 illustrates part of the stem templates for true and false questions (e.g., question 3 in Table 4.1), multiple choice questions (e.g., question 4 in Table 4.1), and short answer questions (e.g. question 8 in Table 4.1). Grubisic's stem templates aimed

Table 4.1: Part of the stem templates integrated into OntoPeFeGe

Question Number	**Stem template**	**Bloom's level**	**Type of question**	**Generation strategy**	**Literature**
1	Which of the following definitions describes the concept Class A?	Knowledge	Multiple choice	Property-based	Cubric and Tosic (Cubric and Tosic, 2011)
2	Read the paragraph and decide which one of the following concepts it defines?	Knowledge	Multiple choice	Property-based	Cubric and Tosic (Cubric and Tosic, 2011)
3	Are Class A and Class B directly connected?	Knowledge	True and false	Terminology-based	Grubisic (Grubisic et al., 2013)
4	What directly connects Class A and Class B?	Knowledge	Multiple choice	Property-based	Grubisic (Grubisic et al., 2013)
5	Which one of the following response pairs relates in the same way as: Class A Property Class B	Comprehension	Multiple choice	Property-based	Cubric and Tosic (Cubric and Tosic, 2011)
6	Are Class A and Class B indirectly connected?	Comprehension	True and false	Terminology-based	Grubisic (Grubisic et al., 2013)
7	Which one of the following demonstrates the concept Class A?	Application	Multiple choice	Class-based	Cubric and Tosic (Cubric and Tosic, 2011)
8	How many concepts is Class A connected with?	Application	Short answer	Property-based	Grubisic (Grubisic et al., 2013)
9	Analyse the following text and decide which one of the following words is a correct replacement for the blank space in the text?	Analysis	Multiple choice	Property-based	Cubric and Tosic (Cubric and Tosic, 2011)

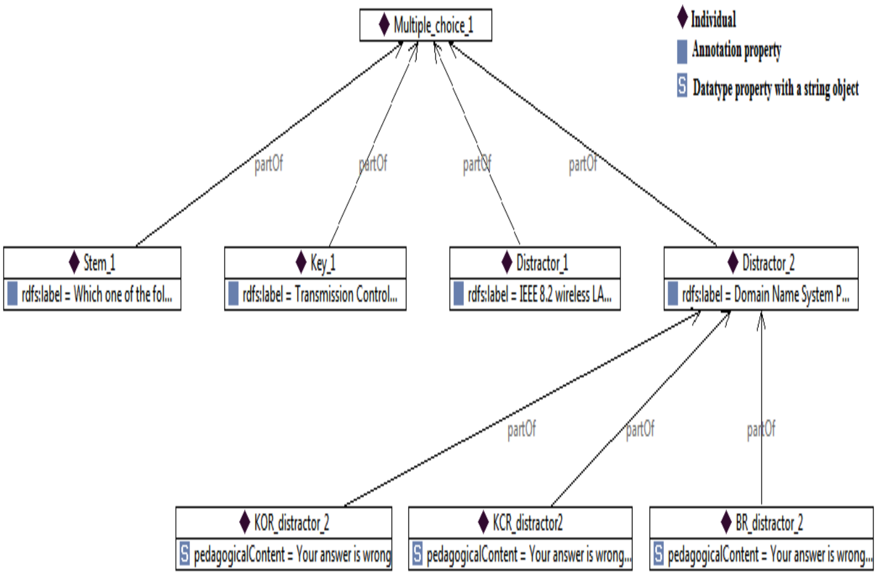

Multiple_choice_1
Individual
Annotation property
Datatype property with a string object
partOf
partOf
partOf
partOf
Stem_1
rdfs:label = Which one of the fol...
Key_1
rdfs:label = Transmission Control...
Distractor_1
rdfs:label = IEEE 8.2 wireless LA...
Distractor_2
rdfs:label = Domain Name System P...
partOf
partOf
partOf
KOR_distractor_2
pedagogicalContent = Your answer is wrong
KCR_distractor2
pedagogicalContent = Your answer is wrong...
BR_distractor_2
pedagogicalContent = Your answer is wrong...

Figure 4.5: A multiple choice question and the types of feedback individuals auto-generated in OntoPeFeGe

to assess students' cognition at the following levels in Bloom's taxonomy:

1. Knowledge level: Questions at this level focus on assessing if the students are aware of the subclasses and superclasses properties between concepts in the domain ontology. See Table 4.3 for a concrete example.

2. Comprehension level: Questions at this level focus on asking the students to identify the educational concept's subclasses and superclasses. See Table 4.4 for a concrete example.

3. Application level: Questions at this level assume that the students are more familiar with the domain ontology being tested, as students are asked to list subconcepts and superconcepts in the domain ontology.

4. Analysis level: Questions at this level focus on assessing the concept's annotation properties and the concept's datatype and object properties with other concepts in the domain ontology. See Table 4.5 for a concrete example.

Cubric and Tosic followed a different approach in forming the stem templates. They used words that define each level in Bloom's taxonomy such as demonstrate, define, relate, and analyse (CAA, 2002; Felder and Brent, 1997). See questions 1, 2, and 5 in Table 4.1.

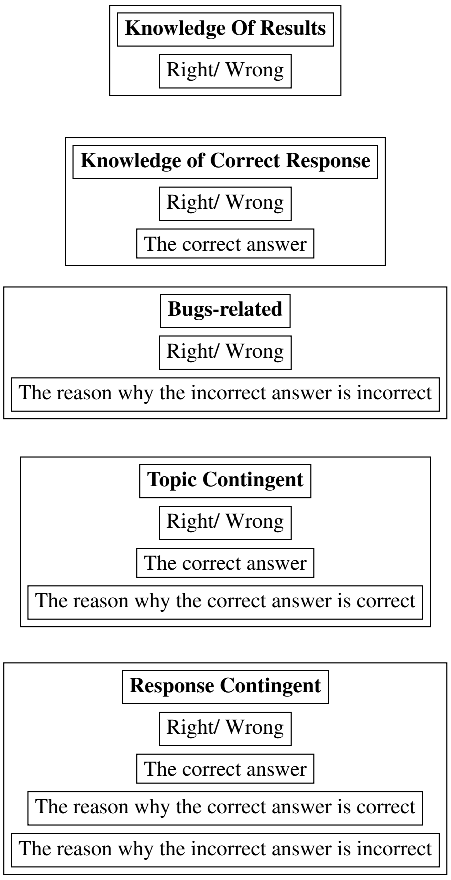

Knowledge Of Results
Right/ Wrong
Knowledge of Correct Response
Right/ Wrong
The correct answer
Bugs-related
Right/ Wrong
The reason why the incorrect answer is incorrect
Topic Contingent
Right/ Wrong
The correct answer
The reason why the correct answer is correct
Response Contingent
Right/ Wrong
The correct answer
The reason why the correct answer is correct
The reason why the incorrect answer is incorrect

Figure 4.6: Types of formative feedback and their pedagogical content

The stem templates are instantiated during the generation process by the ontology-based generation strategies defined by Papasalouros (Papasalouros et al., 2008; Papasalouros et al., 2011), Grubisic (Grubisic, 2012; Grubisic et al., 2013), Al-Yahya (Al-Yahya, 2011; Al-Yahya, 2014), Cubric, and Tosic (Cubric and Tosic, 2011). The questions generated using the stem templates are syntactically checked by a domain expert who is a teacher at the University of Manchester.

4.1.4 Generating Feedback Using a Domain Ontology

Based on the preliminary study carried out and reported in Appendix A, OntoPeFeGe supports the generation of KOR, KCR, BR, TC, and RC types of feedback which teachers often provide to students in VLEs immediately after answering an assessment question.

Fig 4.6 shows that the different types of feedback are formed from one or more of the following four pedagogical contents:

1. Right/wrong.
2. The correct answer.
3. The reason why the correct answer is correct.
4. The reason why an incorrect answer is incorrect.

The feedback pedagogical contents are auto-generated by traversing the domain ontology and filling the pedagogical content templates, which may change according to the ontology-based generation strategies (class, terminology, and property-based strategies) used during the generation process.

The *right/wrong* pedagogical content is specified in Algorithm 1 and it is used to auto-generate the KOR feedback. The algorithm does not depend on the ontology-based generation strategies. It only depends on the auto-generated question's *Key* and *Distractor* individuals. Each *Key* individual is associated with *your answer is right feedback* (line 5), and each *Distractor* individual is associated with *your answer is wrong feedback* (line 7).

Algorithm 1: Right/ wrong pedagogical content

```
1 op ← options which consist of a key and distractors;
2 K ← key;
3 KOR ← Knowledge Of Results feedback;
4 if op == K then
5     KOR=GenerateRight();
6 else
7     KOR=GenerateWrong();
```

Similarly, *the correct answer pedagogical content* does not depend on the ontology-based generation strategies. It only requires the auto-generated question's *Key*, which represent the correct answer. The *Key* could be an individual, class, or property in the domain ontology. This depends on the ontology-based generation strategy used during the generation process. For example, in a class-based strategy the question's *Key* will be an individual in the domain ontology while in a terminology-based strategy the question's *Key* will be a class in the domain ontology. The correct answer pedagogical content is generated using Algorithm 2 which uses the Key *label* (line 4).

The KCR feedback is formed by calling Algorithm 1 and Algorithm 2 for the auto-generated question's *Key* and *Distractors*.

Algorithm 2: Correct answer pedagogical content

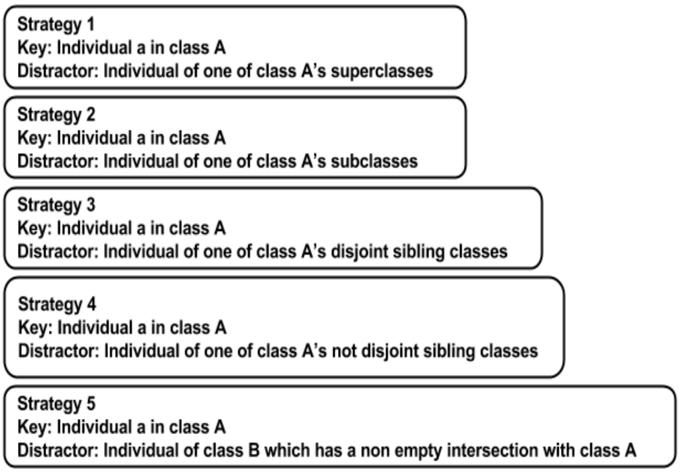
```
Function CorrectAnswer (Key)
    $CA \leftarrow$ Correct Answer;
    $CA$.append(”The correct answer is”);
    $CA$.append(key $\rightarrow$ label);
    return $CA$;
```

Neither the *right/wrong pedagogical content* nor the *correct answer pedagogical content* depend on the ontology-based generation strategies used in the generation process. Whereas *the reason why the correct answer is correct* and *the reason why an incorrect answer is incorrect* pedagogical contents depend on the ontology-based generation strategies. This means that the Bugs-Related (BR), Topic Contingent (TC), and Response Contingent (RC) feedback pedagogical content will change based on the ontology-based generation strategy used in the generation process. The following sections illustrate the algorithms used to generate BR, TC and RC types of feedback in OntoPeFeGe. The algorithms are presented according to the ontology-based generation strategies. The ontologies used in the following examples are OpenCyc (Matuszek et al., 2006) and Sakathi’s Computer network ontology (Murugan.R et al., 2013).

Class-based Strategies

The class-based strategies in the current OntoPeFeGe prototype traverse the input domain ontology to auto-generate multiple choice questions which assess students’ cognition at the application level in Bloom’s taxonomy (see question 7 in Table 4.1). The true and false and short answer stem templates defined by Grubisic (Grubisic, 2012; Grubisic et al., 2013), Cubric and Tosic (Cubric and Tosic, 2011) were not designed to use the class-based generation strategies. Instead, these questions were generated using the terminology-based and the property-based strategies (see Appendix C). The class-based strategies exploit the property between the individuals and the class in the input domain ontology (e.g., Sakathi’s Computer Networks ontology (Murugan.R et al., 2013)) to generate the question’s Key and Distractor individuals using the five class-based generation strategies shown in Fig 4.7 (Papasalouros et al., 2008; Papasalouros et al., 2011). OntoPeFeGe associates the question’s Key and Distractor individuals with different types of formative feedback which are formed from the four pedagogical contents shown in Fig 4.6.

For a concrete example, Fig 4.8 shows the Transport Layer Protocol class in the OpenCyc ontology (Matuszek et al., 2006) which has six individuals. Applying a



Strategy 1
Key: Individual a in class A
Distractor: Individual of one of class A’s superclasses

Strategy 2
Key: Individual a in class A
Distractor: Individual of one of class A’s subclasses

Strategy 3
Key: Individual a in class A
Distractor: Individual of one of class A’s disjoint sibling classes

Strategy 4
Key: Individual a in class A
Distractor: Individual of one of class A’s not disjoint sibling classes

Strategy 5
Key: Individual a in class A
Distractor: Individual of class B which has a non empty intersection with class A

Figure 4.7: Class-based strategies

class-based strategy (strategy 3 in Fig 4.7) to the ontology will generate the multiple choice question shown in Table 4.2, which assess students at the application level.

Table 4.2 shows that the question’s *Key* is the *Transmission Control Protocol* which is an individual in the *Transport Layer Protocol* class, while the *Distractors* are generated from sibling classes such as the *Domain Name System Protocol* which is an individual in the *Application Layer Protocol* class.

Algorithm 3: Reason why correct (class-based strategies)

```
Function ClassBasedReasonCorrect (Key)
    KR ← Reason why the Key option is correct;
    KR.append(”The reason why”);
    KR.append(Key → label);
    KR.append(”is the correct answer is due to the following:”);
    KR.append(Key → label);
    KR.append(Key → class);
    return KR;
```

Algorithm 4: Reason why incorrect (Class-based strategies)

```
Function ClassBasedReasonIncorrect (Distractor)
    DI ← Reason why the distractor option is incorrect;
    DI.append(”The reason why”);
    DI.append(Distractor → label);
    DI.append(”is the incorrect answer is due to the following:”);
    DI.append(Distractor → label);
    DI.append(Distractor → class);
    return DI;
```

When a student chooses the *Domain Name System Protocol* as an answer, he or she will be provided with the auto-generated formative feedback shown in Table 4.2. For

Table 4.2: Question and feedback generated using a class-based strategy

Ontology-based generation strategy	**Class-based generation strategy (Strategy 3)**
Stem template	Which one of the following demonstrates the concept *Class A*?
Stem individual	Which one of the following demonstrates the concept **Transport Layer Protocol**?
Key	Transmission Control Protocol
Distractors	IEEE 8.2 wireless LAN protocol Domain Name System Protocol Internet Protocol
Generated feedback pedagogical content when a student selects the **Domain Name System Protocol**.	1. Your answer is wrong. 2. The correct answer is **Transmission Control Protocol**. 3. The reason why **Transmission Control Protocol** is the correct answer is due to the following: Transmission Control Protocol is a Transport Layer Protocol. 4. The reason why **Domain Name System Protocol** is the incorrect answer is due to the following:**Domain Name System Protocol** is an **Application Layer Protocol**.

example, in Table 4.2 the feedback pedagogical contents *your answer is wrong* and *the correct answer is Transmission Control Protocol* are generated using Algorithm 1 and Algorithm 2 respectively. OntoPeFeGe also auto-generates *the reason why the correct answer is correct* and *the reason why the incorrect answer is incorrect* pedagogical contents using Algorithm 3 and Algorithm 4 respectively. Algorithm 3 takes the question's Key individual (line 1) as a parameter (e.g., *Transmission Control Protocol*) and provides students with the ontology class (e.g., *Transport Layer Protocol*) which the *Key* individual belongs to (line 7). See *the reason why the correct answer is correct* pedagogical content auto-generated for the question example in Table 4.2.

On the other hand, Algorithm 4 takes the question's *Distractor* individual as a parameter (e.g., *Domain Name System Protocol*) and provides students with information about the *Distractor* class in which the individual they selected belongs to (e.g., *Application Layer Protocol*). See *the reason why the incorrect answer is incorrect* pedagogical content auto-generated for the question example in Table 4.2.

Terminology-based Strategies

The two terminology-based generation strategies (Strategy 6 and Strategy 7) shown in Fig 4.9 are used in the current OntoPeFeGe prototype to generate true and false questions which assess students' cognition at the knowledge, comprehension, and application levels in Bloom's taxonomy (see Appendix C). The terminology-based strategies exploit the *subClass* property which relates the subject resource to the object resource

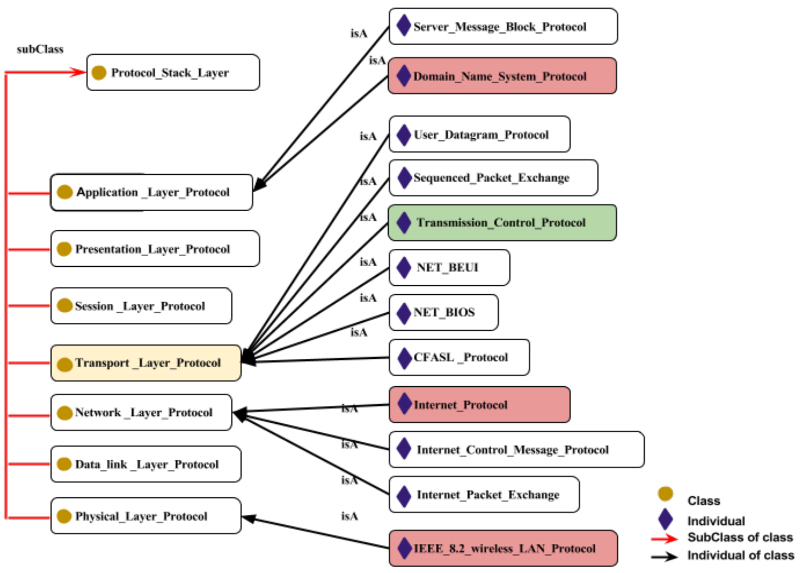

subClass
Protocol_Stack_Layer
Application _Layer_Protocol
Presentation_Layer_Protocol
Session _Layer_Protocol
Transport _Layer_Protocol
Network _Layer_Protocol
Data_link _Layer_Protocol
Physical_Layer_Protocol
isA
Server_Message_Block_Protocol
Domain_Name_System_Protocol
User_Datagram_Protocol
Sequenced_Packet_Exchange
Transmission_Control_Protocol
NET_BEUI
NET_BIOS
CFASL _Protocol
Internet_Protocol
Internet_Control_Message_Protocol
Internet_Packet_Exchange
IEEE_8.2_wireless_LAN_Protocol
Class
Individual
SubClass of class
Individual of class

Figure 4.8: Transport Layer Protocol class and individuals in OpenCyc ontology

Strategy 6
Key: subclass of class A
Distractor: Class A's sibling classes to replace object

Strategy 7
Key: subclass of class A
Distractor: Class A's sibling classes to replace subject

Figure 4.9: Terminology-based strategies

in the domain ontology as follows:

Subject subClass Object

The subject is a class in the domain ontology (e.g., *Transport Layer* class shown in Fig 4.10) and the object could be either a class or a restriction (a restriction in OWL is a class defined by describing the individuals it contains (Allemang and Hendler, 2011)) such as the *transmits only frames* and the *transmits only datagrams* restriction classes shown in Fig 4.10.

OntoPeFeGe auto-generates *the reason why the correct answer is correct* pedagogical content for the true and false questions using Algorithm 5. The algorithm uses the question's *Key*, and the subject parameters (line 1). The subject of the *subClass* property is used as a parameter because the *Key* in true and false questions is either a *yes* or *no* individual. Algorithm 5 retrieves the superclasses for the subject to help the

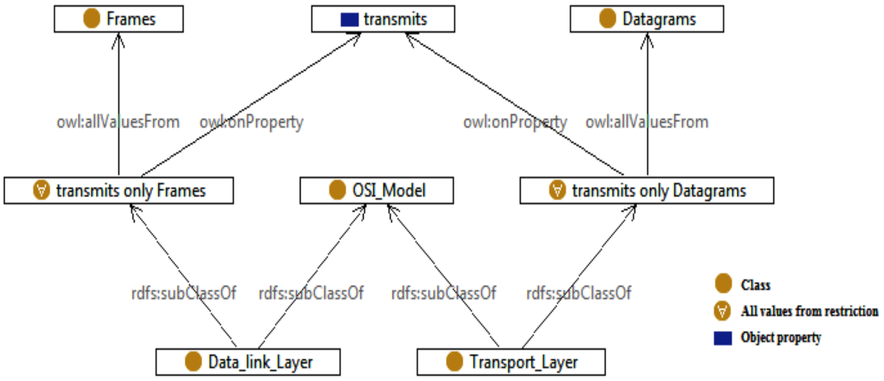


Figure 4.10: Transport Layer and Data link Layer concepts in Sakathi's ontology (Murugan.R et al., 2013)

student relate the subject to the correct object (line 6). For each superclass (Object) the algorithm checks if it is a class (line 7) or a restriction. If the superclass is a class, then the algorithm retrieves the superclass label (line 10). On the other hand, if the superclass is a restriction (line 11) the algorithm retrieves the type of the restriction (line 14) which could be *owl:allValuesFrom*, *owl:someValuesFrom*, or *owl:hasValue* (see Section 2.3.1 in Chapter 2, page 25), and then retrieves the property label (line 15) and the class label (line 16) which the restriction is applied on.

Algorithm 5: Reason why correct (Terminology-based strategies)

```
1  Function TerminologyBasedReasonCorrect (Key, Subject)
2      KR ← Reason why the Key option is correct;
3      KR.append(”The reason why”);
4      KR.append(Key → label);
5      KR.append (is the correct answer is due to the following:);
6      foreach class ε Subject.listSuperclasses do
7          if class.isRestriction() == false then
8              KR.append(Subject → label);
9              KR.append(”is”);
10             KR.append(class → label);
11         else
12             KR.append(Subject → label);
13             Restriction = class → asRestriction();
14             Type = Restriction → type ;
15             KR.append(Type → getPropertyLabel);
16             KR.append(Type → getValuesFromLabel);
17     return KR;
```

For example, Table 4.3 shows a true and false knowledge level question auto-generated using the terminology-based strategy 6 shown in Fig 4.9. The question is auto-generated after traversing the domain ontology shown in Fig 4.10. The ontology shows that the *Transport Layer* is a subclass of *transmits only datagrams* restriction class, and the *Data link Layer* is a subclass of *transmits only frames* restriction class. The question is auto-generated by replacing the object in the following statement from *transmits only datagrams* to *transmits only frames*:

$$\overbrace{`Transport\ \ Layer'}^{Subject}\ \overbrace{subclass}^{Property}\ \overbrace{`transmits\ \ only\ \ datagram'}^{Object}$$

Table 4.3 also shows an example of *the reason why the correct answer is correct* pedagogical content, which explained to students that the *Transport layer* transmits datagrams and not frames. OntoPeFeGe also auto-generates *the reason why the incorrect answer is incorrect* pedagogical content using Algorithm 6. The algorithm uses the question's *Distractor* and the object parameter (line 1) which is used to retrieve the object subclasses (line 6). To auto-generate the pedagogical content the algorithm uses the subclass label (line 7) and checks if the object parameter is a class (line 8) or a restriction. The object in the example shown in Table 4.3 is *transmits only frames*, which is a restriction and the subclass of the object is the *Data link Layer* (see Fig 4.10). The pedagogical content is auto-generated to explain to students that the *Data link Layer* transmits frames.

Algorithm 6: Reason why incorrect (Terminology-based strategies)

```
1  Function TerminologyBasedReasonIncorrect (Distractor, Object)
2      DI ← Reason why the distractor option is incorrect;
3      DI.append(The reason why);
4      DI.append(Distractor → label);
5      DI.append(is the incorrect answer is due to the following:);
6      foreach class ε Object.listSubClasses do
7          DI.append(class → label);
8          if Object.isRestriction() == false then
9              DI.append(Object → label);
10         else
11             Restriction = Object → asRestriction();
12             Type = Restriction → type ;
13             DI.append(Type → getPropertyLabel);
14             DI.append(Type → getValuesFromLabel);
15     return DI;
```

Property-based Strategies

The property-based generation strategies are used to generate true and false, multiple choice, and short answer questions from the domain ontologies. The questions generated using the property-based strategies assess the students' cognition at the knowledge, comprehension, application and analysis levels in Bloom's taxonomy. See Appendix C.

Table 4.3: Question and feedback generated using a terminology-based strategy

Ontology-based generation strategy	**Terminology-based generation strategy (Strategy 6)**	
Stem template	Is *Class A* subclass of *Class B*?	
Stem individual	Is Transport layer transmits frames?	
Key	No	Transport layer transmits datagrams
Distractors	Yes	Transport layer transmits frames
Generated feedback pedagogical content when a student selects the Yes option.	1. Your answer is wrong. 2. The correct answer is No. 3. The reason why No is the correct answer is due to the following: **Transport layer transmits datagrams**. 4. The reason why Yes is the incorrect answer is due to the following: **Data link layer transmits frames**.	

The property-based strategies are categorised into:

1. Object-based strategies, which exploit the object properties in the domain ontology. Object properties are used to connect two resources together where the subject resource and the object resource are classes in the domain ontology (see Section 2.3.1).

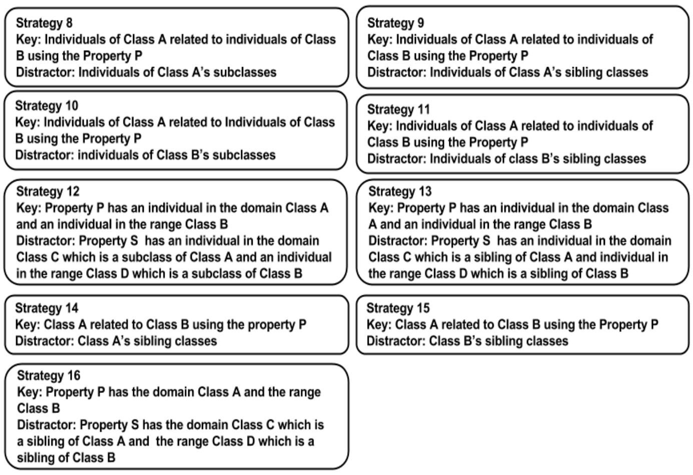
2. Datatype-based strategies, which exploit the datatype properties in the domain ontology. Datatype properties are used to connect a resource to an RDFS:Literal or to an XML schema built-in datatype value (W3C, 2012).

3. Annotation-based strategies, which exploit the *rdfs:comment* (a property which provides human readable descriptions to concepts in the domain ontology), and the *rdfs:label* (a *property* which is used to provide a name for the class or the property in the domain ontology) properties.

The following sections explain how *the reason why the correct answer is correct* and *the reason why an incorrect answer is incorrect* templates are generated using the object, datatype, and annotation-based strategies.

1) Object-based Strategies

The object-based strategies are used to auto-generate true and false, multiple choice, and short answer questions which assess students on the knowledge, comprehension, application and analysis levels in Bloom's taxonomy. Fig 4.11 shows the nine object-based generation strategies which are used in the current OntoPeFeGe prototype to generate questions and feedback. The OntoPeFeGe auto-generates *the reason why the correct answer is correct* pedagogical content using Algorithm 7. The algorithm uses



Strategy 8
Key: Individuals of Class A related to individuals of Class B using the Property P
Distractor: Individuals of Class A’s subclasses

Strategy 9
Key: Individuals of Class A related to individuals of Class B using the Property P
Distractor: Individuals of Class A’s sibling classes

Strategy 10
Key: Individuals of Class A related to Individuals of Class B using the Property P
Distractor: individuals of Class B’s subclasses

Strategy 11
Key: Individuals of Class A related to individuals of Class B using the Property P
Distractor: Individuals of class B’s sibling classes

Strategy 12
Key: Property P has an individual in the domain Class A and an individual in the range Class B
Distractor: Property S has an individual in the domain Class C which is a subclass of Class A and an individual in the range Class D which is a subclass of Class B

Strategy 13
Key: Property P has an individual in the domain Class A and an individual in the range Class B
Distractor: Property S has an individual in the domain Class C which is a sibling of Class A and individual in the range Class D which is a sibling of Class B

Strategy 14
Key: Class A related to Class B using the property P
Distractor: Class A’s sibling classes

Strategy 15
Key: Class A related to Class B using the Property P
Distractor: Class B’s sibling classes

Strategy 16
Key: Property P has the domain Class A and the range Class B
Distractor: Property S has the domain Class C which is a sibling of Class A and the range Class D which is a sibling of Class B

Figure 4.11: Object-based strategies

the question’s *Key*, which could be an individual, class, property, or *yes/no* (if a true and false question is generated). The algorithm also takes the *Key* object property, the *Key* subject, and the *Key* object parameters to capture the statement associated with the correct answer (subject property object). OntoPeFeGe also auto-generates *the reason why the incorrect answer is incorrect* using Algorithm 8 which takes the *Distractor* object property, the *Distractor* subject, and the *Distractor* object parameters which capture the statement associated with the incorrect answer (subject property object). *The reason why the correct answer is correct* pedagogical content auto-generated using the object-based strategies provides students with the statement associated with the correct answer, while *the reason why the incorrect answer is incorrect* pedagogical content provides students with the statement associated with the incorrect answer.

Table 4.4 shows an example of a comprehension level question auto-generated using the object-based strategy 14 shown in Fig 4.11. The table shows *the reason why the correct answer is correct* pedagogical which explains to the student that the Connection Control (*Key object*) is a function (*Key object property*) of the Transport Layer (*Key subject*). While *the reason why the incorrect answer is incorrect* explains to the student that the Logical Addressing (*Distractor object*) is a function (*Distractor object property*) of the Network Layer (*Distractor subject*).

2) Datatype-based Strategies

True and false, multiple choice, and short answer questions are generated by exploiting the datatype properties in the domain ontology.

Table 4.4: Question and feedback generated using an object-based strategy

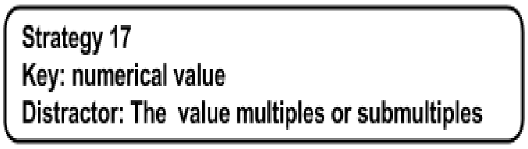
Ontology-based generation strategy	**Property-based generation strategy (Strategy 14)**
Stem template	Which superclass is directly connected by *Property* with *Class A*?
Stem individual	Which one of the following is a function of the **Transport Layer**?
Key	Connection Control
Distractors	Synchronisation Logical Addressing Physical Addressing
Generated feedback pedagogical content when a student selects the **Logical Addressing**.	1. Your answer is wrong. 2. The correct answer is **Connection Control**. 3. The reason why **Connection Control** is the correct answer is due to the following: **Transport Layer functions Connection Control**. 4. The reason why **Logical Addressing** is the incorrect answer is due to the following: **Network Layer functions Logical Addressing**.

Algorithm 7: Reason why correct (Object-based strategies)

```
Function ObjectPropertyBasedReasonCorrect (Key, keyObjProperty, keySubject, keyObject)
    KR ← Reason why the Key option is correct;
    KR.append(”The reason why”);
    KR.append(Key → label);
    KR.append(”is the correct answer is due to the following:”);
    KR.append(keySubject → label);
    KR.append(keyObjProperty → label);
    KR.append(keyObject → label);
    return KR;
```

Algorithm 8: Reason why incorrect (Object-based strategies)

```
Function ObjectPropertyBasedReasonIncorrect (Distractor, distractorObjProperty, distractorSubject,distractorObject)
    DI ← Reason why the distractor option is incorrect;
    DI.append(”The reason why”);
    DI.append(Distractor → label);
    DI.append(”is the incorrect answer is due to the following:”);
    DI.append(distractorSubject → label);
    DI.append(distractorObjProperty → label);
    DI.append(distractorObject → label);
    return DI;
```



Strategy 17
Key: numerical value
Distractor: The value multiples or submultiples

Figure 4.12: Datatype-based strategy

Fig 4.12 shows strategy 17 (Grubisic, 2012), which generates the question's *Key* and the question's *Distractors*. The *Key* is the object of the datatype property and it is a numerical value while the *Distractors* are the multiples or submultiples of the numerical value. OntoPeFeGe uses Algorithm 9 to auto-generate *the reason why the correct answer is correct* pedagogical content. The algorithm takes the question's *Key* and the datatype property associated with the *Key* (keyDatatypeProperty) as parameters (line 1) and then retrieves the object of the key datatype property (line 9). Table 4.5 illustrates a true and false analysis level question auto-generated using Strategy 17. The question aims to assess if the students know the *number of layers* in the Transmission Control Protocol/ Internet Protocol model (*TCP/IP model*). *TCP/IP model* is a class in the domain ontology which has the *number of layers* datatype property. Table 4.5 shows that *the reason why the correct answer is correct* pedagogical content explained to students that the *number of layers* in the *TCP/IP* model is *4*.

Algorithm 9: Reason why correct (Datatype-based strategies)

```
1  Function DatatypePropertyBasedReasonCorrect (Key, keyDatatypeProperty)
2      KR ← Reason why the Key option is correct;
3      KR.append(”The reason why”);
4      KR.append(Key → label);
5      KR.append(”is the correct answer is due to the following:”);
6      KR.append(keyDatatypeProperty → label);
7      KR.append(”of”);
8      KR.append(Key → label;
9      KR.append(”is”);
10     object = keyDatatypeProperty → Object;
11     KR.append(object → label);
12     return KR;
```

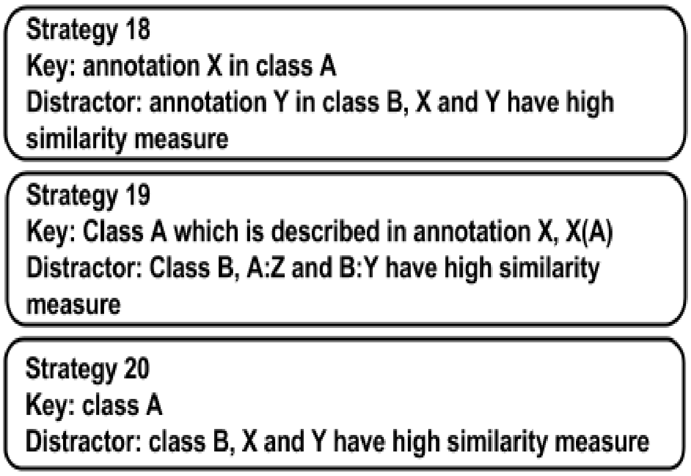
In addition to *the reason why the correct answer is correct* pedagogical content, OntoPeFeGe auto-generates *the reason why the incorrect answer is incorrect* using Algorithm 10. The algorithm uses the question's *Distractor* (line 1). It starts by providing students with information about their selected answer (line 4), and then explains that the selected answer is double, triple, or quadruple the correct answer (line 8). After that, the algorithm provides the students with more details about the correct answer.

Algorithm 10: Reason why incorrect (Datatype-based strategies)

```
Function ObjectPropertyBasedReasonIncorrect (Distractor, Key, keyDatatypeProperty)
    DI ← Reason why the distractor option is incorrect;
    DI.append(”The reason why”);
    DI.append(Distractor → label);
    DI.append(”is the incorrect answer is due to the following:”);
    DI.append(Distractor → label);
    DI.append(”is”);
    DI.append(multiplierValue → label);
    DI.append(keyDatatypeProperty → label);
    DI.append(Key → label);
    DI.append(”and”);
    DI.append(keyDatatypeProperty → label);
    DI.append(”of”);
    DI.append(Key → label);
    object = keyDatatypeProperty → Object;
    DI.append(object → label);
    return DI;
```

Table 4.5: Question and feedback generated using a datatype-based strategy

Ontology-based generation strategy	**Property-based generation strategy (Strategy 17)**
Stem template	Is *Property* of *Subject Object*?
Stem individual	Is number of layers of TCP/IP model 8?
Key	No
Distractors	Yes
Generated feedback pedagogical content when a student selects the **Yes** option.	1. Your answer is wrong. 2. The correct answer is **4**. 3. The reason why **4** is the correct answer is due to the following: The number of layers of TCP/IP model is 4. 4. The reason why **8** is the incorrect answer is due to the following: **8** is double the number of layers of TCP/IP model and number of layers of TCP/IP model is 4.



Strategy 18
Key: annotation X in class A
Distractor: annotation Y in class B, X and Y have high similarity measure

Strategy 19
Key: Class A which is described in annotation X, X(A)
Distractor: Class B, A:Z and B:Y have high similarity measure

Strategy 20
Key: class A
Distractor: class B, X and Y have high similarity measure

Figure 4.13: Annotation-based strategies

Table 4.5 shows an example of *the reason why the incorrect answer is incorrect* pedagogical content which is auto-generated in OntoPeFeGe. The pedagogical content explained to students that 8 is double the *number of layers* in the TCP/IP model. It also provided the students with information about the number of layers in the TCP/IP model.

3) Annotation-based Strategies

The annotation-based strategies exploit the rdfs:comment and the rdfs:label associated with the ontology classes and individuals in the domain ontology. Fig 4.13 shows the annotation-based strategies, which were used to generate the multiple choice questions illustrated in Table 4.1 and Appendix C. The true and false and short answer stem templates defined by Grubisic (Grubisic, 2012; Grubisic et al., 2013), and Cubric and Tosic (Cubric and Tosic, 2011) were not designed to use the annotation-based strategies. Instead, they focused on assessing the students on the object properties in the educational domain.

OntoPeFeGe auto-generates *the reason why the correct answer is correct* pedagogical content using Algorithm 11. The Algorithm takes the following parameters: the *Key* in the auto-generated question, the name of the annotation-based strategy (e.g., Strategy 18), and the ontology class having the annotation property used to auto-generate the question's *Key (ClassAnnot)*. The *ClassAnnot* parameter is used when questions are generated using strategy 19 (see Fig 4.13). Strategy 19 shows that the question's *Key* is a class in the domain ontology, which is described in the annotation property of another class in the same domain ontology.

The algorithm shows that the annotation-based strategies auto-generate different pedagogical contents for *the reason why the correct answer is correct*. When strategy 18 (Cubric and Tosic, 2011) is used in the generation process, students are provided with questions to assess if they could provide a definition of the educational concepts (class or individual) in the domain ontology (see question 1 in Table 4.1). The options (*Key* and *Distractors*) in the auto-generated question are definitions retrieved from several classes or individuals in the domain ontology. OntoPeFeGe auto-generates *the reason why the correct answer is correct* pedagogical content to provide the students with the *Key* class (the correct educational concept) which the definition belongs to (line 9).

Algorithm 11: Reason why correct (Annotation-based strategies)

```
1  Function AnnotationReasonCorrect (Key , strategyName, concept, ClassAnnot)
2      KR ← Reason why the Key option is correct;
3      if strategyName == Strategy18 then
4          KR.append(”The reason why”);
5          KR.append(Key → comment);
6          KR.append(”is the correct answer is due to the following:”);
7          KR.append(Key → comment);
8          KR.append(”is the definition for”);
9          KR.append(Key → label);
10     else if strategyName == Strategy19 then
11         KR.append(”The reason why”);
12         KR.append(Key → label);
13         KR.append(”is the correct answer is due to the following:”);
14         KR.append(ClassAnnot → comment);
15     else if strategyName == Strategy20 then
16         KR.append(”The reason why”);
17         KR.append(Key → label);
18         KR.append(”is the correct answer is due to the following:”);
19         KR.append(Key → label);
20         KR.append(”is defined as”);
21         KR.append(Key → comment);
22     return KR;
```

On the other hand, when strategy 19 (Cubric and Tosic, 2011) is used to auto-generate the multiple choice questions, the question's *Key* is auto-generated from an ontology class having an annotation property containing the *Key*. Therefore, the pedagogical content is auto-generated by querying the class annotation property (line 14). For example, Table 4.6 illustrates an analysis level question generated using strategy 19. The question's *Key* is the *Application layer protocol*, which is contained in the *Presentation Layer Protocol* annotation property (rdfs:comment). The table shows *the reason why the correct answer is correct* pedagogical content, which provides the students with the *rdfs:comment* of the *Presentation Layer Protocol*.

In addition to strategies 18 and 19, strategy 20 (Cubric and Tosic, 2011) is used to

Algorithm 12: Reason why incorrect (Annotation-based strategies)

```
Function AnnotationReasonIncorrect (Distractor, strategyName)
    DI ← Reason why the distractor option is incorrect;
    if strategyName == Strategy18 then
        DI.append(”The reason why”);
        DI.append(Distractor → comment);
        DI.append(”is the incorrect answer is due to the following:”);
        DI.append(Distractor → comment);
        DI.append(”is the definition for”);
        DI.append(Distractor → label);
    else if strategyName == Strategy19 then
        DI.append(”The reason why”);
        DI.append(Distractor → label);
        DI.append(”is the incorrect answer is due to the following:”);
        DI.append(Distractor → comment);
    else if strategyName == Strategy20 then
        DI.append(”The reason why”);
        DI.append(Distractor → label);
        DI.append(”is the incorrect answer is due to the following:”);
        DI.append(Distractor → label);
        DI.append(”is defined as”);
        DI.append(Distractor → comment);
    return KR;
```

Table 4.6: Question and feedback generated using an annotation-based strategy

Ontology-based generation strategy	**Property-based generation strategy (Strategy 19)**
Stem template	Analyse the following text and decide which one of the following words is a correct replacement for the blank space in the text: *Note: the text is Class B’s annotation property (comment) and the blank space shown below is Class A, which is contained in the comment.*
Stem individual	Analyse the following text and decide which one of the following words is a correct replacement for the blank space in the text: ‘A presentation layer protocol takes the responsibility for routine tasks from an ——, such as converting between character sets.’
Key	Application Layer Protocol
Distractors	Presentation Layer Protocol Transport Layer Protocol Session Layer Protocol
Generated feedback pedagogical content when a student selects the **Session Layer Protocol** option.	1. Your answer is wrong. 2. The correct answer is **Application Layer Protocol**. 3. The reason why **Application Layer Protocol** is the correct answer is due to the following: ‘A presentation layer protocol takes the responsibility for routine tasks from an **Application Layer Protocol**, such as converting between character sets’. 4. The reason why **Session Layer Protocol** is the incorrect answer is due to the following: **Session Layer Protocol** allows sessions to be established between two machines. A session facilitates processes that involve intensive data transfer between two computers, such as transferring a large file.

auto-generate questions which assess if the students could relate a specific definition to a concept in the domain ontology. Algorithm 11 shows that *the reason why the correct answer is correct* pedagogical content is generated to provide the student with the correct definition that is related to the question's *Key* (see line 22 in Algorithm 11).

OntoPeFeGe also auto-generates *the reason why the incorrect answer is incorrect* using Algorithm 12. The generation process is similar to Algorithm 11. However, instead of using the *Key* parameter the function used the *Distractor* parameter. For example, Table 4.6 shows *the reason why the incorrect answer is incorrect* pedagogical content auto-generated in OntoPeFeGe when strategy 19 is used. The table shows that when a student chose the *Session Layer Protocol* he or she was provided with the annotation property associated with the chosen *Distractor* (*Session Layer Protocol*).

4.2 Personalised Feedback Algorithm for the Moodle VLE

The previous section introduced a feedback generator, which auto-generates KOR, KCR, BR, TC, and RC types of feedback from a domain ontology. The generator associated the different types of feedback with questions aimed to assess the students at different levels in Bloom's taxonomy. Both the generated questions and the different types of feedback were structured using the OntoPeFeGe questions and feedback ontology, and then mapped to Moodle XML format to allow teachers import the questions and feedback into Moodle VLE. This section explains the personalised feedback algorithm developed in the Moodle VLE to provide the appropriate type of formative feedback to the students immediately after answering an assessment question. The algorithm adopts and implements the theoretical personalised feedback framework proposed by Mason and Bruning (Mason and Bruning, 2001) for the following reasons:

1. Recent research studies considered the task's difficulty as an important factor in the process of personalising feedback in learning environments (Narciss et al., 2014). However, the personalised feedback frameworks illustrated in Section 2.2 focused on providing students with personalised feedback based on the student's characteristics and ignored the task's characteristics except for Mason and Bruning's framework which considered the student and the task's characteristics.

2. The preliminary study (Study 2) carried out in Appendix A to investigate the types of feedback teachers provide to students after answering an assessment

question revealed that teachers highly used the BR, TC, and RC types of feedback. These types of feedback were only supported by Mason and Bruning's theoretical personalised feedback framework and the Adaptive feedback framework (AFF) (Narciss et al., 2014) (see Table 2.1 in Chapter 2, page 17). However, the AFF focused on providing the students with personalised feedback based on the student's characteristics and ignored the task's characteristics. Therefore, Mason and Bruning's framework is adopted in this thesis.

3. The framework is theoretical and no empirical evaluation has yet been carried out to investigate the relationship between the student's characteristics (background knowledge, gender), the task's characteristics (level of the question in Bloom's taxonomy (Bloom et al., 1956)) and the personalised feedback, and how they affect the students' performance and learning gain. Therefore, this thesis will evaluate the framework for the first time in Chapter 6.

The activity diagram of the personalised feedback algorithm is illustrated in Fig 4.14. The algorithm starts by fetching the first question in a test. Students with low background knowledge receive Response Contingent feedback regardless of the correctness of their answer or the level of the question in Bloom's taxonomy. On the other hand, students with high background knowledge are provided with different types of feedback based on their current level of knowledge and the level of the question in Bloom's taxonomy. Students who answer the knowledge level questions correctly are provided with Bugs-Related feedback, and the students who answer the knowledge level questions incorrectly are provided with Topic Contingent feedback. Fig 4.14 also shows that high background knowledge students are provided with Topic Contingent feedback after answering comprehension, application and analysis level questions regardless of the correctness of their answer.

4.3 Chapter Summary

This chapter presents the design and implementation of the Ontology-based Personalised Feedback Generator (OntoPeFeGe) system prototype, which consists of the feedback generator and the personalised feedback algorithm. The generator exploits the OWL domain ontology to auto-generate assessment questions and associates them with a range of different types of feedback. The personalised feedback algorithm provides students with the appropriate type of feedback immediately after answering an assessment question. The OntoPeFeGe system prototype is novel in being the first to:

1. Integrate the ontology-based generation strategies defined by Papasalouros (Papasalouros et al., 2008; Papasalouros et al., 2011), Grubisic (Grubisic, 2012; Grubisic et al., 2013), Al-Yahya (Al-Yahya, 2011; Al-Yahya, 2014), Cubric, and Tosic (Cubric and Tosic, 2011) into one system. The ontology-based generation strategies traverse the domain ontology to instantiate a set of stem templates which are designed to assess student's cognition at different levels in Bloom's taxonomy. Integrating the different ontology-based generation strategies and the stem templates into OntoPeFeGe allowed achieving Contribution 4 in this thesis where the quality of tests and questions auto-generated is quantitatively analysed for the first time (see Chapter 5 for more details).

2. Auto-generate five types of formative feedback (KOR, KCR, BR, TC, RC) from domain ontologies. These types of feedback were either neglected (Kazi *et al.* (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013) focused on auto-generating hint feedback) or partially supported (S$\tilde{a}$nchez-Vera *et al.* (Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011) focused on auto-generating KOR and KCR feedback) by the feedback generators introduced in Section 2.3.3.

3. Auto-generate domain independent feedback where no expert knowledge base or human intervention (teachers or domain experts) is needed.

4. Implement Mason and Bruning's theoretical framework in Moodle VLE. This contributes to the field of personalised feedback frameworks and the ontology-based formative feedback generators by: firstly, investigating the relationship between student's characteristics, the task's characteristics, and the personalised feedback, and how they affect students' performance and learning gain (see Chapter 6); secondly, OntoPeFeGe is the first prototype to provide the students with personalised feedback auto-generated from a domain ontology.

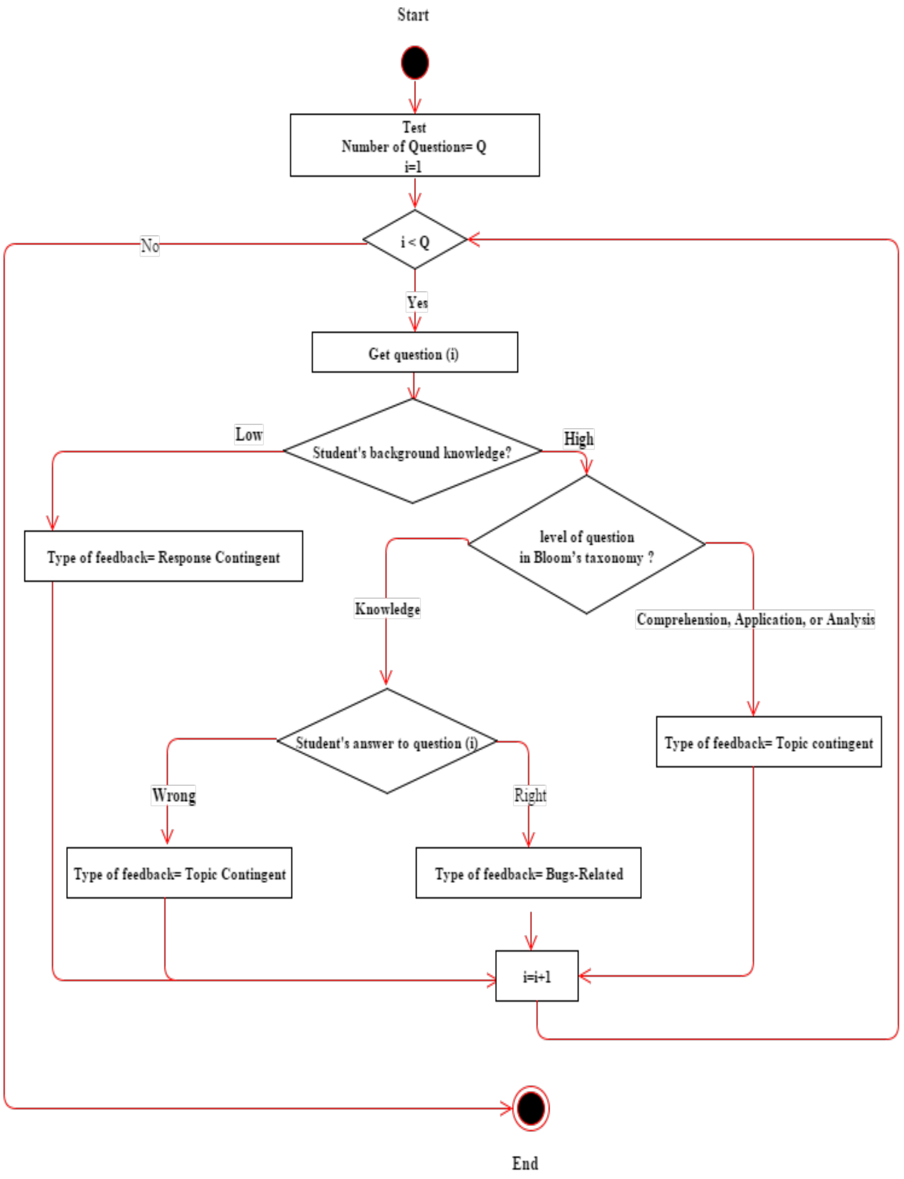

Start
Test
Number of Questions= Q
i=1
i < Q
No
Yes
Get question (i)
Low
Student's background knowledge?
High
Type of feedback= Response Contingent
level of question
in Bloom's taxonomy ?
Knowledge
Comprehension, Application, or Analysis
Student's answer to question (i)
Type of feedback= Topic contingent
Wrong
Right
Type of feedback= Topic Contingent
Type of feedback= Bugs-Related
i=i+1
End

Figure 4.14: Activity diagram of the personalised feedback algorithm

Chapter 5

Analysing the Quality of Auto-generated Tests and Questions

Tests are formed from auto-generated questions, and the quality of tests and questions is determined by their difficulty, discrimination, and reliability measurements. Several ontology-based question generators used ontologies to auto-generate questions, which aimed to assess students' cognition at different levels in Bloom's taxonomy (see Section 2.3.3). However, the evaluation of the questions was confined to measuring the qualitative satisfaction of domain experts and students (Papasalouros et al., 2008; Papasalouros et al., 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014). The domain experts assessed if the questions were syntactically correct and could be used in an assessment test, while the students assessed if the auto-generated questions were comprehensible. None of the question generators tested the questions on students and analysed the difficulty, discrimination, and reliability of the auto-generated questions and tests. The lack of quantitative analysis resulted in having no evidence on the quality of questions, and how the quality is affected by the ontology-based generation strategies, the level of the question in Bloom's taxonomy (determined by the stem templates which are the text stating the question), and the type of question used by the different question generators. In addition, the domain experts and students' opinions are subjective. For example, the domain experts may agree that the question could be used in an assessment test, however, there is no evidence that the question could discriminate between high ability and low ability students.

This chapter presents the experiment carried out to address the drawbacks mentioned above and achieve Contribution 4 by assessing the following for the first time:

1. The auto-generated questions' and tests' difficulty, discrimination, and reliability

using the Classical Test Theory (CTT). However, the CTT measurements depend on the sample of students who take the tests (Alagumalai and Curtis, 2005; Ding and Beichner, 2009; Doran, 1980; Cohen et al., 2013; Erguven, 2014; De Ayala, 2009; Haladyna, 1994). Therefore, in this chapter the questions' quality is also assessed using the Item Response Theory (IRT) which assumes that the questions' difficulty and discrimination are sample independent and does not change across different groups of students (De Ayala, 2009; Hambleton and Swaminathan, 1985). See Section 5.2 for more details on the CTT and IRT.

2. The effect of the ontology-based generation strategy, the level of the questions in Bloom's taxonomy, and the type of questions on the quality of questions obtained using the CTT and the IRT. This will provide question design guidance for developers and researchers working in the field of question generators.

The chapter is organised into the following sections. Section 5.1 presents an overview of the experiment. Section 5.2 explains the evaluation methods. Section 5.3 illustrates the experiment results and discussion, and Section 5.4 concludes the chapter.

5.1 Experiment Overview

This section illustrates the experiment questions, the experiment participants, and describes the experimental set-up used to evaluate the quality of tests and questions.

5.1.1 Experiment Questions

The experiment was designed to achieve Contribution 4 by answering the following questions:

Q1: Do the questions and tests generated from domain ontologies have satisfactory difficulty, discrimination and reliability?

Q2: Do the ontology-based generation strategies, the levels of the questions in Bloom's taxonomy, and the types of the questions affect the questions' difficulty and discrimination?

5.1.2 Participants

In 2013/2014, second and third year undergraduate students registered in the Data Networking course (of Manchester, 2013c) and the Computer Networks course (of Manchester, 2013a) at the University of Manchester volunteered to take part in the experiment. Students were asked to answer three different tests. In total, 126 students attempted test-one, 88 students attempted test-two, and 89 students attempted test-three.

Students accessed the three tests using the Moodle Virtual Learning Environment (VLE) (of Manchester, 2013b). Their responses were recorded and used to analyse the quality of the assessment tests, and the quality of questions in each test.

5.1.3 Experimental Set-up

The Ontology-based Personalised Feedback Generator (OntoPeFeGe) system explained in Chapter 4 was used to generate 44 questions. The domain ontologies used in the generation process were selected using TONE. TONE selected the Computer Networks (CN) (Murugan.R et al., 2013) and the OpenCyc (Matuszek et al., 2006), as the CN ontology had better coverage of the Data Networking and the Computer Networks courses compared to the OpenCyc ontology. On the other hand, the OpenCyc ontology had higher semantic richness scores compared to the CN ontology (see Fig 3.10 and Fig 3.11).

The questions were syntactically checked by a domain expert who is a lecturer in the School of Computer Science and teaches the Computer Networks course. After that, the questions were imported into the Moodle VLE to form three different tests. Table 5.1 illustrates the distribution of the questions generated using the ontology-based generation strategies. Each test contained true and false, multiple choice and short answer questions (see Table 5.2), and consisted of questions which assess students' cognition at different levels in Bloom's taxonomy (see Table 5.3). Table 5.2 shows that the number of short answer questions used in the experiment was small compared to the true and false and multiple choice questions. This is due to that fact that Grubisic (Grubisic, 2012; Grubisic et al., 2013) and Al-Yahya (Al-Yahya, 2011; Al-Yahya, 2014) defined only two generation strategies and stem templates for generating short answer questions. See Appendix D for more details on the assessment tests.

Table 5.1: Distribution of questions based on the generation strategies

Test	Number of Questions	Generation Strategies		
		Class	Terminology	Property
1	14	1	4	9
2	16	1	4	11
3	14	1	4	9

Table 5.2: Distribution of questions based on types of questions

Test	Number of Questions	Types of Question		
		True/ False	Multiple Choice	Short Answer
1	14	4	10	0
2	16	4	11	1
3	14	4	9	1

5.2 Evaluation Methods

This section presents two statistical evaluation theories, the Classical Test Theory (Alagumalai and Curtis, 2005; Ding and Beichner, 2009; Doran, 1980; Cohen et al., 2013; Erguven, 2014) and the Item Response Theory (Baker, 2001; De Ayala, 2009; Hambleton and Swaminathan, 1985; Hambleton, 1991; Reckase, 2009; Schmidt and Embretson, 2003; Toland, 2014), which have been used to quantitatively evaluate the quality of the ontology-based auto-generated questions and tests.

5.2.1 Classical Test Theory (CTT)

Classical Test Theory (CTT) is used to evaluate the quality of questions and assessment tests in learning environments using the statistical measures described in the following sections(Alagumalai and Curtis, 2005; Ding and Beichner, 2009; Doran, 1980; Cohen et al., 2013; Erguven, 2014).

Question difficulty index

The question's difficulty index (P) measures the question easiness and it is defined as the proportion of students choosing the correct answer (Ding and Beichner, 2009;

Table 5.3: Distribution of questions based on the level of the questions in Bloom's taxonomy

Test	Number of Questions	Level of the Question Bloom's Taxonomy			
		Knowledge	Comprehension	Application	Analysis
1	14	4	4	4	2
2	16	4	4	4	4
3	14	4	4	4	2

Table 5.4: Range of values and descriptions of the question's quality measurements

Question quality measurements	Range of values	Description	Literature
Difficulty index	[0, 0.35)	Very difficult	(Doran, 1980)
	[0.35, 0.60)	Moderately difficult	(Doran, 1980)
	[0.60, 0.85)	Moderately easy	(Doran, 1980)
	[0.85, 1]	Very easy	(Doran, 1980)
Discrimination index	[0, 0.3)	Low	(Doran, 1980; Ebel, 1979)
	[0.3, 0.6)	Medium	(Doran, 1980; Ebel, 1979)
	[0.6 , 1]	High	(Doran, 1980; Ebel, 1979)
Reliability (point biserial correlation coefficient)	[0.0, 0.3)	Low	(Dancey and Reidy, 2004)
	[0.3, 6.0)	Medium	(Dancey and Reidy, 2004)
	[0.6, 1]	High	(Dancey and Reidy, 2004)

Doran, 1980; Cohen et al., 2013; Schmidt and Embretson, 2003):

$$P = \frac{N1}{N} \tag{5.1}$$

Where N1 is the number of correct answers and N is the total number of students taking the test. P values range from 0 to 1. Table 5.4 shows that questions with high difficulty indices are easy while questions with low difficulty indices are difficult.

Question discrimination index

The question's discrimination index measures how well the question could discriminate between high ability (students with high scores) and low ability students (students with low scores) (Ding and Beichner, 2009; Doran, 1980; Cohen et al., 2013). The discrimination index is defined as the difference between the proportion of the top quartile students who answered the question correctly and the proportion of the bottom quartile students who answered the question correctly (Ding and Beichner, 2009; Doran, 1980):

$$Discrimination\ index = \frac{N_H - N_L}{N/4} \tag{5.2}$$

Where $\mathrm{N_H}$ and $\mathrm{N_L}$ are the number of correct answers in the top quartile and bottom quartile, and N is the total number of students taking the test. Table 5.4 shows that questions with discrimination indices < 0.3 have low discrimination, while questions with discrimination indices ≥ 0.6 have high discrimination.

Question reliability

The question's reliability is measured using the point biserial correlation coefficient, which is the correlation between students scores in the question and students' total scores (Ding and Beichner, 2009; Schmidt and Embretson, 2003; Brown, 1996):

$$R_{\text{pbi}} = \frac{(\bar{x_1} - \bar{x_0})}{\sigma_{\text{x}}} \sqrt{P_{\text{i}} \times (1 - P_{\text{i}})} \tag{5.3}$$

Where R_{pbi} is the point biserial correlation coefficient for question i, X is students' total score in the test, $\bar{x_1}$ is the average total score of students who correctly answered question i, $\bar{x_0}$ is the average total score for students who did not answer question i correctly, σ_{x} is the standard deviation of students' total scores, and P_{i} is the difficulty index for question i. R_{pbi} value ranges from [-1, 1] and high R_{pbi} value means that students who selected the correct answer are students with high total scores and students who selected the incorrect answer are students with low total scores. Higher R_{pbi} values are better (Ding and Beichner, 2009). The reliability is also used to measure the question's discrimination. Table 5.4 shows that questions with $\text{R}_{\text{pbi}} < 0.3$ have low reliability (discrimination) while questions with $\text{R}_{\text{pbi}} \geq 0.6$ have strong reliability (discrimination).

Test discrimination power

The test discrimination power is measured using Ferguson's delta (δ) (Ferguson, 1949), which investigates how broadly the test scores are distributed over the possible range of scores (Zhang and Lidbury, 2013). Ferguson's delta (δ) is measured using the following formula:

$$\delta = (\frac{N^2 - \sum_{i=1}^{K} f_{\text{i}}^2}{N^2 - N^2/(K+1)}) \tag{5.4}$$

Where N is the total number of student who attempted the test, f_{i} is the number of students whose total score is i, k is the number of questions in a test. δ ranges from 0 to 1, where 0 indicates that the test has minimal discrimination and this occurs when all students have the same score. On the other hand, when δ is 1 this means all possible scores occur in the test with the same frequency (Hankins, 2007). Ferguson's delta (δ) value greater than 0.9 is considered a good discrimination as it represent the normal distribution of scores (Kline, 1986; Kline, 2013a; Kline, 2013b).

Test reliability

The test reliability is measured using Cronbach's α (Cronbach and Shavelson, 2004), which measures the internal consistency of the test by finding the correlation between each question's score in the test and the whole test score. In other words, Cronbach's

Table 5.5: Cronbach's α range of values and description

Cronbach's α	**Internal consistency**
$\alpha \geq 0.9$	Excellent
$0.7 \leq \alpha < 0.9$	Good
$0.6 \leq \alpha < 0.7$	Acceptable
$0.5 \leq \alpha < 0.6$	Poor
$\alpha < 0.5$	Unacceptable

α examines whether a test is constructed from questions that address the same material and it is measured using the following formula:

$$\text{Cronbach } \alpha = \frac{K}{K-1}\left(1 - \frac{\sum_{i=1}^{n} P_\mathrm{i}(1-P_\mathrm{i})}{\sigma_\mathrm{x}{}^2}\right) \tag{5.5}$$

Where K is the number of questions in a test, P_i is the difficulty index of the i^{th} question in the test, $\sigma_\mathrm{x}{}^2$ is the variation of the total test scores. Table 5.5 illustrates the Cronbach's α values and the test internal consistency description.

The CTT statistical measures have a range of desired values that questions and tests in learning environments are recommended to achieve (see Table 5.6).

Table 5.6: CTT statistical measures desired values

CTT statistical measures	**Desired values**	**Literature**
Difficulty index average	[0.30-0.90]	(Doran, 1980; Cronbach and Shavelson, 2004)
Discrimination index average	≥ 0.30	(Doran, 1980; Zhang and Lidbury, 2013)
Point biserial coefficient average	≥ 0.20	(Ding and Beichner, 2009; Zhang and Lidbury, 2013; Kline, 1986) (Kline, 2013a; Ding et al., 2006; Corkins, 2009; Lord, 1952)
Cronbach's α	≥ 0.70	(Doran, 1980; Cronbach and Shavelson, 2004)
Ferguson's δ	≥ 0.90	(Kline, 1986; Kline, 2013a; Kline, 2013b)

Even though the CTT is widely used in evaluating the questions and tests in learning environments (Schmidt and Embretson, 2003), it is limited in several ways: 1) Question's difficulty, discrimination, and reliability values vary across different samples of students (Haladyna, 1994). For example, questions' are easy when the sample of students used in the analysis have high ability, and questions are difficult when the sample of students have low ability (De Ayala, 2009). 2) Students and test characteristics can not be separated and they are interpreted in the context of each other (Hambleton, 1991). Question's difficulty, discrimination, and reliability values depend on the sample of students and the ability of students depends on the assessment test. For example, an easy test will appear students to have high ability and vice-versa. 3) CTT is test oriented rather than question oriented, as it can not predict how a particular student may do in a particular assessment question (Hambleton, 1991).

These limitations have been addressed by the IRT, which is explained in the following section.

5.2.2 Item Response Theory (IRT)

Item Response Theory (IRT)[1] is a family of probabilistic models that relates students' ability (θ) to the probability of answering a test question within a particular category (Lord, 1980). Questions could either have two response categories (correct or incorrect) and are known as dichotomous questions, or several response categories such as a Likert scale with five responses ranging from strongly agree to strongly disagree. These types of questions are known as polytomous questions.

Similar to CTT, IRT models are used to assess the question's difficulty and discrimination. However, IRT addresses the CTT drawbacks by achieving the following (Baker, 2001; Reckase, 2009; Fotaris et al., 2010): 1) The question's difficulty and discrimination values measured using IRT are sample independent, i.e., question's difficulty and discrimination values does not change across different samples of students such as high ability and low ability students. 2) Students and test characteristics in IRT can be separated; the question's difficulty and discrimination are independent of the sample of students used in the analysis. Moreover, students' ability is independent of the assessment questions.

Models

IRT includes the following set of probabilistic models, which differentiate in the number of parameters used to describe the characteristics of the assessment questions:

1) One parameter logistic model (1PL): This is the simplest model in IRT as it has one parameter for describing the characteristics of a student (ability), and one parameter for describing the characteristics of an assessment question (difficulty). This model assumes that all questions in the test are equally discriminating. 1PL model is presented in the following equation:

$$P(X_{\mathrm{ij}} = 1 | \theta_{\mathrm{j}}, b_{\mathrm{i}}) = \frac{e^{\theta_{\mathrm{j}} - b_{\mathrm{i}}}}{1 + e^{\theta_{\mathrm{j}} - b_{\mathrm{i}}}} \tag{5.6}$$

Where X_{ij} represents the response of a student j to question i, X_{ij} = 1 means that question i is answered correctly and X_{ij} = 0 means that question i is answered incorrectly. θ_{j} represents the ability of student j, and b_{i} is the difficulty parameter of question i.

[1] Item is an assessment question.

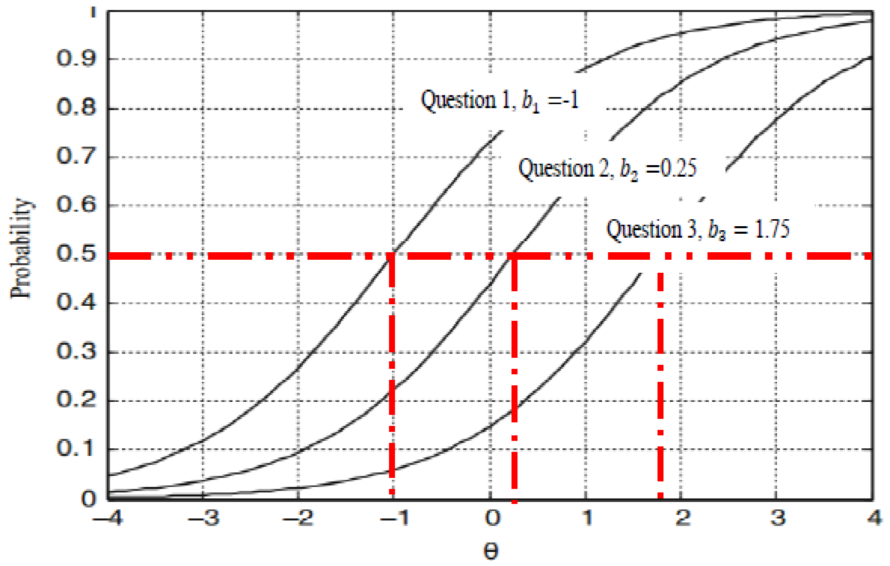


Figure 5.1: ICCs for three questions which have different difficulty indices b_i = -1, 0.25, 1.75 (Reckase, 2009)

The IPL model which represents the probability of a correct response as a function of θ_j can be illustrated in a graph called the Item Characteristic Curve (ICC). Fig 5.1 shows the ICCs for three different questions which have different difficulty indices b_i= -1, 0.25, and 1.75 and identical discrimination values (maximum slope in the ICC). Questions with low b_i values (for example question 1 in Fig 5.1) are easy and most students even those with low ability have a high probability of answering them correctly, while questions with high b_i values (for example question 3 in Fig 5.1) are difficult and students with low ability have a low probability of answering them correctly. Fig 5.1 also shows that b_i is the point on the ability scale (θ) where the students have a 50% probability of answering the question correctly.

2) Two parameter logistic model (2PL): This model is a slightly more complex model, as it considers both the question's difficulty and discrimination. The model is presented in the following equation:

$$P(X_{\mathrm{ij}} = 1|\theta_\mathrm{j}, a_\mathrm{i}, b_\mathrm{i}) = \frac{e^{a_\mathrm{i}(\theta_\mathrm{j} - b_\mathrm{i})}}{1 + e^{a_\mathrm{i}(\theta_\mathrm{j} - b_\mathrm{i})}} \tag{5.7}$$

Where a_i is the question's discrimination parameter. The higher the value of a_i, the more sharply the question discriminates between high ability and low ability students. Fig 5.2 illustrates the ICC for three questions, which have different discrimination and

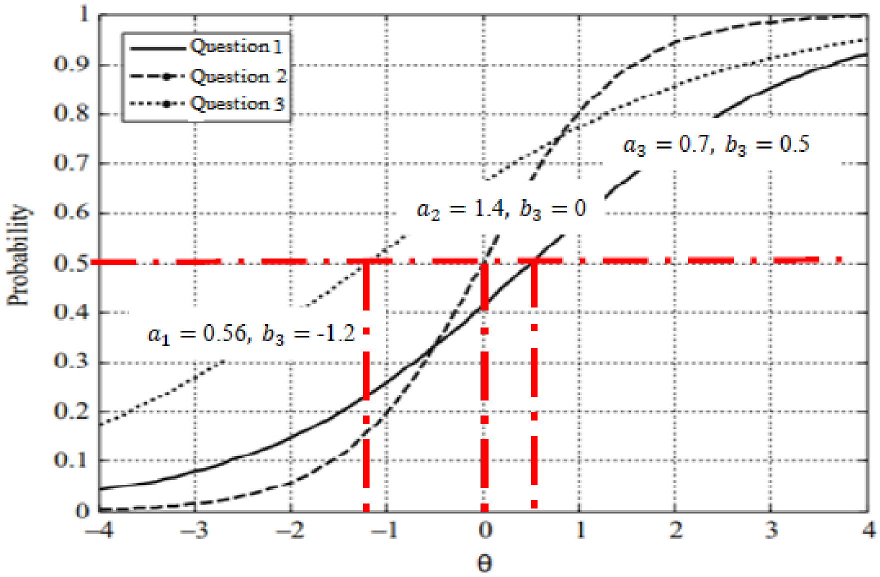


Figure 5.2: ICCs for three different assessment questions (Reckase, 2009)

difficulty values. Question 2 (a_2 = 1.4) has a higher discrimination value compared to question 1 and question 3. Fig 5.2 also shows that students with low ability (θ = -1) find question 1 easier to answer compared to question 2, as they have a higher probability of answering it correctly. On the other hand, students with a higher ability (θ = 1) find question 1 more difficult compared to question 2.

3) Three parameter logistic model (3PL): This model is more complex than the previous models. It considers the possibility that the student correct answers could be obtained by guessing. The model is presented in the following equation:

$$P(X_{\mathrm{ij}} = 1|\theta_{\mathrm{j}}, a_{\mathrm{i}}, b_{\mathrm{i}}, G_{\mathrm{i}}) = G_{\mathrm{i}} + (1 - G_{\mathrm{i}})\frac{e^{a_{\mathrm{i}}(\theta_{\mathrm{j}} - b_{\mathrm{i}})}}{1 + e^{a_{\mathrm{i}}(\theta_{\mathrm{j}} - b_{\mathrm{i}})}} \tag{5.8}$$

Where G_{i} is the guessing parameter which accounts for the possibility that all students even the ones with very low ability have a non-zero probability of answering a question correctly by guessing. Fig 5.3 illustrates the ICC for the 3PL model. The figure shows a question in which students with low ability have a probability of 0.16 (lower dashed line) to answer it correctly.

The question's difficulty and discrimination parameters obtained using the IRT models could be described in Table 5.7 (Baker, 2001; Hambleton, 1991; Hardouin and Mesbah, 2004; Hasmy, 2014).

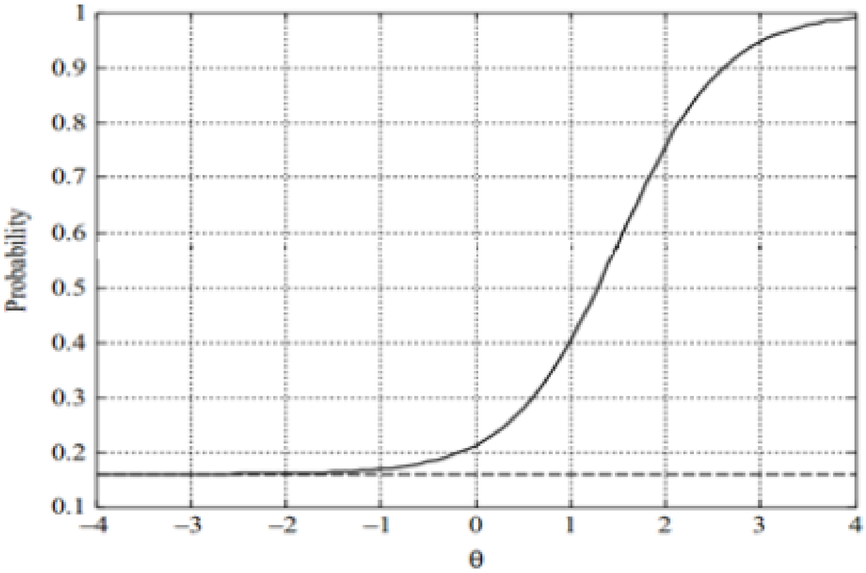


Figure 5.3: ICC for the 3PL model (Reckase, 2009)

Table 5.7: Difficulty and discrimination values and description in IRT

Difficulty parameter	**Description**	**Discrimination parameter**	**Description**
[-2, - ∞)	Very easy	[0.01, 0.35)	Very low
[-0.5, -2)	Easy	[0.35, 0.65)	Low
(-0.5, 0.5)	Medium	[0.65, 1.35)	Moderate
[0.5, 2)	Difficult	[1.35, 1.70)	High
[2, ∞)	Very difficult	[1.70, ∞)	Perfect

Assumptions

In order to use the IRT models to analyse an assessment's test data, the following two assumptions underlying the model must be satisfied (De Ayala, 2009; Hambleton and Swaminathan, 1985; Reckase, 2009; Toland, 2014; Comer and Kendall, 2013):

1) Unidimensionality: The unidimensionality assumption means that the assessment test measures only one ability parameter (θ). For example, a computer networks test is assumed to be unidimensional if it only examines students' ability in the field of computer networks (e.g., the network topology and the Open Systems Interconnection model). Unidimensionality is examined using the Principle Component Analysis (PCA) test (Chou and Wang, 2010). PCA is a dimension reduction technique, which could be used to examine if the data (students responses to questions in the assessment test) can be reduced from T-components which are the number of questions in the assessment test to C-components where C < T (Jeong et al., 2009; James et al., 2014). If the T-components could be reduced to one component (C = 1) which explains the maximum variance in the observed data, then only one ability parameter (θ) is underlying the assessment test and the test is unidimensional. See Appendix E for more details on the PCA.

PCA outputs the number of components underlying the assessment test. If one component is found the Unidimensional IRT (UIRT) models can be used to analyse the assessment test data, otherwise the Multidimensional IRT (MIRT) models can be applied to the assessment test data.

2) Local independence: This assumption states that the only influence on an individual question response is that of the ability parameter being measured (De Ayala, 2009). This indicates that there is no influence on the individual question response from other questions or other ability variables. The term local is used to indicate that responses are assumed to be independent at the level of individual students having the same ability (θ). Local independence is examined using the Local Dependence chi-square (LD x^2) test which is applied for each pair of questions in the assessment test (Chen and Thissen, 1997). The LD x^2 is computed by comparing the observed and expected frequencies of students' responses for each pair of questions. In addition, it is applied under the *null hypothesis* that there is local independence between each pair of questions. See Appendix E for more details on the LD x^2 test.

Multidimensional Item Response Theory

Multidimensional Item Response Theory (MIRT) is an extension to the UIRT and it is used when the unidimensionality assumption is violated (Reckase, 2009). MIRT models relate students' ability (θ) to the probability of a correct response using an item characteristics surface (ICS). Similar to the ICC graph, ICS illustrates the question's difficulty, the question's discrimination, and the guessing parameter.

Fig 5.4 illustrates an ICS for a two-dimensional IRT model, which has only one difficulty parameter and two discrimination parameters (one discrimination parameter per ability dimension). Similar to the UIRT models, large discrimination parameter values indicate better discrimination.

Model selection methods

Selecting the IRT model, which is the closest fit to the assessment test data is essential to obtain question's difficulty and discrimination values which are invariant across

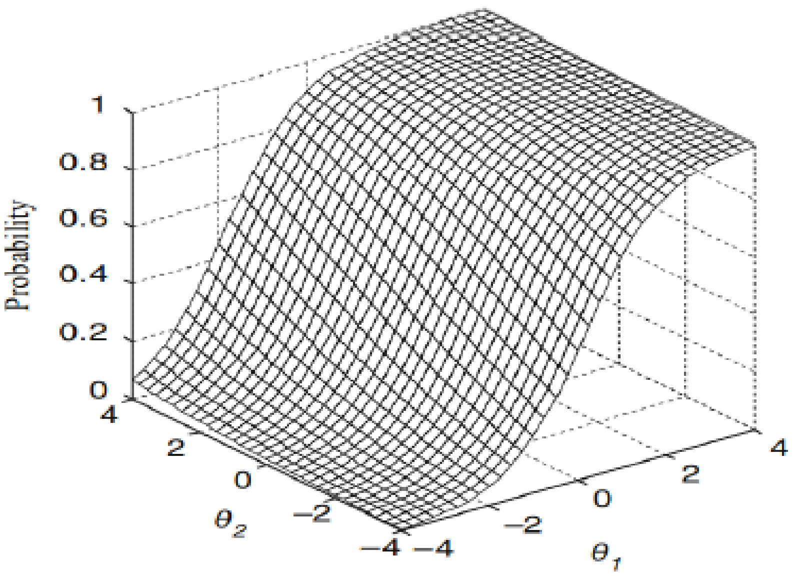


Figure 5.4: ICS for a two-dimensional IRT model (Reckase, 2009)

different samples of students (Hambleton and Swaminathan, 1985; Gler et al., 2014). In this thesis the following methods have been used to select the IRT model with the closest fit to the assessment test data:

1) The likelihood ratio: The Liklihood Ratio (LR) statistical test (De Ayala, 2009; Toland, 2014; Comer and Kendall, 2013) could be used to select the best IRT model from the three nested models (1PL, 2PL, and 3PL). Moreover, it could be used to select the best model from UIRT and MIRT models, which have different dimensions and the same number of parameters. LR is a chi-square based statistical test and it is measured as the difference between deviances for the two IRT models being compared. The deviance statistic is defined as:

$$-2 \times \log(MaximumLikelihood(model)) \tag{5.9}$$

The maximum likelihood (ML) is obtained for the IRT models using Bock and Aitkin's Expectation-Maximization algorithm (BAEM) (Bock and Aitkin, 1982). The LR statistical test is applied under the *null hypothesis* that there is no difference between the two compared models (model 1 and model 2). If the difference between the models deviances which has a chi-square distribution is statistically significant then model 2 has better fit to the assessment test data compared to model 1, otherwise, model 1 has a better fit to the assessment test data.

2) Information theoretic methods: The LR test tends to select models with more parameters (e.g., the 2PL model) which are more complex models and may be a better fit to the assessment test data compared to the models with fewer parameters (e.g., 1PL model) (De Ayala, 2009; Kang and Cohen, 2007). Therefore, the Akaike's Information Criterion (AIC) (Akaike, 1974) and the Bayesian Information Criterion (BIC) (Schwarz, 1978) are model selection methods, which penalise the IRT models according to their complexity. They are used as a trade-off between the complexity of the model and the goodness of fit between the model and the assessment test data. Akaike's Information Criterion is measured using the following equation:

$$AIC(model) = -2 \times \log(MaximumLikelihood(model)) + 2 \times N_{\text{parm}} \tag{5.10}$$

Where $-2 \times \log(MaximumLikelihood(model))$ is the deviance and N_{parm} is the number of parameters being estimated. The model with the smallest AIC is the closest fit to the assessment test data (De Ayala, 2009; Toland, 2014).

Bayesian Information Criterion is measured using the following equation:

$$BIC(model) = -2 \times \log(MaximumLikelihood(model)) + \log(N) \times N_{\text{parm}} \tag{5.11}$$

Where N_{parm} is the number of parameters being estimated, N is the sample size which is the total number of students who attempt the assessment test. The model with the smallest BIC is the closest fit to the assessment test data (De Ayala, 2009; Toland, 2014). Equation. 5.10 shows that AIC penalise the model based on the number of parameters estimated and it does not take into account the sample size. This results in AIC favouring more complex models when the sample size increase (Kang and Cohen, 2007; DeMars, 2012). On the other hand, BIC tends to select models that are simpler than those selected by AIC when the sample size is large (Kang and Cohen, 2007). Equation. 5.11 shows that BIC takes into account the sample size and the penalty for model complexity increases for large samples (DeMars, 2012).

5.3 Results and Discussion

This section illustrates the experiment results obtained using the CTT and IRT.

Table 5.8: Questions' analysis in test-one (number of students' attempts = 126)

Question	**Difficulty index**	**Discrimination index**	$\mathbf{R_{pb}}$
1	Moderately easy (0.770)	Low (0.226)	Low (0.216)
2	Very easy (0.897)	Low (0.258)	Low (0.261)
3	Moderately difficult (0.579)	Medium (0.516)	Medium (0.445)
4	Moderately difficult (0.365)	Medium (0.355)	Low (0.270)
5	Moderately difficult (0.373)	Low (0.290)	Low (0.233)
6	Moderately difficult (0.508)	High (0.710)	Medium (0.479)
7	Moderately difficult (0.532)	High (0.613)	Medium (0.460)
8	Moderately easy (0.635)	High (0.677)	Medium (0.541)
9	Moderately easy (0.651)	High (0.613)	Medium (0.521)
10	Very difficult (0.333)	Low (0.226)	Medium (0.333)
11	Very difficult (0.087)	Low (0.129)	Low (0.267)
12	Moderately difficult (0.540)	Medium (0.516)	Medium (0.347)
13	Moderately difficult (0.595)	High (0.645)	Medium (0.496)
14	Moderately difficult (0.484)	Medium (0.484)	Medium (0.362)

Q1: Do the questions and tests generated from domain ontologies have satisfactory difficulty, discrimination and reliability?

The quality of the questions and tests was obtained using the CTT measurements which are widely used in learning environments. However, the CTT measurements are sample dependent. Therefore, the IRT was also used to obtain the questions' difficulty and discrimination in the three tests, as it has the invariance assumption, which states that the question's difficulty and discrimination values are sample independent.

Analysis using the Classical Test Theory (CTT)

Tables 5.8 to 5.13 illustrate the quality measurements of the auto-generated questions and tests after applying the CTT to test-one, test-two and test-three respectively. The analysis of the auto-generated questions and tests revealed the following:

1) Difficulty

The difficulty is measured for the individual questions in the tests using the difficulty index, which is the proportion of students choosing the correct answer. The question difficulty indices varied from very easy to very difficult in test-one (see Table 5.8) with values in the range 0.085 to 0.897, and very easy to moderately difficult in test-two (see Table 5.10) and test-three (see Table 5.12) with difficulty indices in the ranges 0.284 to 0.773 and 0.247 to 0.798 respectively. 16% (7 questions out of 44) of the questions in the three tests were either very easy or very difficult. These questions were more likely to have low discrimination. The CTT analysis results also revealed when applied to tests administered to third year undergraduate students that the three tests had medium difficulty with 0.525, 0.540, and 0.564 average difficulty index values. The difficulty

Table 5.9: Test-one analysis (number of students' attempts = 126)

Test statistics	Value	Description	Desired values
Questions' difficulty index average	0.525	Medium difficulty	[0.3-0.9]
Questions' discrimination index average	0.445	Satisfactory	≥ 0.3
Questions' reliability average (point biserial coefficient)	0.266	Satisfactory	≥ 0.2
Test reliability index	0.540	Poor	≥ 0.7
Test discrimination power (Ferguson's delta)	0.947	Satisfactory	≥ 0.9

Table 5.10: Questions' analysis in test-two (number of students' attempts = 88)

Question	Difficulty index	Discrimination index	R_{pb}
1	Moderately difficult (0.580)	Medium (0.409)	Low (0.297)
2	Moderately easy (0.739)	Medium (0.318)	Low (0.241)
3	Moderately difficult (0.545)	Low (0.273)	Low (0.196)
4	Moderately easy (0.761)	Medium (0.318)	Low (0.274)
5	Moderately difficult (0.432)	Medium (0.409)	Medium (0.382)
6	Moderately difficult (0.420)	Medium (0.500)	Medium (0.488)
7	Moderately easy (0.614)	Low (0.182)	Low (0.193)
8	Very difficult (0.295)	Medium (0.455)	Medium (0.368)
9	Moderately difficult (0.432)	High (0.682)	Medium (0.489)
10	Moderately difficult (0.523)	Medium (0.318)	Low (0.263)
11	Very difficult (0.284)	Medium (0.409)	Medium (0.378)
12	Moderately easy (0.773)	Medium (0.545)	Medium (0.457)
13	Moderately difficult (0.489)	High (0.727)	Medium (0.524)
14	Moderately easy (0.625)	Medium (0.455)	Medium (0.455)
15	Moderately easy (0.636)	Medium (0.500)	Medium (0.371)
16	Moderately difficult (0.500)	Medium (0.545)	Medium (0.518)

fall within the CTT desired range of values. In addition, the tests' average difficulty index values are very close to 0.5, which is the value that the test authors are advised to try and achieve when constructing questions and where the test has the maximum discrimination (Doran, 1980; Mitkov et al., 2009; Mitkov et al., 2006). The maximum discrimination is obtained when all students with high ability (students with high scores) answer the questions correctly and all students with low ability do not answer the questions correctly.

2) Discrimination

The discrimination is measured for the individual questions and the entire assessment tests using the discrimination index and the discrimination power metrics respectively. A question discrimination index measures how well the question could discriminate between high ability and low ability students. On the other hand, the test's discrimination power examines how students' total scores are distributed over the possible range of scores in an assessment test. Tests with a broad distribution of total scores have better discrimination among students. The question discrimination indices in the three tests

Table 5.11: Test-two analysis (number of students' attempts = 88)

Test statistics	Value	Description	Desired values
Questions' difficulty index average	0.540	Medium difficulty	[0.3-0.9]
Questions' discrimination index average	0.440	Satisfactory	≥ 0.3
Questions' reliability average (point biserial coefficient)	0.360	Satisfactory	≥ 0.2
Test reliability index	0.560	Poor	≥ 0.7
Test discrimination power (Ferguson's delta)	0.955	Satisfactory	≥ 0.9

have positive values, which indicate that the questions may not need to be reviewed or eliminated from the assessment tests (Doran, 1980; Mitkov et al., 2006; Mitkov and Ha, 2003). According to Baker, questions with negative discrimination may need to be reviewed either because they are miswritten or there is a misinformation prevalent among high ability students (Baker, 2001). Hambleton added that negatively discriminating questions may need to be discarded from the assessment tests because these questions have a problem such as miskeying (a question is a miskey if there are more students with high ability who choose the incorrect answer than the key) (Hambleton and Swaminathan, 1985; Hambleton, 1991).

The question discrimination indices varied from low to high in the three tests with values in the ranges 0.129 to 0.710, 0.182 to 0.727, and 0.273 to 0.727 respectively. Nevertheless, the three tests have satisfactory average discrimination values above 0.30 (see Table 5.9, Table 5.11, and Table 5.13) which indicate that the questions could efficiently discriminate between high ability and low ability students (Corkins, 2009; Thorndike and Hagen, 1961).

The CTT was also used to obtain the tests' discrimination power using Ferguson's delta. The results revealed that the three tests had satisfactory discrimination power with Ferguson's delta values above 0.90, which is the discrimination power for normally distributed test scores.

3) Reliability

The questions' reliability was measured using the point biserial correlation coefficients[2] (R_{pb}) which revealed that the questions' reliability values in the three tests were positive and varied between low and medium with values in the ranges 0.216 to 0.541, 0.193 to 0.524, and 0.234 to 0.524 respectively. The results also revealed that the three tests had satisfactory average reliability values above 0.2. The tests' reliability values were also obtained using Cronbach's α which revealed that test-one and test-two had poor reliability with 0.54 and 0.56 reliability index values respectively,

[2] Point biserial correlation coefficients are used as discrimination indices.

Table 5.12: Questions' analysis in test-three (number of students' attempts = 89)

Question	Difficulty index	Discrimination index	$\mathbf{R_{pb}}$
1	Moderately easy (0.708)	Medium (0.409)	Low(0.294)
2	Moderately easy (0.652)	Medium (0.318)	Low (0.258)
3	Moderately easy (0.764)	Medium (0.409)	Medium (0.360)
4	Moderately easy (0.640)	Medium (0.318)	Medium (0.300)
5	Moderately easy (0.663)	High (0.636)	Medium (0.504)
6	Moderately difficult (0.483)	Medium (0.591)	Medium (0.429)
7	Very difficult (0.337)	Medium (0.455)	Medium (0.433)
8	Moderately difficult (0.427)	Medium (0.591)	Medium (0.524)
9	Moderately easy (0.798)	Medium (0.409)	Medium (0.373)
10	Moderately easy (0.674)	Medium (0.500)	Medium (0.428)
11	Very difficult (0.247)	Low (0.273)	Low (0.239)
12	Moderately difficult (0.360)	High (0.727)	Medium (0.508)
13	Moderately difficult (0.416)	Medium (0.591)	Medium (0.465)
14	Moderately easy (0.730)	Medium (0.545)	Medium (0.476)

Table 5.13: Test-three analysis (number of students' attempts = 89)

Test statistics	Value	Description	Desired values
Questions' difficulty index average	0.564	Medium difficulty	[0.3-0.9]
Questions' discrimination index average	0.484	Satisfactory	≥ 0.3
Questions' reliability average (point biserial coefficient)	0.399	Satisfactory	≥ 0.2
Test reliability index	0.604	Acceptable	≥ 0.7
Test discrimination power (Ferguson's delta)	0.966	Satisfactory	≥ 0.9

while test-three had a higher reliability value of 0.604 which is considered acceptable (see Table 5.5). The tests' low reliability values obtained using Cronbach's α were due to the fact that the individual questions in each test had satisfactory reliability values (R_{pb}) which were not high enough to improve the tests' overall reliability (Jones, 2009). Higher R_{pb} values are desired and lower R_{pb} values indicate that a question is not testing the same educational material or may not be testing the same educational material at the same level (Ding and Beichner, 2009). In this experiment, the questions were generated from the same domain ontologies (OpenCyc and CN). As a result, the context of the educational material being tested is known. However, the auto-generated questions were designed to assess different educational concepts at different levels of Bloom's taxonomy, which may result in satisfactory reliability values at the questions' level (average R_{pb}) but low reliability values at the test's level (Cronbach's α). Further investigation is carried out in the following sections to examine the relationship between the measurements of questions' quality derived using the CTT with each of the following: the ontology-based generation strategy, the level of the question in Bloom's taxonomy and the type of question generated.

Analysis using the Item Response Theory (IRT)

This section illustrates the measurements of the questions quality obtained using the IRT. The IRT model estimation program, IRTPRO (Cai et al., 2011; Paek and Han, 2013), was used to apply several IRT models to test-one, test-two, and test-three after checking the following IRT assumptions:

1. The unidimensionality assumption, which means that the assessment test measures only one ability (θ) parameter. The assumption was checked using the Principle Component Analysis (PCA), which is a dimension reduction technique (see Appendix E for more details) (Chou and Wang, 2010). PCA outputs the number of components (ability parameters) underlying the assessment test. If one component is found, the Unidimensional IRT (UIRT) models can be used to analyse the assessment test data. Otherwise, the Multidimensional IRT (MIRT) models can be applied to the assessment test data.

2. The local independence assumption, which states that the only influence on an individual question response is that of the ability parameter being measured. This indicates that there is no influence on an individual question response from other questions or ability parameters. The local independence assumption was checked using the Local Dependence chi-square test (LD x^2) (see Appendix E for more details).

The IRT model with the best fit to the assessment test's data was chosen using the IRT model selection methods. After that, the model was used to obtain the questions' difficulty and discrimination.

1) Checking the IRT model assumptions

Fig 5.5 and Table 5.14 illustrate the results obtained by applying the PCA to test-one, which consists of 14 questions and was answered by 126 students. Initially, 14 components were identified; i.e., the number of components equals the number of questions in test-one.

As shown in Fig 5.5 and Table 5.14 applying the PCA to test-one data results in six components with eigenvalues greater than one. The first component had a 2.225 eigenvalue which is higher than the next five components (1.635, 1.248, 1.213, 1.078, and 1.004). 15.894% of the test variance was explained by the first component and a cumulative variance of 60.02% was explained by the first six components (see Table

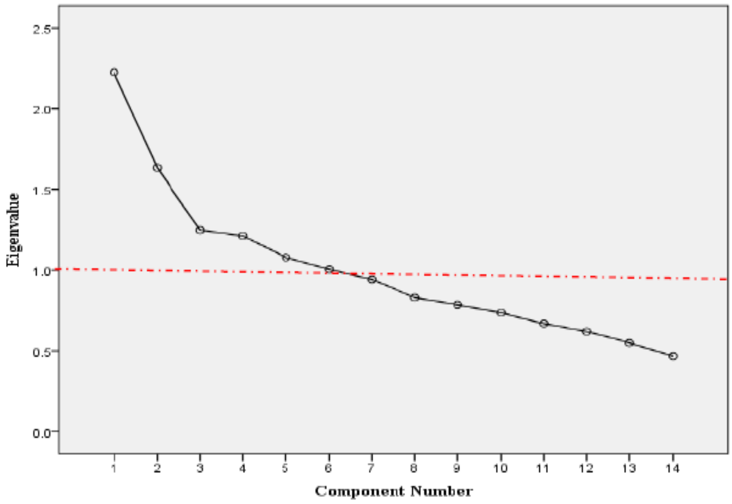


Figure 5.5: Scree plot for the 14 questions in test-one

Table 5.14: Total variance explained by each component in test-one

Component	Initial Eigenvalues		
	Total	% of Variance	Cumulative %
1	2.225	15.894	15.894
2	1.635	11.678	27.572
3	1.248	8.914	36.485
4	1.213	8.661	45.146
5	1.078	7.701	52.847
6	1.004	7.173	60.020
7	0.942	6.731	66.751
8	0.830	5.926	72.677
9	0.785	5.608	78.285
10	0.736	5.258	83.543
11	0.668	4.772	88.315
12	0.620	4.428	92.743
13	0.549	3.923	96.666
14	0.467	3.334	100.000

5.14). The results obtained using the PCA suggests that test-one is not unidimensional and it does not measure a single ability parameter.

The local dependence assumption was also investigated on test-one data using the LD x^2 test. The results revealed that the questions are independent of each other. The LD x^2 test for questions in test-one is illustrated in detail in Appendix F.

2) Analysis of the model data fit

The PCA analysis revealed that test-one is not unidimensional, and six components had eigenvalues greater than one. Therefore, the model's data fit analysis was examined using the UIRT models, and the MIRT models starting from two dimensions and up to six dimensions. The following abbreviations are used throughout the analysis:

Table 5.15: Goodness of fit statistics for UIRT (2PL) and MIRT (2PL) models in test-one

IRT models	**-2 log likelihood**	**Free parameters**	**AIC**	**BIC**
UIRT (2PL)	2085.82	28	2141.82	2144.63
2-MIRT (2PL)	2057.31	41	2139.31	2143.43
3-MIRT (2PL)	2044.58	56	2156.58	2162.2

Table 5.16: Chi-square tests comparing the fit of UIRT (2PL) and MIRT (2PL) models

Comparison	**x^2**	**Degree of Freedom (DF)**	**P-value**
UIRT (2PL) 2-MIRT (2PL)	29.5	13	Pvalue <0.01
2-MIRT (2PL) 3-MIRT (2PL)	11.47	15	Pvalue >0.05

UIRT (M)

[D]- MIRT (M)

Where M is the type of IRT model which could be One Parameter Logistic model (1PL), Two Parameter Logistic model (2PL), or Three Parameter Logistic model (3PL). D is only used with MIRT as it represents the number of dimensions in IRT.

The analysis started with the 2PL model. Table 5.15 illustrates the -2 log likelihood, Akaike's information criterion (AIC), and the Bayesian information criterion (BIC) goodness of fit statistics after applying UIRT (2PL) and MIRT (2PL) models to test-one.

Table 5.16 shows the chi-square test between several models. The results revealed that AIC, BIC and chi-square tests gave consistent results identifying the 2-MIRT (2PL) model as the best fit for test-one data, as 2-MIRT (2PL) had the smallest AIC and BIC values, and the chi-square test revealed a statistically significant difference between the 2-MIRT (2PL) and the UIRT (2PL) models.

Further investigations were carried out to examine the effect of changing the type of IRT model (e.g., 2PL and 3PL) in 2-MIRT on the goodness of fit statistics. Table 5.17 shows the goodness of fit statistics for the 2-MIRT (2PL) and the 2-MIRT (3PL) models. The results revealed that 2- MIRT (2PL) fits test-one data better than 2-MIRT (3PL), as it has lower AIC and BIC values, and the chi-square test revealed no statistically significant difference (P-value $>$ 0.05) between the 2- MIRT (2PL) and the 2-MIRT (3PL) models (see Table 5.18). In summary, the 2-MIRT (2PL) model is the best fit to test-one.

The 2-MIRT (2PL) model has one difficulty index and a discrimination vector with two indices, one in each ability dimension. Table 5.19 illustrates the difficulty indices and discrimination indices which are the discrimination vector magnitude (Reckase,

Table 5.17: Goodness of fit statistics for 2-MIRT (2PL) and 2-MIRT (3PL) models

IRT models	-2 log likelihood	Free parameters	AIC	BIC
2-MIRT (2PL)	2057.31	41	2139.31	2143.43
2-MIRT (3PL)	2042.87	70	2182.87	2389.9

Table 5.18: Chi-square tests comparing the fit of 2-MIRT (2PL) and 2-MIRT (3PL) models

Comparison	x^2	Degree of Freedom (DF)	P-value
2-MIRT (2PL) -2-MIRT (3PL)	14.44	29	P-value >0.05

2009). The questions' difficulty indices range from very easy to very difficult. Lower difficulty indices indicate that the questions are easy and more students are able to answer them correctly, while higher difficulty indices mean that the questions are difficult and fewer students could answer them correctly. Table 5.19 also shows the questions' discrimination indices obtained using the IRT analysis. The results revealed that the questions' discrimination indices had similar results to the questions' discrimination indices obtained using the CTT, as all indices were positive. This indicates that the questions may not need to be reviewed or eliminated from the assessment tests (Baker, 2001; Hambleton and Swaminathan, 1985).

The questions' discrimination indices in test-one ranged from very low to perfect. Higher discrimination indices mean that the questions discriminate better between low ability and high ability students while lower discrimination indices represent the opposite.

The IRT assumptions and the analysis of the model data fit were also applied to test-two and test-three. The results revealed that the UIRT (2PL) model is the best fit to both test-two and test-three. Therefore, the UIRT (2PL) model was used to obtain the questions' difficulty and discrimination indices in test-two and test-three (see Table 5.20 and Table 5.21). The questions' difficulty indices in test-two ranged from very easy to difficult, and the questions' discrimination indices ranged from very low to perfect. The questions' difficulty indices in test-three ranged from very easy to very difficult, and the questions' discrimination indices ranged from very low to moderate. In both tests, the questions' discrimination indices were positive, which indicate that the questions may not need to be reviewed or eliminated from the assessment tests (Baker, 2001; Hambleton and Swaminathan, 1985). The analysis results also revealed that 22.7% (10 questions out of 44) of the questions in the three tests were either very easy or very difficult, which resulted in low discriminating questions.

Similar to the CTT, the questions' difficulty and discrimination values obtained using the IRT in the three tests varied across the different ontology-based generation

Table 5.19: Questions' difficulty and discrimination indices using IRT (test-one)

Q	Difficulty index	Discrimination index
1	Medium (-0.030)	Moderate (0.986)
2	Very easy (-2.603)	Low (0.484)
3	Easy (-1.800)	Perfect (1.712)
4	Medium (-0.374)	Moderate (1.043)
5	Difficult (0.837)	Moderate (0.740)
6	Very difficult (7.354)	Very low (0.071)
7	Medium (-0.131)	Moderate (1.073)
8	Easy (-0.515)	High (1.458)
9	Easy (-0.584)	High (1.421)
10	Difficult (0.854)	Moderate (0.972)
11	Very difficult (3.323)	Moderate (0.779)
12	Medium (-0.133)	High (1.500)
13	Medium (-0.391)	Moderate (1.305)
14	Medium (0.072)	Moderate (1.105)

Table 5.20: Questions' difficulty and discrimination indices using IRT (test-two)

Question	Difficulty index	Discrimination index
1	Easy (-1.81)	Very low (0.18)
2	Very easy (-2.23)	Low (0.49)
3	Easy (-0.99)	Very low (0.19)
4	Very easy (-10.33)	Very low (0.11)
5	Difficult (0.66)	Low (0.43)
6	Medium (0.42)	Moderate (0.89)
7	Very easy (-8.44)	Very low (0.05)
8	Difficult (1.16)	Moderate (0.86)
9	Medium (0.28)	Moderate (1.22)
10	Medium (-0.29)	Very low (0.32)
11	Difficult (1.87)	Low (0.52)
12	Easy (-1.66)	Moderate (0.84)
13	Medium (0.02)	Perfect (2.75)
14	Easy (-1.5)	Low (0.35)
15	Easy (-0.98)	Low (0.63)
16	Medium (-0.01)	High (1.48)

strategies, the levels of Bloom's taxonomy, and the types of questions. Further investigation is carried out in the following sections to study the effect of the ontology-based generation strategies, the level of the questions in Bloom's taxonomy, and the type of questions on the questions' difficulty and discrimination indices.

Summary

The CTT analysis revealed that the questions auto-generated by the Ontology-based Personalised Feedback Generator (OntoPeFeGe) and the assessment tests formed from these questions had medium difficulty values, which are very close to the value that the test authors are advised to achieve (0.5) when constructing questions (Doran, 1980; Mitkov et al., 2009; Mitkov et al., 2006). In addition, the questions and tests have satisfactory discrimination values which indicate that the questions and tests could effectively discriminate between high ability and low ability students. The results also

Table 5.21: Questions' difficulty and discrimination indices using IRT (test-three)

Question	Difficulty index	Discrimination index
1	Very easy (-2.51)	Low (0.37)
2	Very easy (-3.32)	Very low (0.19)
3	Very easy (-2.06)	Low (0.62)
4	Easy (-1.51)	Low (0.42)
5	Easy (-0.76)	Moderate (1.17)
6	Medium (0.09)	Moderate (0.70)
7	Difficult (0.88)	Moderate (0.87)
8	Medium (0.31)	Moderate (1.08)
9	Easy (-1.48)	Moderate (1.17)
10	Easy (-0.99)	Moderate (0.86)
11	Very difficult (6.10)	Very low (0.18)
12	Difficult (0.60)	Moderate (1.18)
13	Medium (0.38)	Moderate (1.04)
14	Easy (-1.05)	Moderate (1.32)

revealed that the question discrimination indices have positive values which indicate that the questions may not need to be reviewed or eliminated from the assessment tests.

IRT was also used to assess the quality of questions and tests due to its invariance assumption. Several IRT models were applied to the three tests and the model with the best fit was chosen to obtain the questions' difficulty and discrimination indices. The IRT revealed similar results to the CTT, as 22.7% (10 questions out of 44) of the questions were very easy or very difficult, which results in low discriminating questions. In addition, the questions' discrimination indices had positive values which justify that the auto-generated questions may not need to be reviewed or eliminated from the assessment tests.

Even though the CTT and IRT were used to obtain the questions' difficulty and discrimination indices, there was a correlation between the indices obtained by each theory. Table 5.22 illustrates the correlation between the difficulty indices and the correlation between the discrimination indices. The results revealed a strong relationship between the questions' difficulty indices and a strong relationship between the questions' discrimination indices. The results obtained are in line with the results in (Gler et al., 2014; Abedalaziz and Leng, 2013). Table 5.22 also shows that the CTT difficulty index had a negative correlation with the IRT difficulty index. This result was expected as a high difficulty index in CTT means the question is easy while a high difficulty index in IRT means that the question is difficult. Moreover, the table shows that two correlations were applied to find the relationship between the discrimination indices obtained using the CTT and the IRT, as both the CTT discrimination index and the CTT R_{pb} are used to measure the questions' discrimination in learning environments.

Table 5.22: Correlation between CTT and IRT measurement results of the questions' quality

Questions quality measurements	**Correlation**
Questions difficulty index 1) CTT difficulty index 2) IRT difficulty index	(Pearson R = -0.602, P-value = 8×10^{-6}, P-value <0.05)
Discrimination index 1) CTT R_{pb} 2) IRT discrimination index	(Pearson R = 0.652, P-value = $1 \times 10^{-}6$, P-value <0.05)
Discrimination index 1) CTT discrimination index 2) IRT discrimination index)	(Pearson R = 0.576, P-value = $21 \times 10^{-}6$, P-value <0.05)

Q2: Do the ontology-based generation strategies, the levels of the questions in Bloom's taxonomy, and the types of the questions affect the question's difficulty and discrimination?

The ontology-based question generators discussed in Section 2.3.3 did not test the questions on students to analyse the quality of the auto-generated questions by examining the question's difficulty, and the question's ability to discriminate between high ability and low ability students (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014). This results in having no evidence on how the ontology-based generation strategies, the level of the questions in Bloom's taxonomy (determined by the stem templates), and the type of questions affect the questions' quality and produce more difficult or more discriminating questions.

This section studies the effect of the ontology-based generation strategies, the level of the questions in Bloom's taxonomy, and the type of questions on the question's difficulty and discrimination obtained using the CTT (dependent on the sample of students) and the IRT (independent of the sample of students). This will provide question design guidance for developers and researchers working in the field of question generators.

The study was carried out on the CTT difficulty and discrimination measurement results obtained for the 44 questions (the total number of assessment questions in test-one, test-two, and test-three), and on the IRT difficulty and discrimination measurement results for questions which did not exhibit variance across different samples of students. The IRT invariance assumption was tested on the 44 questions, and questions which exhibited variance in difficulty or discrimination were removed to improve the reliability of the upcoming evaluations.

The following sections investigated the IRT invariance assumption in test-one, test-two, and test-three. After that the effects of the ontology-based generation strategies, the levels of questions in Bloom's taxonomy, and the types of questions on the questions' quality measurements was examined.

Examination of the IRT Invariance Assumption

The invariance assumption in IRT indicates that the IRT measurements of the questions' quality (difficulty and discrimination) across different samples of students should roughly be the same (Baker, 2001). This section aims to study the validity of the IRT invariance assumption in test-one, test-two, and test-three by applying the following:

1. Divide students in each test (test-one, test-two, and test-three) into two groups: low ability students (students with test scores less than 50%) and high ability students (students with test scores above or equal 50%) following the approach in (Hambleton and Swaminathan, 1985). Students in each test could also be divided according to their gender or year of study (De Ayala, 2009; Crocker and Algina, 1986). However, this was not applicable to the experiment carried out in this chapter due to the large difference in students' numbers when students in each test were divided according to their gender or year of study.

2. The IRT model which has the best fit to the whole sample of students in each test (e.g., 2- MIRT (2PL) model is the best fit to test-one data) was applied to the low ability and high ability sample of students separately, to obtain the questions' difficulty and discrimination indices.

3. The question's difficulty and discrimination variance was investigated across the following three groups of students: the whole sample of students, students with low ability, and students with high ability. The standard deviation was measured for the question's difficulty and discrimination across the three groups of students, and then a box plot (Tukey, 1977) of the standard deviations for the question's difficulty and discrimination in each test was drawn. Questions with large standard deviation values compared to other questions in the assessment test are considered outliers as they exhibited high variance across the three groups of students.

Fig 5.6 illustrates the difficulty indices for the 14 questions in test-one after applying the 2-MIRT (2PL)[3] model to the whole sample of students (126 students), low ability

[3] 2-MIRT (2PL) model had the best fit for test-one data.

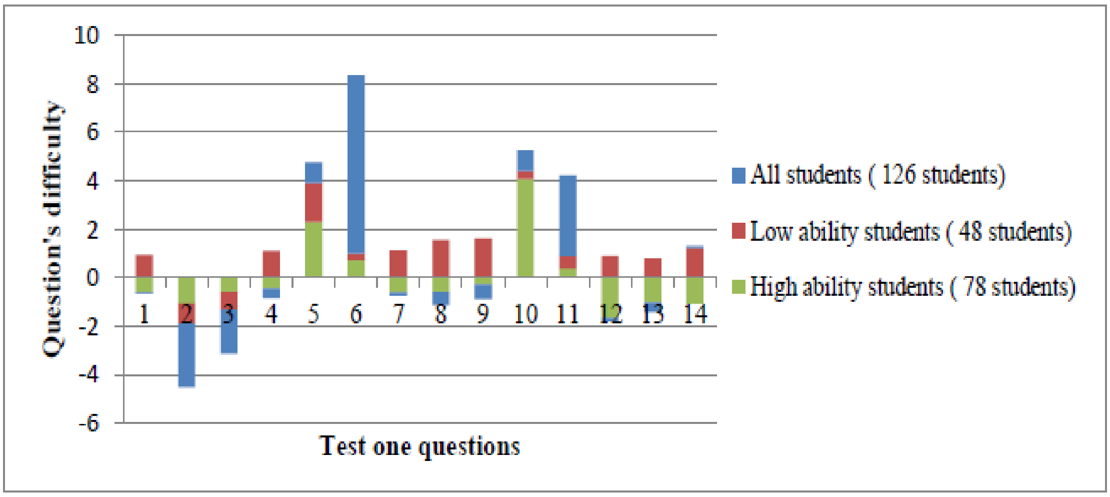


Figure 5.6: Question's difficulty indices obtained using three samples of students

students (48 students), and high ability students (78 students). The blue boxes illustrate the absolute difficulty indices obtained using the 2-MIRT (2PL) model when the whole sample of students is used in the analysis. The red boxes and the green boxes show the absolute difficulty indices obtained using the 2-MIRT (2PL) model when students with low ability and students with high ability are used respectively in the analysis. For example, Fig 5.6 shows that there is a large difference between the difficulty indices of question 6 when different samples of students are used in the analysis. The question has a high difficulty index when the whole sample of students are used (blue box), however, the difficulty indices decrease when low ability (red box) and high ability (green box) samples of students are used in the analysis.

The standard deviation between the difficulty indices for each question was found across the three samples of students and the results revealed that questions 6 and 10 had the highest standard deviation values (outliers) as shown in the box plot in Fig 5.7. The results are expected, as questions 6 and 10 had a large difference in their difficulty indices compared to other question in the test when different samples of students were used in the analysis (see Fig 5.6).

As mentioned earlier, the 2-MIRT (2PL) model has one difficulty index and a discrimination vector with two indices, one in each ability dimension. Fig 5.8 illustrates the questions' discrimination indices in the first dimension (a1) across the three samples of students. As shown in Fig 5.9, questions 3 and 5 had high standard deviation values compared to the other questions in the test. Similar results were obtained for the questions' discrimination indices in the second dimension (a2). Fig 5.10 shows

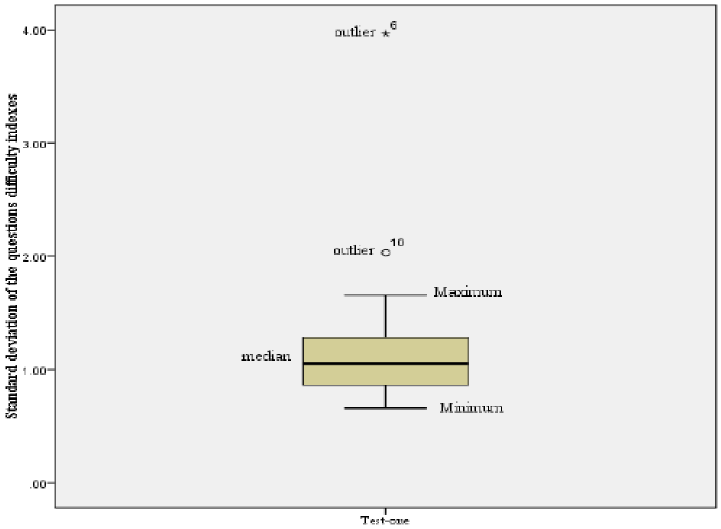


Figure 5.7: Box plot of the difficulty indices standard deviation across the three samples

that questions 3, 5 and 11 have high standard deviation values (see Fig 5.11).

The IRT invariance assumption was valid for 64.28% of questions in test-one, as five questions out of fourteen exhibited variance across the three samples of students. These five questions were removed from the sample of questions used in the upcoming evaluations.

The same methodology was applied to test-two and test-three and the results revealed that the invariance assumption was valid for 87.5% of questions in test-two, and 78.58% of questions in test-three. In total 10 questions out of 44 violated the IRT invariance assumption and were not used in the upcoming evaluations.

Does the Ontology-based Question Generation Strategy Affect the Question's Difficulty and Discrimination?

This section studies the effect of the ontology-based generation strategy (class-based, terminology-based and property-based) on the 44 questions' difficulty and discrimination indices obtained using the CTT, and the 34 questions' (10 questions have been removed) difficulty and discrimination indices obtained using the IRT. Tables 5.23 and 5.24 show the Spearman's rank correlation coefficient between the questions' generation strategies and the measurements of questions' quality results (CTT difficulty index, IRT difficulty index, CTT discrimination index, CTT R_{pb}, and IRT discrimination index).

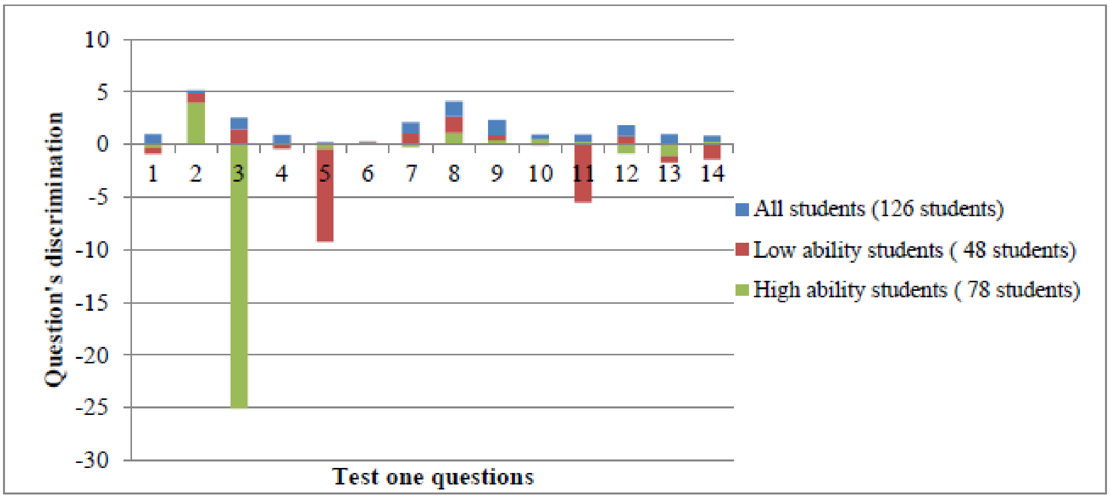


Figure 5.8: Question's discrimination indices (a1) obtained using three samples of students

Table 5.23: Correlation between the generation strategies and the questions' difficulty

	Difficulty	
	CTT difficulty index	IRT difficulty index

Generation strategy	R = -0.405*
	P-value = 0.006
		R = 0.408*
		P-value = 0.017

* P-value < 0.05.

The results revealed a medium correlation between the question generation strategy and most of the measurements of questions' quality results at the 0.05 significance level. However, no statistically significant correlation was found between the question generation strategy and the IRT discrimination indices.

Further investigation was carried out to examine if there was any statistically significant difference between the measurements of questions' quality results generated using the class, terminology, and property-based strategies.

1) Effect on the question's difficulty

The analysis revealed that using different ontology-based generation strategies affects

Table 5.24: Correlation between the generation strategies and the questions' discrimination

	Discrimination		
	CTT discrimination index	CTT Rpb	IRT discrimination index

Generation strategy	R = 0.323*
	P-value = 0.033
		R = 0.357*
		P-value = 0.017
			R = 0.314
			P-value = 0.070

* P-value < 0.05.

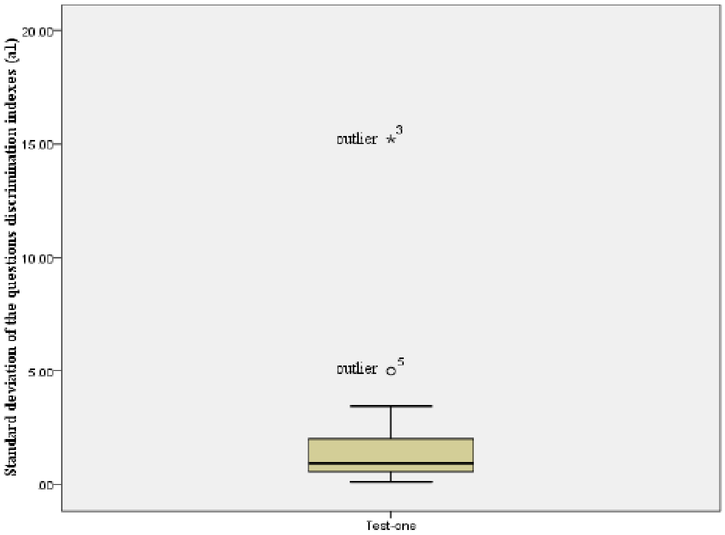


Figure 5.9: Box plot of the discrimination indices (a1) standard deviation across the three samples

the question's difficulty. Assessment questions auto-generated using class, terminology, and property-based generation strategies had a statistically significant difference in the CTT difficulty indices (H(2) = 9.149, P-value = 0.010, P-value < 0.05) and the IRT difficulty indices (H(2) = 11.03, P-value = 0.004, P-value < 0.05). Further analysis revealed a statistically significant difference in the CTT difficulty indices (U = 69, P-value = 0.002, P-value < 0.05) and the IRT difficulty indices (U = 26, P-value = 0.001, P-value < 0.05) between questions generated using the terminology-based strategies and questions generated using the property-based strategies. This suggests that students found questions which assess their knowledge about an educational concept and how it is related to other concepts using the superclass and subclass properties easier than questions which assess their knowledge about the concept's object, datatype, and annotation properties. Questions generated using the terminology-based strategies had higher CTT difficulty indices (Spearman's R = 0.476, P-value = 0.002, P-value < 0.01) and lower IRT difficulty indices (Spearman's R = -0.583, P-value = 5×10^{-4}, P-value < 0.01) compared to questions auto-generated using the property-based strategies. Higher difficulty indices in CTT means the question is easy while in IRT it means the question is more difficult.

No statistically significant difference in the CTT difficulty indices (U = 11, P-value = 0.365, Pvalue > 0.05) and the IRT difficulty indices (U = 2, P-value = 0.145, P-value > 0.05) was found between questions generated using the class-based strategies and questions generated using the terminology-based strategies. Moreover, no statistically

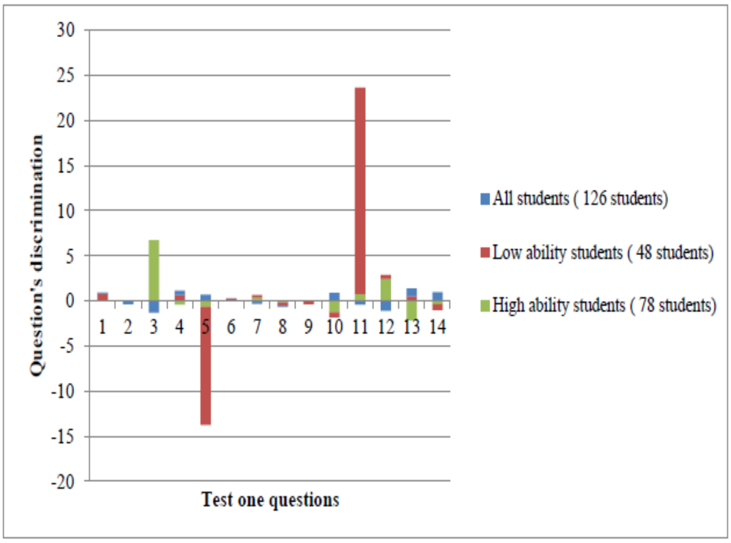


Figure 5.10: Questions' discrimination indices (a2) obtained using three samples of students

significant difference in the CTT difficulty indices (U = 33, P-value = 0.538, P-value > 0.05) and the IRT difficulty indices (U = 18, P-value = 0.667, P-value > 0.05) was found between questions generated using the class-based strategies and questions generated using the property based strategies. This suggests that students found questions auto-generated using the class-based strategy as difficult as questions generated using the terminology-based strategies or the property-based strategies. This suggests that the students found questions auto-generated using the individual and class relationship in the ontology as difficult as questions generated using the terminology-based strategies and the property-based strategies.

2) Effect on the question's discrimination

The analysis revealed that using different generation strategies affect the question's discrimination which was measured using the CTT discrimination index, the CTT R_{pb}, and the IRT discrimination index. The questions had a statistically significant difference in the CTT discrimination indices (H(2) = 9.479, P-value = 0.009, P-value < 0.05), the CTT R_{pb} (H(2) = 12.068, P-value = 0.002, P-value < 0.05), and the IRT discrimination indices (H(2) = 7.117, P-value = 0.028, P-value < 0.05) between the class-based, terminology-based and property-based generation strategies. Further analysis revealed a statistically significant difference in the CTT discrimination indices

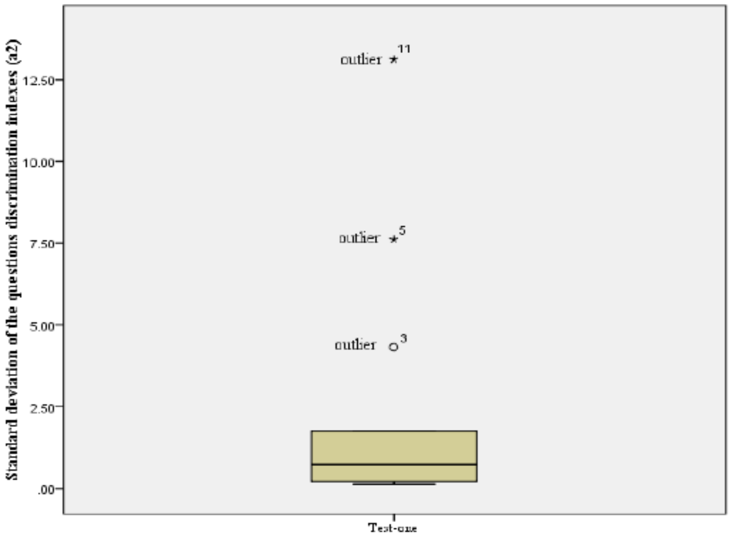


Figure 5.11: Box plot of the discrimination indices (a2) standard deviation across the three samples

(U = 74, P-value = 0.003, P-value < 0.05), the CTT R_{pb} (U = 59, P-value = 0.001, P-value < 0.05), and the IRT discrimination indices (U = 43, P-value = 0.01, P-value < 0.05) between questions auto-generated using the terminology-based strategies and questions auto-generated using the property-based strategies. Questions generated using the property-based strategies had higher CTT discrimination indices (Spearman's R = 0.454, P-value = 0.003, P-value < 0.01), higher CTT R_{pb} (Spearman's R = 0.521, P-value = 5×10^{-4}, P-value < 0.01), and higher IRT discrimination indices (Spearman's R = 0.456, P-value = 0.009, P-value < 0.01) compared to questions generated using the terminology-based strategies.

The results also revealed that there is a statistically significant difference in the CTT discrimination indices (U = 2, P-value = 0.018, P-value < 0.05) and CTT Rpb (U = 2, P-value = 0.018, P-value < 0.05) between questions auto-generated using the class-based strategies and questions auto-generated using the terminology-based strategies. Questions generated using the class-based strategies had higher CTT discrimination indices (Spearman's R = 0.63, P-value = 0.013, P-value < 0.05), and higher CTT Rpb (Spearman's R = 0.617, P-value = 0.014, P-value < 0.05) compared to questions generated using the terminology-based strategies. On the other hand, no statistically significant difference was found in the IRT discrimination indices (U = 2, P-value = 0.145, P-value > 0.05) between questions generated using the class-based strategies and questions generated using the terminology-based strategies. Moreover, no statistically significant difference was found in the CTT discrimination indices (U = 42, P-value = 0.952, P-value $>$ 0.05), CTT R_{pb} (U = 39, P-value = 0.808, P-value $>$ 0.05), and the IRT discrimination indices (U = 19, P-value = 0.733, P-value $>$ 0.05) between questions generated using the class-based strategies and questions generated using the property based strategies. This suggests that the class-based and the property-based strategies produce questions which have similar discrimination indices.

3) Summary

The results of the study can be summarised as follows: generating questions using different generation strategies appeared to affect the question's difficulty and discrimination which were obtained using the CTT and IRT. Questions generated using the terminology-based strategies tend to be easier than questions generated using the property-based strategies, i.e., questions generated using the terminology-based strategies had higher CTT difficulty indices and lower IRT difficulty indices compared to the questions generated using the property-based strategies.

In addition, the results revealed that the property-based strategies generate questions which have better discrimination values compared to questions generated using the terminology-based strategies, i.e., questions generated using the property-based strategies had higher CTT discrimination indices, higher CTT R_{pb} , and higher IRT discrimination indices compared to questions generated using the terminology-based strategies.

Does the Level of the Questions in Bloom's Taxonomy Affect the Question's Difficulty and Discrimination?

Grubisic (Grubisic, 2012; Grubisic et al., 2013), Cubric, and Tosic (Cubric and Tosic, 2011) defined several stem templates (the text stating the question) to auto-generate questions which aim to assess students' cognition at different levels in Bloom's taxonomy (see Section 4.1.3). However, they never investigated if the stem templates order the auto-generated questions according to their easiness in Bloom's taxonomy or if the stem templates affect the question's difficulty and discrimination. Therefore, this section examines the effect of the level of the question in Bloom's taxonomy (Knowledge, comprehension, application, and analysis) on the question's difficulty and discrimination.

Tables 5.25 and 5.26 show the results obtained after applying Spearman's R correlation between Bloom's taxonomy levels and the measurements of questions' quality results. The results revealed a medium correlation between the levels of questions in

Table 5.25: Correlation between Bloom's taxonomy levels and the questions' difficulty

	Difficulty	
	CTT difficulty index	**IRT difficulty index**

Bloom's taxonomy levels	R = -0.268
	P-value = 0.079
		R = 0.357*
		P-value = 0.038

[*] P-value $<$ 0.05.

Table 5.26: Correlation between Bloom's taxonomy levels and the questions' discrimination

	Discrimination		
	CTT discrimination index	**CTT Rpb**	**IRT discrimination index**

Bloom's taxonomy levels	R = 0.419**
	P-value = 0.005
		R = 0.442**
		P-value = 0.003
			R = 0.491**
			P-value = 0.003

[**] P-value $<$ 0.01.

Bloom's taxonomy and the measurements of questions' quality results. However, no statistically significant correlation was found between the CTT difficulty index and the level of the questions in Bloom's taxonomy. Further analysis was carried out to investigate the effect of the level of the question in Bloom's taxonomy on the question's difficulty and discrimination.

1) Effect on the question's difficulty

The results revealed that the level of the questions in Bloom's taxonomy affects the question difficulty, as a statistically significant difference was found in the CTT difficulty indices (H(3) = 9.779, P-value = 0.021, P-value $<$ 0.05) and the IRT difficulty indices (H(3) = 11.772, P-value = 0.008, P-value $<$ 0.05) between different levels in Bloom's taxonomy (knowledge, comprehension, application, and analysis). Further analysis revealed the following:

- A statistically significant difference in the CTT difficulty indices (U = 21, P-value = 0.002, P-value $<$ 0.05) and the IRT difficulty indices (U = 13, P-value = 0.008, P-value $<$ 0.05) between questions in the knowledge level and questions in the comprehension level. Questions generated to assess students in the knowledge level are easier than questions generated to assess students in the comprehension level, as they have higher CTT difficulty indices (Spearman's R = 0.614, P-value = 0.001, P-value $<$ 0.01) and lower IRT difficulty indices (Spearman's R = -0.616, P-value = 0.005, P-value $<$ 0.01) compared to questions generated to assess students in the comprehension level. The results are expected as the knowledge level stem template shown in Table 4.1 focused on assessing whether students could recall concepts and are aware of the subclass and superclass relationships between concepts. However, the comprehension level stem templates focused on students' understanding about the similarity of the relationship between concepts (see question 5 in Table 4.1) and whether students know all the concept's subclasses and superclasses.

- A statistically significant difference in the CTT difficulty indices (U = 36, P-value = 0.039, P-value $<$ 0.05) and the IRT difficulty indices (U = 7, P-value = 0.008, P-value $<$ 0.05) between questions in the knowledge level and questions in the application level. Questions auto-generated to assess students in the knowledge level are easier than questions in the application level, as they have higher CTT difficulty indices (Spearman's R = 0.433, P-value = 0.034, P-value $<$ 0.05) and lower IRT difficulty indices (Spearman's R = -0.670, P-value = 0.005, P-value $<$ 0.01) compared to the application level questions. This is due to the fact that the application level stem templates defined by Grubisic (Grubisic, 2012; Grubisic et al., 2013), Cubric, and Tosic (Cubric and Tosic, 2011) focused on the relationship between the individual and superclass (see question 7 in Table 4.1) as students need to provide an example of the concept he/she learned.

- A statistically significant difference in the IRT difficulty indices between questions in the knowledge level and questions in the analysis level (U = 8, P-value = 0.006, P-value $<$ 0.05). Questions auto-generated to assess students in the Knowledge level are easier than questions auto-generated to assess students in the analysis level, as they had lower IRT difficulty indices compared to the analysis level questions (Spearman's R = -0.674, Pvalue = 0.003, P-value $<$ 0.01). This is due to the fact that the analysis level stem templates defined by Grubisic (Grubisic, 2012; Grubisic et al., 2013), Cubric, and Tosic (Cubric and Tosic, 2011) assess students on the annotation and object properties in classes and individuals (see question 9 in Table 4.1). For students, these stem templates are harder than knowledge level stem templates which focus on recalling concepts in the domain ontology.

In summary, the stem templates defined by Grubisic (Grubisic, 2012; Grubisic et al., 2013), Cubric, and Tosic (Cubric and Tosic, 2011) appear to order the questions according to their easiness in Bloom's taxonomy. Questions in the knowledge level are easier than questions in the comprehension, application and analysis levels. However, no statistically significant difference in the CTT difficulty indices and the IRT difficulty indices was found between questions in the other levels of Bloom's taxonomy. This suggests that the comprehension, application, and analysis level questions appear to have the same difficulty to students.

2) Effect on the question's discrimination

The results revealed that the level of the questions in Bloom's taxonomy affects the question's discrimination, as a statistically significant difference was found in the CTT discrimination indices (H(3) = 12.013, P-value = 0.007, P-value < 0.05), the CTT R_{pb} (H(3) = 13.491, P-value = 0.004, P-value < 0.05), and the IRT discrimination indices (H(3)= 8.422, P-value = 0.038, P-value > 0.05) between the different levels of questions in Bloom's taxonomy. Further analysis revealed a statistically significant difference in the CTT discrimination indices (U = 3, P-value =11×10^{-5}, P-value $<$ 0.05), CTT R_{pb} (U = 4, P-value = 19×10^{-5}, P-value < 0.05) and the IRT discrimination indices (U = 8.5, P-value = 0.006, P-value < 0.01) between the knowledge level and the analysis level questions. Questions auto-generated to assess students on the analysis level had higher CTT discrimination indices (Spearman's R = 0.801, P-value = 22×10^{-6}, P-value < 0.01), higher CTT R_{pb} (Spearman's R = 0.779, P-value =5×10^{-5}, P-value < 0.01), and higher IRT discrimination indices (Spearman's R = 0.674, P-value = 0.003, P-value < 0.01) compared to the knowledge level questions.

The results also revealed no statistically significant difference in the IRT discrimination indices between the other levels of questions in Bloom's taxonomy. On the other hand, the results revealed a statistically significant difference in the CTT discrimination indices (U = 26.5, P-value = 0.007, P-value < 0.05) and CTT Rpb (U = 26, P-value = 0.007, P-value < 0.05) between questions auto-generated to assess students on the knowledge level and questions auto-generated to assess students on the comprehension level. Comprehension level questions had higher CTT discrimination indices (Spearman's R = 0.550, P-value = 0.005, P-value < 0.01) and higher CTT R_{pb} (Spearman's R = 0.554, P-value = 0.005, P-value < 0.01) compared to the knowledge level questions. The results also revealed a statistically significant difference in CTT R_{pb} (U = 31, P-value = 0.017, P-value < 0.05) between questions in the knowledge level and questions in the application level. Application level question had higher CTT discrimination indices compared to knowledge level questions (Spearman's R = 0.494, P-value = 0.014, P-value < 0.05).

3) Summary

Table 5.27: Summary of the level of the question effect on the questions' difficulty

Level of the question in Bloom's taxonomy	Difficulty	
	CTT difficulty index	IRT difficulty index
Knowledge (K)	K >C	C >K
Comprehension (C)	-----	-----
Application (App)	K >App	App >K
Analysis (Ana)	-----	Ana >K

Table 5.28: Summary of the level of the question effect on the questions' discrimination

Level of the question in Bloom's taxonomy	Discrimination		
	CTT discrimination index	CTT Rpb	IRT discrimination index
Knowledge (K)	C >K	C >K	-----
Comprehension (C)	-----	-----	-----
Application (App)	-----	App >K	-----
Analysis (Ana)	Ana >K	Ana >K	Ana >K

Tables 5.27 and 5.28 summarise the effect of the level of the question in Bloom's taxonomy on the question's difficulty and discrimination. The results revealed that questions in the knowledge level were easier than questions in the comprehension, application, and analysis levels. However, no statistically significant difference in the CTT difficulty indices and the IRT difficulty indices was found between the other levels in Bloom's taxonomy. This suggests that the comprehension, application, and analysis level questions appeared to have the same difficulty to students. Questions' discrimination was also investigated and the results revealed that the knowledge level questions which are the easiest questions tend to have lower discrimination compared to the comprehension, application and analysis level questions. On the other hand, no statistically significant difference in the CTT discrimination indices, the CTT R_{pb}, and the IRT discrimination indices was found between the comprehension, application, and analysis level questions. This suggests that the comprehension, application and analysis stem templates auto-generate questions which have the same discrimination.

Does the Type of Question Affect the Question's Difficulty and Discrimination?

Grubisic (Grubisic, 2012; Grubisic et al., 2013) generated true and false, multiple choice, and short answer questions, while Cubric, and Tosic (Cubric and Tosic, 2011) focused on auto-generating multiple choice questions. This section studies the effect of the type of question (true and false, multiple choice, and short answer) on the question's difficulty and discrimination. Tables 5.29 and 5.30 show the results obtained after applying Spearman's R correlation between the type of questions and the question's difficulty and discrimination. The results revealed that the type of questions had

Table 5.29: Correlation between the types of questions and the questions' difficulty

	Difficulty	
	CTT difficulty index	**IRT difficulty index**

Type of questions	R = -0.396 *
	P-value = 0.008
		R = 0.518*
		P-value = 0.002

* P-value $<$ 0.05.

Table 5.30: Correlation between the types of questions and the questions' discrimination

	Discrimination		
	CTT discrimination index	**CTT Rpb**	**IRT discrimination index**

Type of questions	R = 0.505*
	P-value = 47×10^{-5}
		R = 0.543 *
		P-value = 14×10^{-5}
			R = 0.431*
			P-value = 0.011

* P-value $<$ 0.05.

a medium correlation with the CTT difficulty index and the IRT discrimination index, and a strong correlation with the CTT discrimination index, the CTT R_{pb}, and the IRT difficulty index.

1) Effect on the question's difficulty

The results revealed that the type of auto-generated questions affects the question difficulty, as a statistically significant difference was found in the CTT difficulty indices (H(2) = 9.103, P-value = 0.011, P-value $<$ 0.05) and IRT difficulty indices (H(2) = 10.96, P-value = 0.004, P-value $<$ 0.05) between true and false, multiple choice and short answer questions. Further analysis revealed a statistically significant difference in the CTT difficulty indices (U = 69, P-value = 0.001, P-value $<$ 0.05) and the IRT discrimination indices (U = 25, P-value = 5×10^{-4}, P-value $<$ 0.05) between true and false questions and multiple choice questions (MCQs). True and false questions are easier than MCQs as they have higher CTT difficulty indices (Spearman's R = 0.483, P-value = 0.001, P-value $<$ 0.01) and lower IRT difficulty indices (Spearman's R = 0.591, P-value = 4×10^{-4}, P-value $<$ 0.01) compared to MCQ.

No statistically significant difference was found in the questions' difficulty between the other types of questions.

2) Effect on the question's discrimination

The type of auto-generated question affects the question discrimination, as a statistically significant difference was found in the CTT discrimination indices (H(2) = 11.004, P-value = 0.004, P-value $<$ 0.05), the CTT R_{pb} (H(2) = 12.726, P-value = 0.002, P-value $<$ 0.05) and the IRT discrimination indices (H(2) = 6.947, P-value =

Table 5.31: Summary of the type of question effect on the questions' difficulty

Type of question	**Difficulty**	
	CTT difficulty index	**IRT difficulty index**
True and false (T/F)	T/F >MCQ	-----
MCQ	-----	MCQ >T/F
Short answer (SA)	-----	-----

0.031, P-value < 0.05) between different types of questions. Further analysis revealed a statistically significant difference in the CTT discrimination indices (U = 76, P-value = 0.001, P-value < 0.05), CTT R_{pb} (U = 61, P-value = 0.001, P-value < 0.05), and IRT discrimination indices (U = 43, P-value = 0.01, P-value < 0.05) between true and false questions and MCQs. MCQs had higher CTT discrimination indices (Spearman's R = 0.454, P-value = 0.003, P-value < 0.01), higher CTT R_{pb} (Spearman's R = 0.517, P-value = 45×10^{-5}, P-value < 0.01), and higher IRT discrimination indices (Spearman's R = 0.456, P-value = 0.009, P-value < 0.01) compared to true and false questions.

No statistically significant difference in IRT discrimination indices was found between the other types of questions. However, a statistically significant difference was found in the CTT discrimination indices (U = 0, P-value = 0.022, P-value < 0.05) and CTT R_{pb} (U = 0, P-value = 0.022, P-value < 0.05) between true and false and short answer questions. Short answer questions had higher discrimination indices (Spearman's R = 0. 0.617, P-value = 0.019, P-value < 0.05) and higher CTT R_{pb} (Spearman's R = 0.608, P-value = 0.021, P-value < 0.05) compared to true and false questions.

No statistically significant difference in CTT discrimination indices and CTT R_{pb} was found between the other types of questions.

3) Summary

The results of the study are shown in Tables 5.31 and 5.32, and can be summarised as follows: the type of question affects the question's difficulty and discrimination. True and false questions were easier than MCQs. However, MCQs had higher discrimination indices compared to true and false questions. This is proved by the CTT discrimination index, CTT R_{pb}, and IRT discrimination index. Moreover the CTT discrimination index and the CTT R_{pb} revealed that short answer question had higher discrimination compared to true and false questions.

Table 5.32: Summary of the type of question effect on the questions' discrimination

Type of question	Discrimination		
	CTT discrimination index	CTT Rpb	IRT discrimination index
True and false (T/F)	-----	-----	-----
MCQ	MCQ >T/F	MCQ >T/F	MCQ >T/F
Short answer (SA)	SA >T/F	SA >T/F	-----

5.4 Chapter Summary

The quantitative analysis carried out in this chapter complements the qualitative analysis carried out by the ontology-based question generators (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014). This chapter presented the experiment carried out to achieve Contribution 4 by assessing the following:

1. The quality of tests and questions auto-generated using Papasalouros's (Papasalouros et al., 2008; Papasalouros et al., 2011), Grubisic's (Grubisic, 2012; Grubisic et al., 2013), Al-Yahya's (Al-Yahya, 2011; Al-Yahya, 2014), Cubric's, and Tosic's (Cubric and Tosic, 2011) generation strategies.

2. The effect of the ontology-based generation strategies, the level of the questions in Bloom's taxonomy and the type of questions on the questions' quality measurements.

The Classical Test Theory (CTT) was used to examine the difficulty and discrimination of the three tests which consisted of 44 questions. The results obtained using the CTT revealed that the three tests had medium difficulty values which are very close to the value that the test authors are advised to achieve when constructing tests. In addition, the results revealed that the questions and tests have satisfactory discrimination values, which indicate that the questions could effectively discriminate between high ability and low ability students, and that the questions may not need to be reviewed or eliminated from the assessment tests.

In addition to the CTT, the Item Response Theory (IRT) was used to assess the quality of auto-generated questions. The IRT has the invariant assumption, which states that questions' difficulty and discrimination are invariant across different samples of students. The IRT revealed similar results to the CTT, as 22.7% (10 questions out of 44) of the questions were very easy or very difficult, which results in low discriminating questions. In addition, the questions' discrimination indices had positive values which justify that the auto-generated questions may not need to be reviewed or eliminated from the assessment tests.

The results obtained using the CTT and IRT had also been used to study the effect of the ontologybased generation strategies, the level of the questions in Bloom's taxonomy, and types of questions on the questions' quality measurements. The results revealed that the generation strategies, the level of the questions in Bloom's taxonomy, and the type of questions affect the questions' difficulty and discrimination. This provides guidance for developers and researchers working in the field of ontology-based question generators. Moreover, the analysis results could be used in future work to predict question's difficulty and discrimination based on the generation strategy, the level of the question in Bloom's taxonomy, and the type of question. This will help researchers and developers save time and effort in terms of testing the auto-generated questions on real students.

After analysing the quality of tests and questions in this chapter, the next chapter will present Contribution 5. The chapter will explain the experimental study carried out to evaluate the effect of feedback on students' performance and learning gain, and analyse the quality of feedback associated with the auto-generated questions.

Chapter 6

Evaluation of the Ontology-based Personalised Feedback Generator

This chapter evaluates the Ontology-based Personalised Feedback Generator (OntoPeFeGe) and contributes to the research carried out in the personalised feedback frameworks (Mason and Bruning, 2001; Narciss et al., 2014; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010) and the ontology-based formative feedback generators (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Duboc et al., 2011; Duboc, 2013; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011) by achieving the following:

1. Examine the effect of personalised feedback which adopts Mason and Bruning's theoretical framework (Mason and Bruning, 2001) on students' performance and learning gain. The types of feedback provided to students are generated from a domain ontology and are adapted based on the student's characteristics (background knowledge and current level of knowledge) and the task's characteristics (the level of each question in Bloom's taxonomy (Bloom et al., 1956)).

2. Study the relationship between student's characteristics (background knowledge, gender), the task's characteristics (the level of each question in Bloom's taxonomy (Bloom et al., 1956)) and the personalised feedback, and how they affect students' performance and learning gain.

3. Observe students' satisfaction regarding the usefulness of the ontology-based auto-generated feedback, the feedback's clarity, and the feedback's impact on answering other questions in the test.

4. Examine teachers' satisfaction regarding the clarity of the auto-generated questions and feedback, the pedagogical content provided in each type of feedback, and the usefulness of the auto-generation process in terms of saving time and effort.

The OntoPeFeGe evaluation carried out in this chapter complements the evaluations conducted in Chapter 5 which analysed the quality of the auto-generated tests and questions. As according to Black and William effective formative assessment is achieved by ensuring the quality of assessment questions and feedback (Black and Wiliam, 1998).

6.1 Experiment Overview

This section illustrates the experiment questions and participants and describes the experimental set-up used to evaluate the OntoPeFeGe. Moreover, it explains the experiment design, the assessment questions, and the questionnaires used in the evaluation.

6.1.1 Experiment Questions

The experiment was designed to achieve Contribution 5 by answering the following questions:

Q1: Does the personalised feedback affect students' performance and learning gain, and how does the effect of personalised feedback differ from Knowledge Of Results (KOR) feedback?

Q2: Does the personalised feedback affect students' performance and learning gain differently when provided to students after answering questions designed to assess them at different levels in Bloom's taxonomy, and does the effect of personalised feedback differ from KOR feedback?

Q3: Does the personalised feedback affect students with low and high background knowledge differently, and how the effect of personalised feedback differs from KOR feedback?

Q4: Does students' gender have an impact on how the personalised feedback affects their performance and learning gain?

Q5: Are students satisfied with the ontology-based auto-generated feedback's usefulness and clarity, and do they agree that the feedback helped them answer other questions in the test?

Q6: Are the domain experts (teachers) satisfied with the clarity of the ontology-based auto-generated questions and feedback, the pedagogical content provided in each type of feedback, and the usefulness of the auto-generation process in terms of saving time and effort?

6.1.2 Participants

In 2013/2014, eighty-eight (69 males, 19 females) second and third year undergraduate students registered in the Data Networking course (of Manchester, 2013c) and the Computer Networks course (of Manchester, 2013a) at the University of Manchester volunteered to take part in the experiment. Students' identities were kept anonymous.

6.1.3 Experimental Set-up

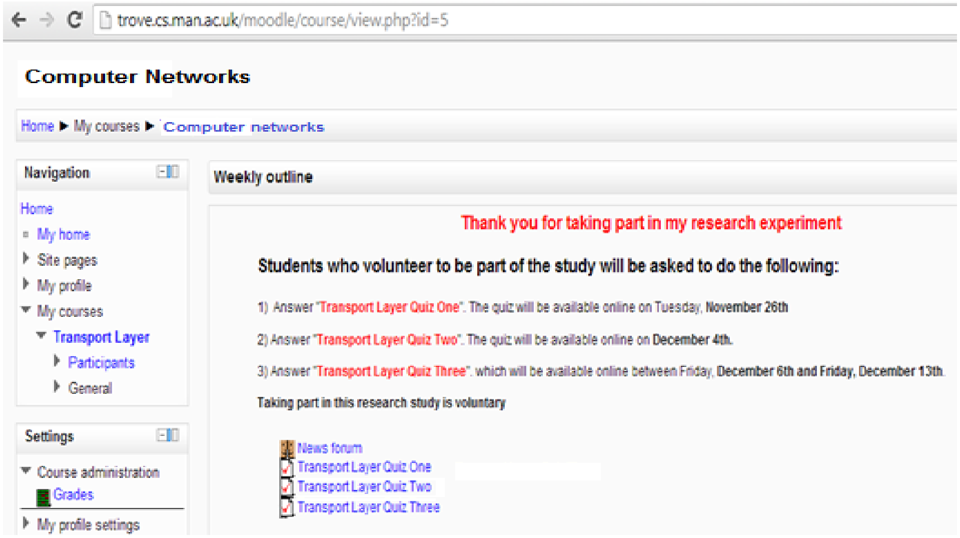
The experiment was carried out in the Moodle Virtual Learning Environment (VLE) (of Manchester, 2013b) in a course called 'Computer Networks', which was created for the purpose of this experiment. The course is shown in Fig 6.1 and it consists of the following three tests which assessed students' knowledge of the transport layer topic:

1. Transport layer test one which is called the pre-test.
2. Transport layer test two which is called the treatment-test.
3. Transport layer test three which is called the post-test.

Students who volunteered to take part in the experiment received an instruction document in Moodle VLE to help them understand how the experiment will be performed and what they are asked to do in details. A copy of the instruction document is in Appendix G.

6.1.4 Experiment Design

This study uses the pre-test/treatment/post-test design (Cohen et al., 2013) shown in Fig 6.2. Students were asked first to answer the pre-test, and the test scores were used to allocate them randomly to the experimental group (40 males, 8 females) and the



Thank you for taking part in my research experiment

Students who volunteer to be part of the study will be asked to do the following:

1) Answer "Transport Layer Quiz One". The quiz will be available online on Tuesday, November 26th

2) Answer "Transport Layer Quiz Two". The quiz will be available online on December 4th.

3) Answer "Transport Layer Quiz Three". which will be available online between Friday, December 6th and Friday, December 13th.

Taking part in this research study is voluntary

Figure 6.1: Computer Networks course in Moodle

control group (29 males, 11 females) using the matched pairs design approach (Cohen et al., 2013). The basis for allocation is matching each member of the experimental group to a member of the control group based on their pre-test scores (background knowledge). This prevents having an unbalanced assignment of students with similar background knowledge in the same group.

Ensuring the background knowledge similarity between the experimental and control groups was proved statistically by running the independent sample t-test between the experimental and control groups pre-test scores. The t-test results showed no statistically significant difference in the pre-test scores between the experimental group (M = 55.06, SD = 17.98) and the control group (M = 55, SD = 14.84), t(86) = 0.017, P-value = 0.987, P-value $>$ 0.05).

In the treatment phase shown in Fig 6.2, students in the experimental group received personalised feedback after answering each question, while students in the control group received KOR feedback. The KOR feedback was chosen because it is the default type of feedback auto-generated to students after answering true and false, multiple choice and short answer questions in VLEs (e.g., Moodle and Blackboard). Moreover, KOR provides students with the lowest level of information (correct or incorrect) compared with other types of feedback.

After receiving personalised and KOR feedback in the treatment phase students were asked to answer the post-test.

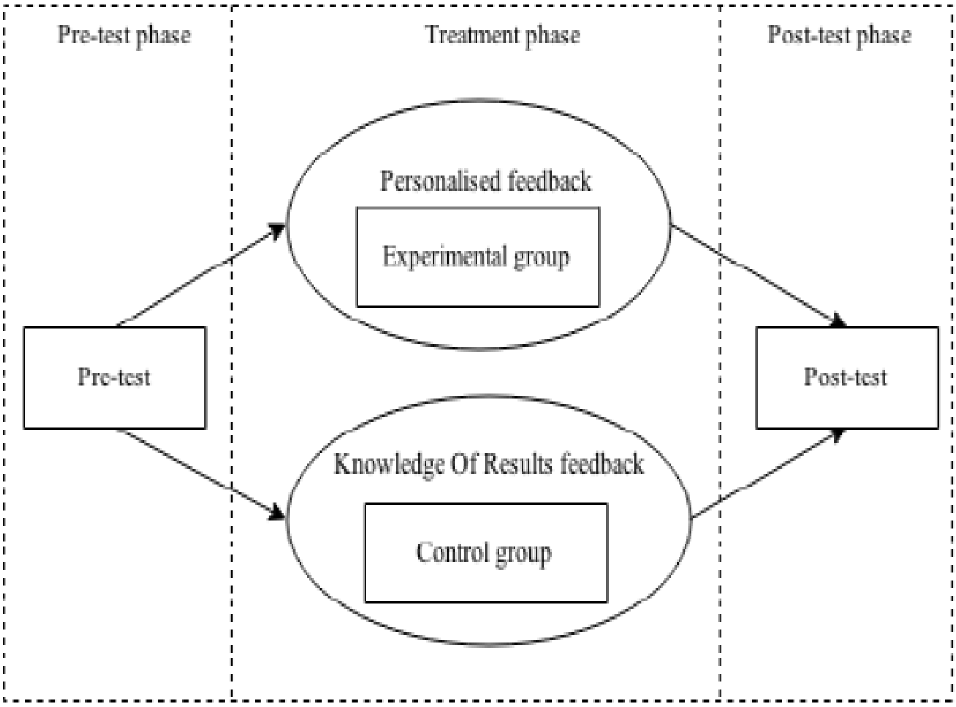


Figure 6.2: Experiment design

6.1.5 Assessment tests

Fig 6.1 shows that the experiment consisted of three tests. The tests are not identical (the assessment questions in each test were different) but have similar structure; i.e., the tests consisted of questions assessing students' cognition at four levels in Bloom's taxonomy (knowledge, comprehension, application, and analysis). See Section 5.1.3 in Chapter 5, page 123. The three tests used in the experiment can be found in Appendix D.

6.1.6 Questionnaires

Two questionnaires were created one for students and another for the teachers. The students' questionnaire aimed at assessing their satisfaction with the usefulness and clarity of the ontology-based auto-generated feedback (see Appendix H). The teachers' questionnaire aimed to assess their satisfaction about the clarity of the ontology-based auto-generated questions, the pedagogical content provided in each type of feedback, and the usefulness of the auto-generation process in terms of saving time and effort (see Appendix I).

6.2 Results and Discussion

This section answers the experiment questions given in Section 6.1.1.

Q1: Does the personalised feedback affect students' performance and learning gain, and how does the effect of personalised feedback differ from KOR feedback?

Kazi *et al.* (Kazi et al., 2013), and Sã nchez-Vera *et al.* (Frutos-Morales et al., 2010; S et al., 2012) evaluated the effect of the KOR, Knowledge of Correct Response (KCR), and hint ontology-based generated feedback on students' performance (see Section 2.3.3). Their results revealed that ontology-based feedback improved students' performance significantly. On the other hand, Duboc *et al.* (Duboc et al., 2011; Duboc, 2013) whose system auto-generated more types of formative feedback (KCR, Bugs-Related (BR), and Topic Contingent (TC))[1] did not evaluate the effect of different types of feedback on students performance, and confined his evaluation to students' satisfaction regarding the importance and usefulness of the auto-generated types of feedback. Therefore, this question aims to:

1. Assess the effect of personalised feedback (KOR, KCR, BR, TC, and Response Contingent (RC)) which is auto-generated from the domain ontology on students' performance and learning gain. The effect of the auto-generated BR, TC, and RC types of feedback on students' performance had not been examined before.

2. Examine if the personalised feedback which adopts Mason and Bruning's theoretical framework improves students' performance similarly to Narciss *et al.* and Arroyo *et al.* (Narciss and Huth, 2004; Narciss et al., 2014; Arroyo et al., 2011; Woolf et al., 2010) personalised feedback frameworks (see Table 2.1 in Chapter 2, page 17).

3. Examine if the personalised feedback has a different effect on students' performance and learning gain compared to KOR feedback.

Fig 6.3 illustrates students' pre-test and post-test performance mean scores in the experimental group and control group plus or minus one standard error (”standard error is a measure of how representative the sample is likely to be of the population” (Field, 2013)). The t-test results revealed no statistically significant difference in the overall

[1] See Section 2.1 for more details on the different types of feedback.

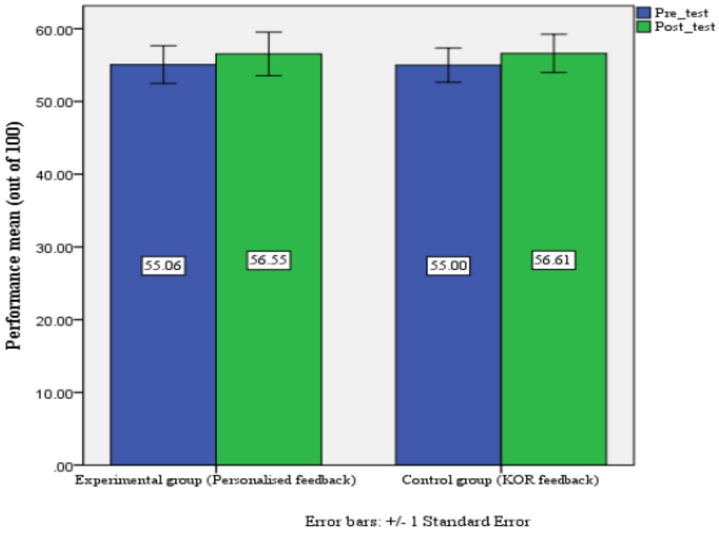


Figure 6.3: Experimental and control groups overall performance at the pre-test and the post-test

post-test performance between the experimental group (M = 56.547, SD = 20.72) and control group (M = 56.607, SD = 16.684), t(86) = 0.015, P-value = 0.988, P-value $>$ 0.05) after receiving personalised or KOR feedback, and both groups had approximately similar post-test performance scores. Fig 6.3 also shows an increase in students' post-test performance after receiving the personalised feedback and KOR feedback. However, no statistically significant difference was found between the pre-test and post-test performance scores for the experimental group (t(47) = -0.555, P-value = 0.582, P-value $>$ 0.05), and control group (t(39) = -0.714, P-value = 0.479, P-value $>$ 0.05). This suggests that students' performance did not change from the pre-test to the post-test after receiving personalised feedback and KOR feedback.

The learning gain was also measured for students in the experimental group and control group using the following equation (Fancher, 2013; Stanek III, 1988):

$$\text{Learning Gain} = \text{Post-test score} - \text{Pre-test score} \tag{6.1}$$

Fig 6.4 illustrates the learning gain distribution for students in the experimental and control groups. The Mann-Whitney U-test was used to examine the significant difference in students learning gain between the experimental group and the control group. The results revealed that students in the experimental group had slightly higher learning gain compared to students in the control group. However, no statistically significant difference in learning gain was found between the experimental and control groups (U

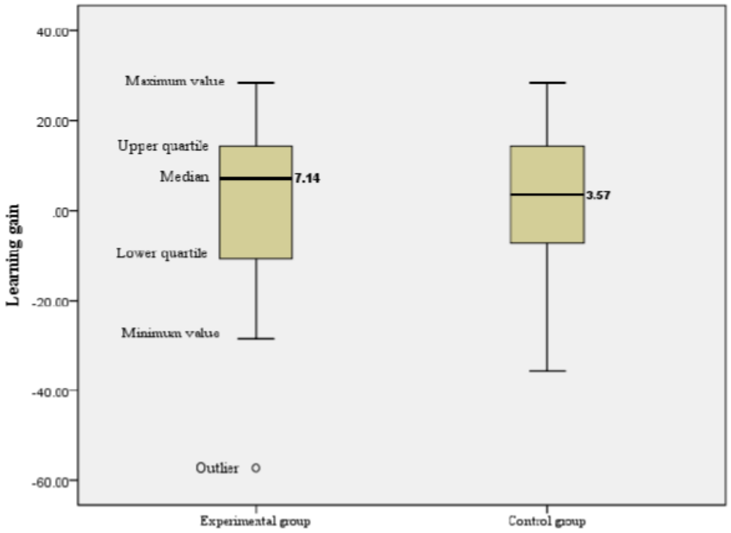


Figure 6.4: Learning gain for students in the experimental and control groups

= 937, P-value = 0.847, P-value $>$ 0.05).

Further investigation was carried out to examine the effect of personalised feedback on students' performance and learning gain. Section 4.2 shows that the personalised feedback algorithm provides students with the following three initial types of feedback (F0) based on their background knowledge, current level of knowledge and the task's characteristics (see Fig 4.14 in Chapter 4, page 119):

1. F0 = Bugs Related (BR) feedback, which is provided to the students when their background knowledge is high and their current level of knowledge is high. Twenty-one students in the experimental group received F0 = BR.

2. F0 = Topic Contingent (TC) feedback, which is provided to the students when their background knowledge is high and their current level of knowledge is low. Four students in the experimental group received F0 = TC.

3. F0 = Response Contingent (RC) feedback, which is provided to the students when their background knowledge is low regardless of their current level of knowledge. Twenty-three students in the experimental group received F0 = RC.

Fig 6.5 illustrates the pre-test and post-test performance distribution for students who received personalised feedback with different initial types and students who received KOR feedback.

To investigate which personalised feedback (F0=BR, F0=TC, F0=RC, KOR) increased students' performance significantly from the pre-test to the post-test, Wilcoxon's

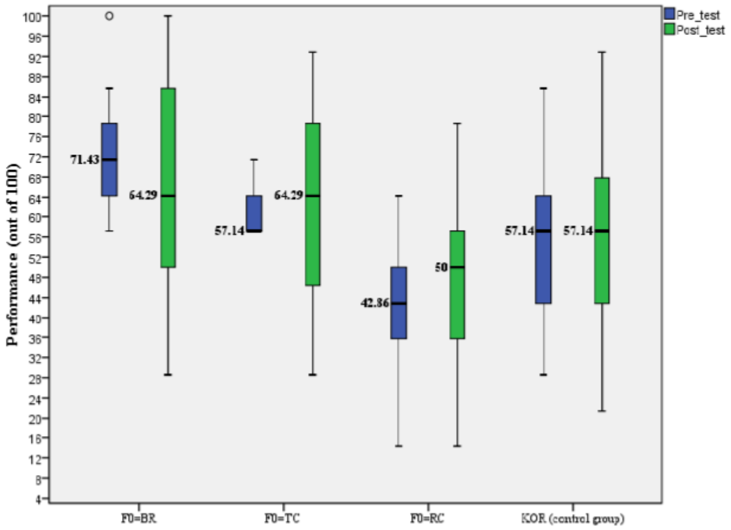


Figure 6.5: Experimental group performance after receiving different initial feedback

signed-rank test was performed between the pre-test and post-test performance scores. The results revealed that the personalised feedback which starts by providing students with the highest level of detail (F0 = RC) increased students' performance significantly (Z = -1.989, P-value = 0.047, P-value < 0.05). However, no statistically significant difference was found between students' pre-test performance and post-test performance after receiving personalised feedback with F0 = TC and F0 = BR and KOR feedback. Fig 6.5 shows that students' performance decreased slightly after receiving personalised feedback with F0 = BR while their performance increased slightly after receiving personalised feedback with F0 = TC. On the other hand, students' performance remained the same after receiving the KOR feedback.

Further investigation revealed that students who received personalised feedback with F0 = RC tend to agree more that the feedback helped them answer other questions in the assessment tests compared to students who received personalised feedback with F0 = TC and F0 = BR and KOR feedback. In addition, the investigation revealed that students who received KOR feedback tend to have lower agreement regarding the usefulness and clarity of feedback compared to students who received personalised feedback (personalised feedback with F0 = BR, F0 = TC, and F0 = RC).

Students' learning gain was also examined and the results revealed no statistically significant difference in learning gain between students in the four different groups (F0 = BR, F0 = TC, F0 = RC, and KOR).

In summary, the personalised feedback which considered the student and task's characteristics only improved students performance significantly when F0 = RC and no statistically significant difference in learning gain was found between students who received personalised feedback and students who received KOR feedback.

Q2: Does the personalised feedback affect students' performance and learning gain differently when provided to students after answering questions designed to assess them at different levels in Bloom's taxonomy, and does the effect of personalised feedback differ from KOR feedback?

The OntoPeFeGe adopts Mason and Bruning's personalised feedback theoretical framework (Mason and Bruning, 2001) which considered both student's characteristics (background knowledge, current level of knowledge) and the task's characteristics (level of assessment question in Bloom's taxonomy). None of the personalised feedback frameworks in Section 2.1 adapted the different types of feedback based on the task's characteristics or studied the relationship between the personalised feedback and the task's characteristics. Therefore, this question aims to examine students' performance and learning gain after receiving the personalised feedback associated with questions designed to assess students at each level of Bloom's taxonomy. Moreover, the effect of the personalised feedback is compared to KOR feedback.

The following represent the analysis carried out on questions designed to assess students at different levels in Bloom's taxonomy.

1) Questions designed to assess students at the knowledge level

Students in the experimental and control groups answered twelve ontology-based auto-generated questions (four question in each of the pre-test, treatment-test and post-test) which aimed to assess them at the knowledge level in Bloom's taxonomy (see Table 5.3). Fig 6.6 illustrates the pre-test and post-test performance distribution of knowledge level questions. The figure shows that 75% of students had high performance scores (above 50) in the pre-test and post-test after receiving the personalised feedback (experimental group) and KOR feedback (control group). This suggests that most students found the knowledge level questions easy. Fig 6.6 also shows that students in both groups had close post-test performance scores and no statistically significant difference was found between their post-test performance scores (U = 901, P-value = 0.602, P-value $>$ 0.05). However, more students in the control group (50%) had post-test performance scores above 75 compared to students in the experimental group (25%).

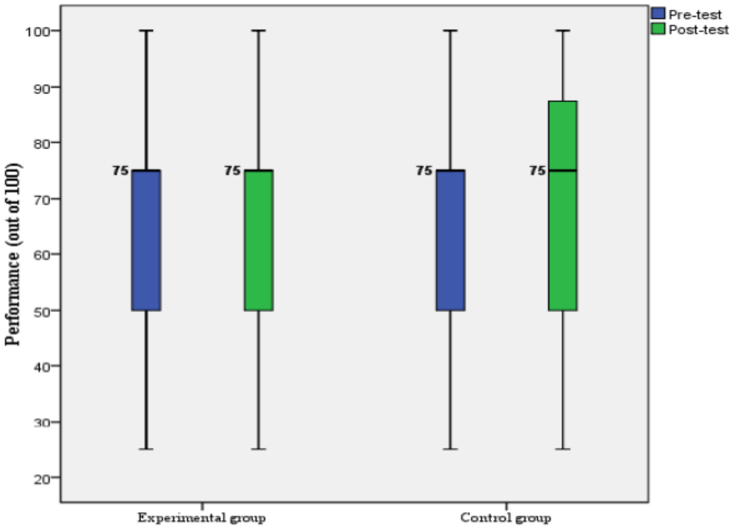


Figure 6.6: Experimental and control groups performance at knowledge level questions

The results also revealed that students' performance did not change from the pre-test to the posttest after receiving the personalised feedback (Z = -0.331, P-value = 0.741, P-value $>$ 0.05) and KOR feedback (Z = -0.696, P-value = 0.486, P-value $>$ 0.05). Moreover, no statistically significant difference in learning gain (see Fig 6.7) was found between students in the experimental group and control group (U = 887, P-value = 0.521, P-value $>$ 0.05). However, Fig 6.7 shows that 25% of students in the experimental group had lower learning gain compared to students in the control group.

Further investigation was carried out to examine if the personalised feedback with several F0 had different effects on students' performance and learning gain. The results revealed that the personalised feedback with F0 = BR, F0 = TC, or F0 = RC provided to students after answering the knowledge level questions had no statistically significant effect on their performance and learning gain.

2) Questions designed to assess students at the comprehension level

Students tend to find the comprehension level questions more difficult compared to the knowledge level questions, as 50% of students had pre-test and post-test performance scores below 50 (see Fig 6.8). However, similar to the knowledge level question students in the experimental group and control group had approximately the same post-test performance scores (U = 911.5, P-value = 0.676, P-value $>$ 0.05) and their performance from the pre-test to the post-test did not change after receiving personalised feedback (Z = -0.641, P-value = 0.522, P-value $>$ 0.05) and KOR feedback (Z =

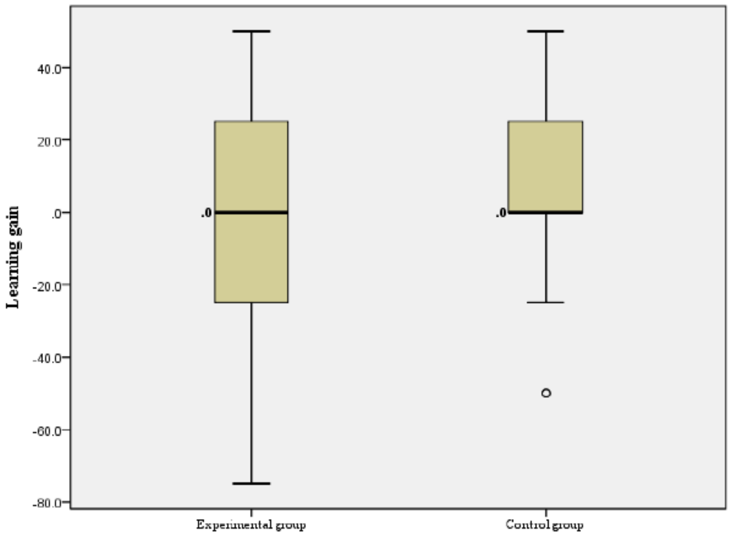


Figure 6.7: Experimental and control groups learning gain at knowledge level questions

-0.361, P-value = 0.718, P-value > 0.05). This suggests that the personalised feedback has no statistically significant effect on students' performance at questions assessing the comprehension level.

Students learning gain was also examined (see Fig 6.9) and the results revealed no statistically significant difference between the experimental and control groups (U = 946, P-value = 0.905, P-value > 0.05).

Similar to the knowledge level questions, further analysis was carried on the comprehension level questions to investigate if the personalised feedback with several F0 had different effects on students' performance and learning gain. The results revealed that the personalised feedback with several F0 has no statistically significant effect on students' performance and learning gain.

3) Questions designed to assess students at the application level

Fig 6.10 shows that 50% of students in the experimental and control groups had pre-test and post-test performance scores below 50. This suggests that students found application level questions more difficult compared to the knowledge level questions where 75% of students had pre-test and post-test performance scores above 50 (see Fig 6.6).

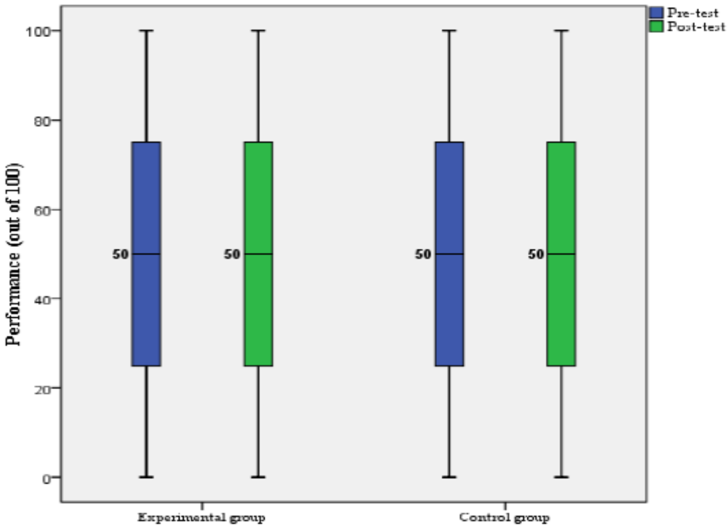


Figure 6.8: Experimental and control groups performance at comprehension level questions

Analysing the effect of personalised feedback and KOR feedback on students' performance revealed that the personalised feedback did not improve students' performance significantly (no statistically significant difference was found in students' performance between the pre-test and post-test). However, the KOR feedback improved students' performance significantly from the pretest to the post-test (Z = -2.495, P-value = 0.013, P-value $<$ 0.05). This suggests that providing students with KOR feedback after answering application level questions is more efficient to improve their performance compared to the personalised feedback where more information is provided to students in the feedback content. Further investigation revealed that students who had an increase in their performance after receiving the KOR feedback reported that the feedback was clear to understand their misconceptions.

Students post-test performance and learning gain in the experimental and control groups were also investigated and the results revealed no statistically significant difference in the post-test performance (U = 763.5, P-value = 0.08, P-value $>$ 0.05) and learning gain (U = 844, P-value = 0.308, P-value $>$ 0.05) between both groups (see Fig 6.11).

The results also revealed no statistically significant difference in students' performance and learning gain after receiving personalised feedback with different F0.

4) Questions designed to assess students at the analysis level

Fig 6.12 shows that students who received personalised feedback after answering the

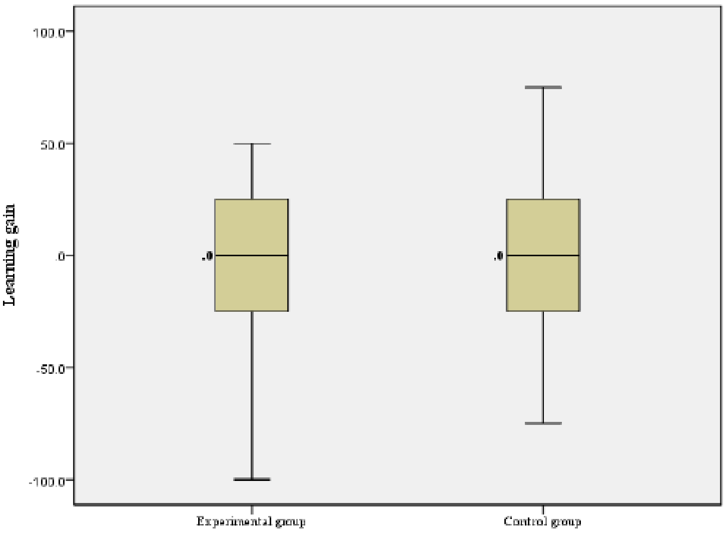


Figure 6.9: Experimental and control groups learning gain at comprehension level questions

analysis level questions had no change in their performance from the pre-test to the post-test. However, students in the control group had a statistically significant decrease in their performance after receiving KOR feedback (Z = -2.202, P-value = 0.028, P-value < 0.05). This suggests that the KOR feedback provided to students after answering the analysis level questions is not sufficient to improve their performance. Further investigation revealed that students who had a decrease in their performance reported that the KOR feedback was not clear to understand their misconceptions. The difference between students learning gain in the experimental and control groups was also investigated. Fig 6.13 shows that students who received personalised feedback had a significant increase in their learning gain (U = 689.5, P-value = 0.017, Spearman's R = 0.3, P-value = 0.016, P-value < 0.05) compared to students who received KOR feedback at questions assessing the analysis level, as 50% of students had learning gain above zero in the experimental group compared to 25% of students in the control group. This result suggests that students benefited more from the personalised feedback at questions assessing the analysis level in Bloom's taxonomy.

Further analysis was carried out to investigate the effect of personalised feedback with several F0 on students' performance. The results revealed no statistically significant difference between the different feedback (F0 = BR, F0 = TC, and F0 = RC) on students post-test performance. However, a significant difference in learning gain (see Fig 6.14) was found between students who received personalised feedback with F0 =

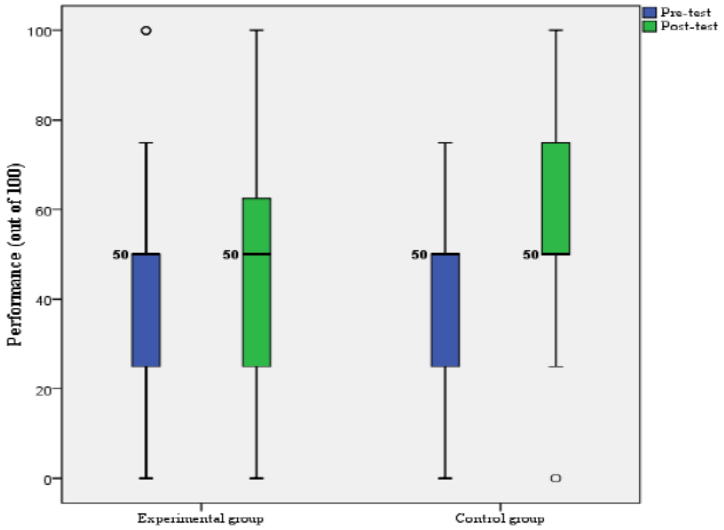


Figure 6.10: Experimental and control groups performance at application level questions

RC and students who received KOR feedback at questions assessing Bloom's taxonomy analysis level (U = 302.5, P-value = 0.018, P-value $<$ 0.05, Spearman's R = 0.3, P-value = 0.017, P-value $<$ 0.05). Students who received personalised feedback with F0= RC had higher learning gain compared to students who received KOR feedback.

In summary, the results revealed that both the personalised feedback and KOR feedback have the same effect on students' performance and learning gain when provided to students after they answered questions designed to assess them at the knowledge and comprehension levels in Bloom's taxonomy. However, the effect of personalised feedback and KOR feedback on students' performance and learning gain differed for questions designed to assess students at the application and analysis levels in Bloom's taxonomy. While the personalised feedback had no statistically significant effect on students' performance and learning gain for questions designed to assess students at the application level, KOR feedback improved students' performance significantly. On the other hand, the personalised feedback improved students' learning gain significantly at questions designed to assess students at the analysis level in Bloom's taxonomy and KOR feedback decreased their performance significantly at the analysis level questions.

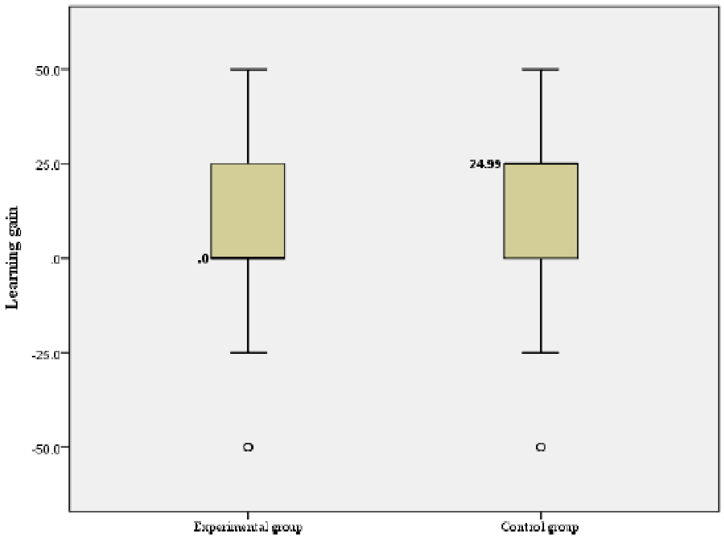


Figure 6.11: Experimental and control groups learning gain at application level questions

Q3: Does personalised feedback affect students with low and high background knowledge differently, and does the effect of personalised feedback differ from KOR feedback?

This question aims to examine the effect of personalised feedback, which tailors the different types of feedback based on the student's and the task's characteristics, on the performance and learning gain of students with low and high background knowledge. The question was partially answered by Arroyo *et al.* (Arroyo et al., 2011; Woolf et al., 2010). They studied the relationship between students' background knowledge and the personalised feedback, which tailored the types of feedback based on student's current level of knowledge. Arroyo *et al.* (Arroyo et al., 2011; Woolf et al., 2010) results revealed that students with low background knowledge had more improvement in their performance compared to students with high background knowledge. Similar results are obtained in this experiment. Fig 6.15 shows the pre-test and post-test performance distribution for low background knowledge students (pre-test score < 50) and high background knowledge students (pre-test score ≥ 50) after receiving personalised feedback (experimental group) and KOR feedback (control group). The results revealed that low background knowledge students had a significant increase in their performance after receiving the personalised feedback (Z = -1.989, P-value = 0.047, P-value < 0.05). On the contrary, students with low background knowledge in the control group who received KOR feedback had no difference in their performance (Z

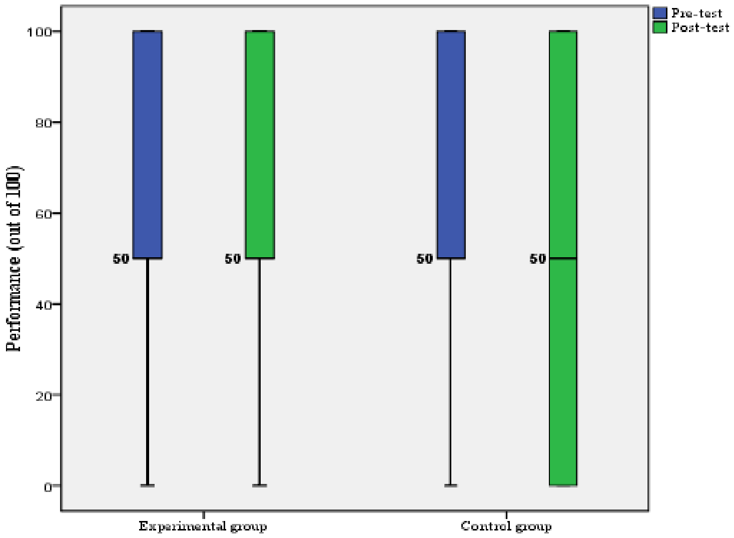


Figure 6.12: Experimental and control groups performance at analysis level questions

= -1.574, P-value = 0.116, P-value > 0.05). This suggests that students with low background knowledge benefit more from the personalised feedback compared to KOR feedback. The results are consistent with the results obtained by Arroyo *et al.* (Arroyo et al., 2011; Woolf et al., 2010). Moreover, the results comply with Black and William's findings in (Black and Wiliam, 1998).

The results also revealed that high background knowledge students had no statistically significant difference in their performance after receiving personalised feedback or KOR feedback (see Fig 6.15).

Students' learning gain was also investigated. Fig 6.16 shows a bar chart, which illustrates the learning gain for low background knowledge and high background knowledge students after receiving the personalised feedback and KOR feedback. The figure shows that students with low background knowledge had positive learning gain after receiving the personalised feedback and KOR feedback. On the other hand, students with high background knowledge tend to have negative learning gain regardless of the feedback received. No statistically significant difference in learning gain was found between students with high and low background knowledge in the experimental and control groups.

Q4: Does students' gender have an impact on how the personalised feedback affects their performance and learning gain?

Several personalised feedback frameworks, which adapt the types of feedback based on students' current level of knowledge (see Table 2.1) studied the relationship

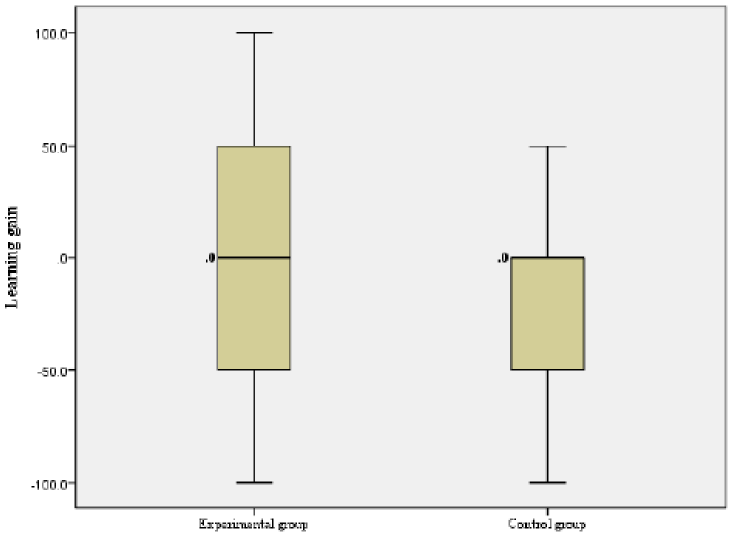


Figure 6.13: Experimental and control groups learning gain at analysis level questions

between students' gender, performance, and the personalised feedback, to examine if gender should be considered as an essential factor in feedback personalisation (Narciss et al., 2014; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010). Narciss *et al.* (Narciss et al., 2014) showed that female students had higher performance than male students after receiving personalised feedback. Arroyo *et al.* had similar results in (Arroyo et al., 2000; Arroyo et al., 2001), however, in another study they carried out in (Arroyo et al., 2011; Woolf et al., 2010) no statistically significant difference in performance was found between male and female students, and both genders had an increase in their performance after receiving personalised feedback. Based on their findings, Arroyo *et al.* concluded that it is still too early to establish clear principles regarding the issue of how to tailor the feedback content specifically to male and female students (Arroyo et al., 2011). On the other hand, Narciss *et al.* (Narciss et al., 2014) suggested that the personalised feedback frameworks should consider students' gender when they are used in the mathematical domain. This question aims to answer the following:

1. Investigates if the personalised feedback affects the performance and learning gain (overall and at each level in Bloom's taxonomy) of male and female students differently, and how the personalised feedback effect differs from KOR feedback

2. Examine if Mason and Bruning's theoretical framework which is adopted in OntoPeFeGe should consider students gender in the personalised feedback algorithm.

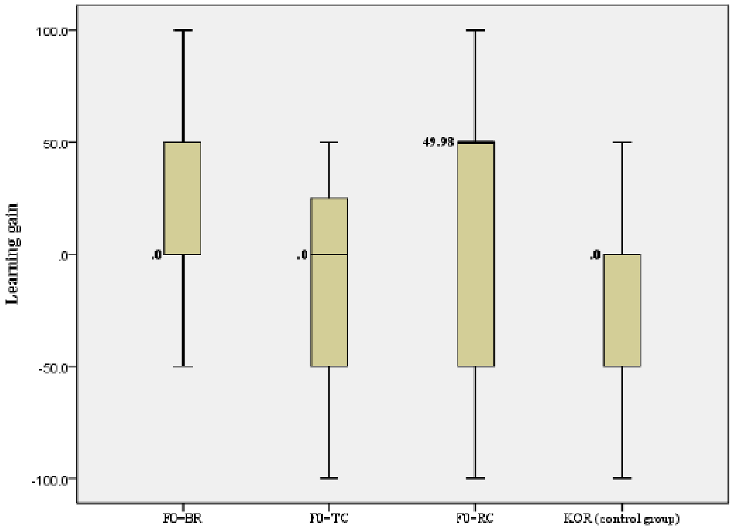


Figure 6.14: Learning gain at the analysis level after receiving different initial feedback

The analysis revealed no statistically significant difference between male and female students' performance after receiving the personalised feedback and KOR feedback (see Fig 6.17).

The results also revealed no statistically significant difference in the overall learning gain between male and female students after receiving the personalised feedback (experimental group) and KOR feedback (control group). However, both male and female students had an increase in their learning gain after receiving the personalised feedback and KOR feedback. Fig 6.18 shows that female students had higher overall learning gain (4.46) compared to male students (0.89) after receiving the personalised feedback. Similarly, female students in the control group also had higher overall learning gain (3.90) compared to male students (0.74).

Further statistical analysis was carried out to investigate if female students had a significantly higher learning gain compared to male students at questions designed to assess the knowledge, comprehension, application, and analysis levels in Bloom's taxonomy. The results revealed no statistically significant difference between male and female students after receiving the personalised feedback and KOR feedback. However, Fig 6.18 shows that the learning gain differs between male and female students after receiving the personalised feedback and KOR feedback at questions assessing several levels in Bloom's taxonomy. The results could be summarised as follows:

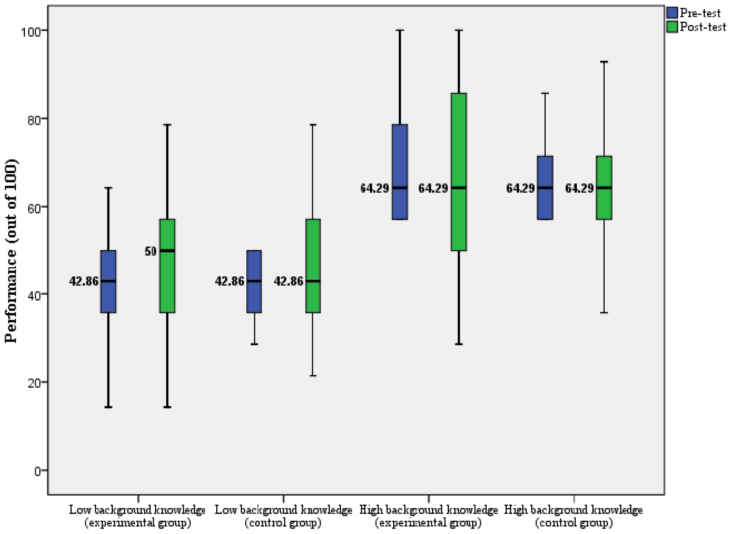


Figure 6.15: Performance of students with different background knowledge

1. Knowledge level questions: Female students had higher learning gain (12.50) after receiving the personalised feedback compared to male students who had negative learning gain (-4.37). However, female students learning gain was lower (2.27) than male students learning gain (2.59) after receiving KOR feedback.

2. Comprehension level questions: Male (-5.0) and female (-3.12) students had negative learning gain after receiving the personalised feedback. On the contrary, female students (11.36) had higher learning gain than male students (-6.89) after receiving KOR feedback.

3. Application level questions: Both male and female students had positive learning gain after receiving the personalised feedback and KOR feedback. Moreover, male students had higher learning gain compared to female students.

4. Analysis level questions: Male and female students benefited more from the personalised feedback compared to KOR feedback. Fig 6.18 shows that male and female students had positive learning gain when they received personalised feedback and negative learning gain when they received KOR feedback. Moreover, male students had higher learning gain compared to female students after receiving both personalised and KOR feedback.

In summary, the results revealed no statistically significant difference in performance and learning gain between male and female students after receiving the personalised feedback. The results are similar to the results obtained by Arroyo *et al.* in (Arroyo

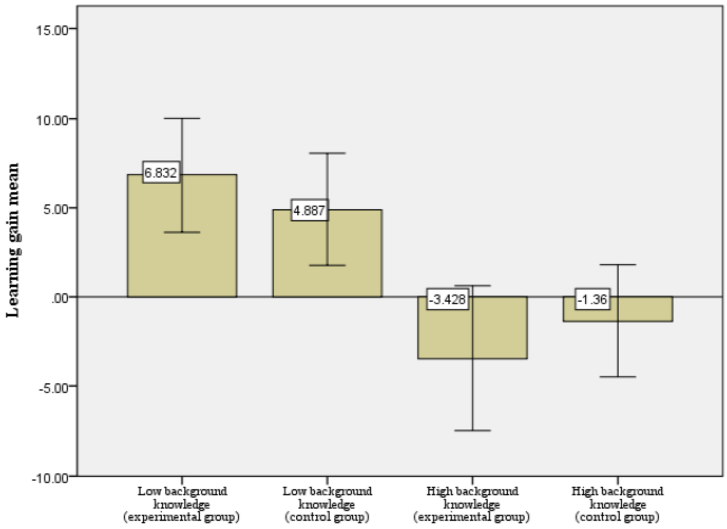


Figure 6.16: learning gain for students with different background knowledge

et al., 2011; Woolf et al., 2010) and this may suggest that it is not essential to consider gender as an important factor in Mason and Bruning's personalised feedback framework. However, the analysis was only carried out in the computer networks domain and there was a big difference between the number of male and female students in the experiment (40 males and 8 females in the experimental group, and 29 males and 11 females in the control group). Further investigation could be carried out across different educational domains and using a larger sample of male and female students.

Q5: Are students satisfied with the ontology-based auto-generated feedback's usefulness and clarity, and do they agree that the feedback helped them answer other questions in the test?

Students in the experimental group (48 students) received ontology-based auto-generated feedback while students in the control group (40 students) received KOR auto-generated feedback. Students in the experimental and control groups answered a questionnaire which aimed to assess if the students understand the formative feedback and are willing and able to act on it (Price et al., 2010). The questionnaire assessed students' satisfaction regarding the feedback's usefulness, clarity, and whether the feedback helped them answer other questions in the test.

The questionnaire had three questions scored on a 3-point Likert scale (agree, neutral, disagree), and is shown in Appendix H. Fig 6.19 and Fig 6.20 illustrate the experimental and control groups responses to the questionnaire. The results revealed that

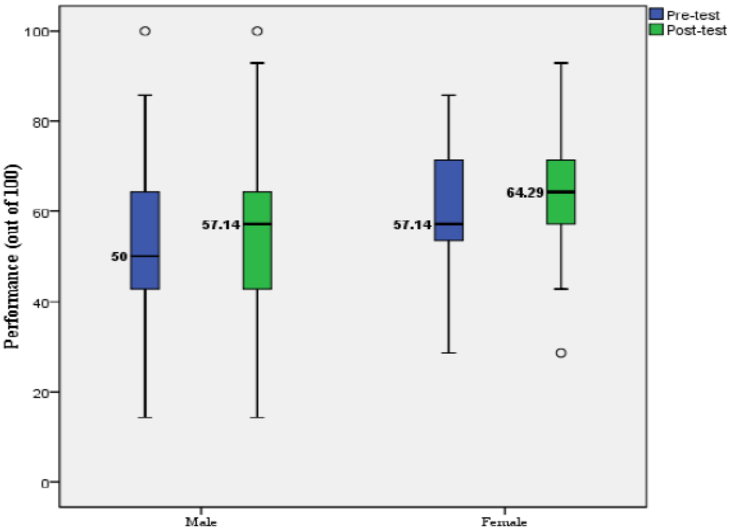


Figure 6.17: Performance of male and female students at the pre-test and the post-test

students in the experimental and control groups had similar responses regardless of the formative feedback provided to them. 72.92% of the students in the experimental group agreed that the feedback is useful, 70.83% agreed that the generated feedback was easy to read, and 68.75% agreed that the formative feedback provided in Moodle VLE helped them in answering some of the following questions in the assessment tests.

The results are consistent with the evaluation results obtained by the ontology-based formative feedback generators in Section 2.3.3 where students accepted the auto-generated feedback and agreed that it was useful (Kazi et al., 2012; Kazi et al., 2010; Duboc, 2013; Frutos-Morales et al., 2010; S et al., 2012).

Fig 6.20 shows that students in the control group who received KOR also found the feedback useful (70%), easy to read (65%), and helpful in terms of answering some of the upcoming questions in the tests (72.5%).

Fig 6.19 and Fig 6.20 also show that approximately one-third of students in the experimental and control groups were not satisfied with the formative feedback provided. Further investigation was carried out to investigate the correlation between students' responses to each question in the questionnaire and their background knowledge, post-test performance and the change in their performance from the pre-test to the post-test (increase, decrease, no change).

1) Feedback is useful

The results revealed no correlation between students (experimental group and control

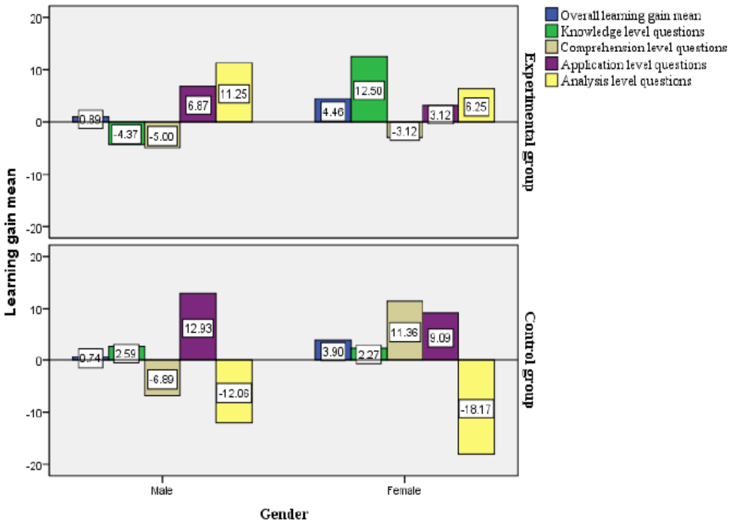


Figure 6.18: Learning gain mean for male and female students

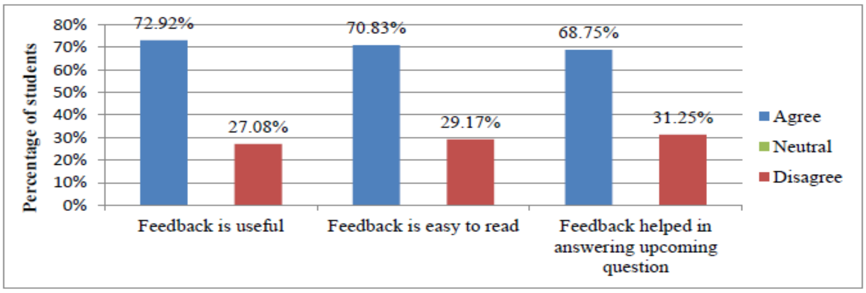


Figure 6.19: OntoPeFeGe feedback evaluated by students (Experimental group)

group) responses to the first question 'feedback is useful' and their background knowledge, post-test performance, and the change in their performance from the pre-test to the post-test.

2) Feedback is easy to read

Students (experimental group and control group) responses to the 'feedback is easy to read' question had no correlation with their background knowledge and post-test performance. However, the percentage of students in the experimental group who had an improvement in their performance (from the pre-test to the post-test) and agreed that the formative feedback was easy to read (64.7%) was higher than the percentage of student who agreed that the feedback was easy to read and had no improvement (5.9%) or decrease (29.4%) in their performance (Spearman's R = 0.378, P-value = 0.008, P-value $<$ 0.01). Moreover, the percentage of students in the experimental group who had a decrease in their performance and disagreed that the feedback was

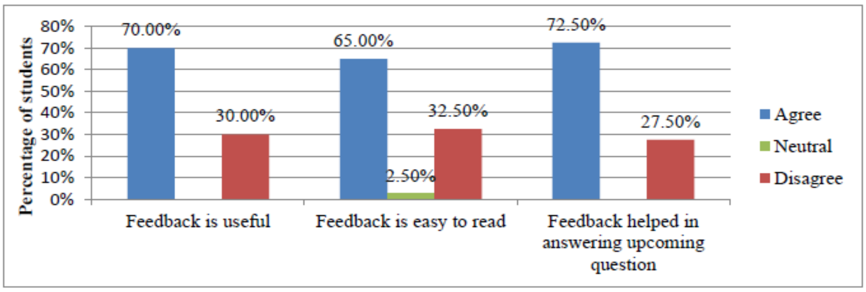


Figure 6.20: KOR feedback evaluated by students (Control group)

easy to read (64.3%) was high compared to students who had no effect (14.3%) or improvement in their performance (21.4%).

No correlation was found between students responses in the control group and the change in their performance.

3) Feedback helped in answering some of the upcoming questions in the tests

93.8% of students in the experimental group with low background knowledge (pre-test performance < 50) agreed that the formative feedback helped them answer some of the upcoming questions in the tests, compared to 56.3% of students with high background knowledge (Spearman's R = 0.381, P-value = 0.007, P-value < 0.01). Moreover, Students in the experimental group with post-test performance below 50 agreed that the formative feedback helped them answer some of the upcoming question in the test while students with post-test performance above 50 disagreed (Spearman's R = 0.358, P-value = 0.013, P-value < 0.05). These results are consistent with the results obtained in both Study 1 (see Section A.1 in Appendix A, page 201) and Bedford and Price's study (Bedford and Price, 2007; Demaidi, 2014). Both studies revealed that students with high performance scores disagreed that the feedback was helpful.

No correlation was found between the background knowledge of students in the control group and their responses to the 'feedback helped in answering some of the upcoming questions in the test' question. Moreover, no correlation was found between students' (experimental group and control group) responses to the question and the change in their performance.

Fig. 6.19 and Fig. 6.20 show that students' agreement percentage in the experimntal group was the lowest compared to other questions. However, in the control group it was the highest. This may be due to the fact that the phrasing of the question was not clear and influenced the results. Students may have interpreted the question as if the feedback helped them have the correct answer in the tests instead of improving their learning about the topic.

Q6: Are the domain experts (teachers) satisfied with the clarity of the ontology-based auto-generated questions and feedback, the pedagogical content provided in each type of feedback, and the usefulness of the auto-generation process in terms of saving time and effort?

The ontology-based auto-generated questions and feedback were evaluated by three domain experts (teachers). One domain expert was a computer networks lecturer at the School of Electrical and Electronic Engineering, University of Manchester and the other two domain experts were specialists in Virtual Learning Environments, however, they do not teach courses related to computer networks. The three experts accessed the ontology-based auto-generated tests in Moodle VLE in order to evaluate the auto-generated questions and formative feedback by answering a 5-point Likert scale (1: strongly disagree, to 5: strongly agree) questionnaire presented in Appendix I . The following represents the domain experts' responses to each question in the questionnaire:

1) The generated questions are clear and easy to read

The assessment questions in OntoPeFeGe are auto-generated using the stem templates defined by Cubric and Tosic (Cubric and Tosic, 2011) and Grubisic (Grubisic, 2012; Grubisic et al., 2013) (see Table 4.1 in Chapter 4, page 98). Cubric and Tosic did not publish any evaluation results. On the contrary, Grubisic evaluated the ontology-based generated questions using two groups of students (Grubisic et al., 2013). 21% of students in the first group found the questions comprehensible while 29% had a neutral opinion, and 50% found the questions incomprehensible. On the other hand, 38% of students in the second group found the questions comprehensible, 38% had a neutral opinion, and 19% found the questions incomprehensible (see Section 2.3.3). Grubisic concluded that students in the second group who were more mature and had more experience working with different VLEs were more satisfied with the ontology-based generated questions.

In this experiment, the three domain experts investigated the ontology-based auto-generated questions and agreed (the average ranking score is 4.34) that the ontology-based auto-generated questions used in the three tests are clear and easy to read.

2) The generated formative feedback is clear and easy to read

The domain experts agreed (the average ranking score is 4.0) that the ontology-based formative feedback provided to students is clear and easy to read.

3) The feedback pedagogical content is reasonable and related to the question

This question aims to assess if the ontology-based auto-generated feedback is related to the question students are asked to answer. The domain experts agreed (the average ranking score is 4.34) that the feedback pedagogical content (right/wrong, the correct answer, the reason why the correct answer is correct, and the reason why the incorrect answer is incorrect) is reasonable and related to the question.

4) The feedback is providing students with different information at different level of detail

This question aims to assess if the domain experts agree that the ontology-based auto-generated feedback is providing students with different types of feedback. The results revealed that the domain experts agreed (the average ranking score is 3.67) that the generated feedback provides students with different information at different levels of detail.

5) The feedback auto-generation saves time in terms of writing different feedback content to students with different background knowledge and performance

The domain experts agreed (the average ranking score is 4.67) that the feedback auto-generation saves time in terms of writing different feedback content to students with different background knowledge and performance.

6) The feedback auto-generation saves time in terms of writing different feedback content for different expected answers

The domain experts agreed (the average ranking score is 4.0) that the feedback auto-generation saves time in terms of writing different feedback content for different expected answers.

7) The ontology-based auto-generated feedback provided to students is useful

The domain experts agreed (the average ranking score is 3.67) that the feedback provided to students is useful.

6.3 Chapter Summary

This chapter evaluated the Ontology-based Personalised Feedback Generator (OntoPeFeGe) and contributed to the research carried in the personalised feedback frameworks (Mason and Bruning, 2001; Narciss et al., 2014; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010) and the ontology-based formative feedback generators (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Duboc et al., 2011; Duboc, 2013; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011) by addressing the six research questions listed in Section 6.1.1. An experiment was carried out in Moodle VLE on eighty-eight students (69 males, 19 females) who were divided into the experimental group and control group. Students in the experimental group received personalised feedback while students in the control group received KOR feedback.

The results revealed that the personalised feedback only improved students performance when the initial type of feedback (F0) provided to students was Response Contingent (RC), and no statistically significant difference in learning gain was found between students who received personalised and students who received KOR feedback (Q1). Moreover, the results revealed that the personalised feedback and KOR feedback had the same effect on students' performance and learning gain at questions designed to assess students at the knowledge and comprehension levels, and a different effect at questions designed to assess students at the application and analysis levels. While the personalised feedback had no statistically significant effect on students' performance and learning gain at questions designed to assess students at the application level, KOR feedback improved students' performance significantly. On the other hand, the personalised feedback improved students' learning gain significantly at questions designed to assess them at the analysis level in Bloom's taxonomy and KOR feedback decreased their performance significantly (Q2).

This chapter also investigated if low background knowledge students benefit more from the personalised feedback (Q3). The results obtained were similar to the results reported by Arroyo et al. (Arroyo et al., 2011; Woolf et al., 2010). Students with low background knowledge benefited more from the personalised feedback compared to students with high background knowledge. Moreover, students with low background knowledge had an increase in their performance after receiving the personalised feedback while students who received KOR feedback had no change in their performance.

The personalised feedback in OntoPeFeGe adopted Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001) which did not consider students' gender. An investigation was carried to examine if gender is an important factor in feedback personalisation (Q4). The results revealed no statistically significant difference in performance and learning gain between male and female students after receiving the personalised feedback and KOR feedback. However, further investigation should be carried out in the future as the number of female students in the experiment was very low compared to the number of male students.

This chapter also evaluated the students (Q5) and teachers' (Q6) satisfaction regarding the ontology-based auto-generated feedback. The results revealed that 72.92% of the students in the experimental group agreed that the feedback is useful, 70.83% agreed that the feedback generated was easy to read, and 68.75% agreed that the formative feedback helped them answer other questions in the assessment tests. However, students in the control group had approximately similar results and one-third of students in both groups were not satisfied. Therefore, further investigation was carried out and the results revealed that students in the experimental group who had an improvement in their performance tend to agree more that the feedback was easy to read. Moreover, students in the experimental group with low background knowledge tend to agree more than students with high background knowledge that the feedback helped in answering some of the upcoming questions in the tests.

The teachers (three domain experts) were also satisfied with the ontology-based auto-generated questions and feedback as they agreed that the questions and feedback were easy to read and that the OntoPeFeGe provides students with different types of feedback. Moreover, they agreed that the feedback's pedagogical content is reasonable and related to the auto-generated question.

Chapter 7

Conclusion and Future Directions

The thesis concludes with a review of the work presented, the significance of the major contributions, and directions of future work.

7.1 Thesis Overview

The work presented in this thesis is motivated by the clear limitation in existing personalised feedback frameworks which are domain dependent, i.e., the different types of feedback are either hard-coded or auto-generated from a restricted set of solutions defined by the teacher or the domain expert (Ana and Macario, 2009; Narciss and Huth, 2004; Narciss et al., 2014; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010). Furthermore, Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001), which adapts the different types of feedback based on the student and the task (question) characteristics was never evaluated on students even though the task characteristics were considered as important factors in the process of personalising feedback (Narciss et al., 2014). Therefore, the primary aim of this thesis was to propose a novel, generic framework which achieves the following:

1. Selects a domain ontology for auto-generating assessment questions and feedback in a specific educational field: Evaluate the candidate domain ontologies which capture the concepts in an educational course using the educational course corpus (e.g., textbooks and lecture slides), and select the candidate domain ontology which has high coverage and semantic richness of the educational corpus.

2. Uses domain ontologies to auto-generate domain independent questions with different characteristics. In particular, questions which assess students at the

knowledge, comprehension, application, and analysis levels in Bloom's taxonomy.

3. Associates each question's option with different types of feedback which are auto-generated from the domain ontology.
4. Provides students with personalised feedback immediately after answering an assessment question. The personalised feedback adapts the type of feedback based on the student and the task characteristics.

7.2 Significance of Major Contributions

The hypotheses set out in Section 1.1 have been successfully fulfilled by the thesis contributions described below:

Contribution 1: Terminological ONtology Evaluator (TONE)

The domain ontology evaluation metrics (e.g., coverage and semantic richness) highly depend on the list of terms extracted from the educational corpus. An increase in the number of irrelevant terms may results in selecting an inappropriate or less suitable candidate domain ontology and consequently generating questions and feedback which are not related to the domain of interest. Therefore, it is essential to consider the weight (reflects how important the term is to the domain of interest) associated with terms extracted from the educational corpus. Most of the terminological data-driven ontology evaluation approaches in the literature ignored the terms' weight except Rospocher (Rospocher et al., 2012), and Jones and Alani's approaches (Jones and Alani, 2006). However, both approaches used an arbitrary cut-off point in the list of terms extracted from the corpus.

In Chapter 3, a Terminological Ontology Evaluator (TONE) was developed which dynamically set a cut-off point in the list of terms extracted from the corpus using the back of textbook index (gold-standard list). TONE implements two approaches to dynamically select a cut-off point and reduce the number of irrelevant terms: TONE-R which selects the best relative cut-off point from a set of relative cut-off points and TONE-A which select the best absolute cut-off point from a set of absolute cut-off points.

Two experiments were carried out to evaluate TONE-R and TONE-A. The first experiment aimed to investigate the validity of TONE-R and TONE-A for selecting the best candidate domain ontology (candidate ontology which has the highest coverage and semantic richness) for generating questions and feedback. The results revealed that TONE-R and TONE-A were able to distinguish between domain ontologies related to the domain of interest and domain ontologies unrelated to the domain of interest.

The second experiment compared TONE-R and TONE-A with Rospocher (Rospocher et al., 2012), and Jones and Alani's (Jones and Alani, 2006) terminological data-driven approaches. The experiment investigated the variance of the ontology evaluation metrics and the agreement between the rankings of the candidate domain ontologies when the lists of terms used to evaluate the ontologies were extracted from the corpus using different term extraction tools and assigned weight using several term recognition algorithms. The results revealed that TONE-R and TONE-A had lower variance in the Semantic Richness Score compared to the other terminological data-driven approaches and higher agreement compared to Rospocher's approach. Moreover, TONE-R and TONE-A had lower variance and higher agreement in the F-measure Scores compared to Rospocher's approach. This helps the user obtain approximately similar ontology evaluation metrics regardless of the list of terms extracted from the corpus.

On the other hand, Jones and Alani's approach which only considered the top fifty terms in the lists of terms extracted from the corpus had lower variance in the Coverage Precision and F-measure scores compared to the other data-driven approaches and 100% agreement in the coverage and semantic richness rankings. However, using TONE-R and TONE-A to select the best candidate domain ontology for auto-generating questions and feedback is more reasonable than using Jones and Alani's approach. As Jones and Alani only used the top fifty terms expressed by single words to capture the educational corpus, which limits the efficiency of ontology evaluation, given that domain-specific concepts could be expressed by multiple word terms (Rospocher et al., 2012; Jones and Alani, 2006).

Contribution 2: Formative feedback generator

The formative feedback generator in this thesis uses the domain ontology to auto-generate the Knowledge Of Results (KOR), the Knowledge of Correct Response (KCR), the Bugs-Related (BR), the Topic Contingent (TC), and the Response Contingent (RC) types of feedback. The types of feedback supported in the generator were identified after carrying out a preliminary study (Study 2) in Appendix A to investigate the types of formative feedback teachers provide to students immediately after answering assessment questions. Study 2 revealed that teachers provide students with different types of feedback, which were either neglected (Kazi *et al.* (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013) focused on auto-generating hint feedback) or partially supported (S*ã*nchez-Vera *et al.* (Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011) focused on auto-generating KOR and KCR feedback) by the feedback generators reviewed in this thesis (see Section 2.3.3). Therefore, the formative feedback generator presented in Chapter 4 is the first to:

1. Auto-generate KOR, KCR, BR, TC, and RC types of feedback.
2. Auto-generate domain independent feedback using pre-existing domain ontology: No expert knowledge base which captures the experts' solutions to the problem scenario or human intervention (teacher or domain expert) is needed.
3. Associate the different types of feedback to questions auto-generated from the domain ontology.

Contribution 3: Personalised feedback algorithm

A formative feedback algorithm had been implemented in Moodle Virtual Learning Environment (VLE) to provide students with the appropriate type of feedback immediately after answering an assessment question. The algorithm is presented in Chapter 4 and it adopts Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001). The algorithm adapts the type of feedback provided to students based on student's characteristics: background knowledge about a specific educational topic, current level of knowledge while answering one question after another, and the task's characteristics which is the level of the question in Bloom's taxonomy. This allowed the relationship between student's characteristics, task's characteristics, and the personalised feedback to be studied for the first time.

Contribution 4: Analysing the quality of ontology-based auto-generated tests and questions

The Ontology-based Personalised Feedback Generator (OntoPeFeGe) associates each auto-generated question with different types of formative feedback. According to Black and William, it is essential to ensure the quality of both assessment questions and feedback in learning environments (Black and Wiliam, 1998). Previous evaluations of the auto-generated questions were confined to measuring the qualitative satisfaction of domain experts and students (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014). None of the ontology-based question generators assessed the quality of questions by examining the questions' difficulty, and the questions' ability to discriminate between high ability and low ability students. Therefore, this thesis assessed the quality of the ontology-based generated tests and questions in Chapter 5 using two statistical methods: the Classical Test Theory (CTT) (Alagumalai and Curtis, 2005; Ding and Beichner, 2009; Doran, 1980; Cohen et al., 2013; Erguven, 2014) and the Item Response Theory (IRT) (Baker, 2001; De Ayala, 2009; Hambleton and Swaminathan, 1985; Hambleton, 1991; Reckase, 2009; Schmidt and Embretson, 2003; Toland, 2014). An experiment was designed to investigate if the questions generated from domain ontologies and the assessment tests formed from these questions have satisfactory difficulty, discrimination and reliability values. Moreover, the experiment investigates if the question's difficulty and discrimination are affected by the ontology-based generation strategies, the level of the question in Bloom's taxonomy which is determined by the stem template (the text stating the question), and the type of question generated.

Three tests with 44 questions in total were analysed. The results obtained using the CTT revealed that the three assessment tests formed from the auto-generated questions had medium difficulty values which are very close to the value (0.5) that test authors are advised to achieve when constructing tests (Doran, 1980; Mitkov et al., 2009; Mitkov et al., 2006). In addition, the results revealed that the questions and tests have satisfactory positive discrimination values, which indicate that the questions and tests could effectively discriminate between high ability and low ability students, and that the questions may not need to be reviewed or eliminated from the assessment tests (Doran, 1980; Mitkov et al., 2006; Mitkov and Ha, 2003).

In addition to the CTT, the Item Response Theory (IRT) was used to assess the quality of auto-generated questions because of its invariant assumption. The IRT analysis revealed similar results to the CTT as the questions' discrimination indices had positive values which justify that the auto-generated questions may not need to be reviewed or eliminated from the assessment tests (Baker, 2001; Hambleton and Swaminathan, 1985). The results also revealed that two tests out of three had poor reliability which is due to the fact that the individual questions in these tests had satisfactory reliability values which were not high enough to improve the tests' overall reliability (Jones, 2009). Higher reliability values are desired and lower reliability values indicate that a question is not testing the same educational material or may not be testing the same educational material at the same level (Ding and Beichner, 2009). In the experiment carried out in Chapter 5 questions were generated from the same domain ontologies (OpenCyc (OpenCyc, 2011) and Computer Networks (Murugan.R et al., 2013) ontologies). As a result, the context of the educational material being tested is known. However, the auto-generated questions were designed to assess different educational concepts at different levels of Bloom's taxonomy, which may result in satisfactory reliability values at the questions' level (assessed using the point biserial correlation coefficient (R_{pb}) (Ding and Beichner, 2009; Schmidt and Embretson, 2003; Brown, 1996)) but low reliability values at the test's level (assessed using Cronbach's α (Cronbach and Shavelson, 2004)).

As mentioned above, Chapter 5 also investigated the effect of the ontology-based generation strategies, the level of the questions in Bloom's taxonomy, and the types of questions on the questions quality measurements. The results revealed that the generation strategies, the level of the questions in Bloom's taxonomy, and the type of questions affect the question's difficulty and discrimination. This provides guidance for developers and researchers working in the field of ontology-based question generators.

Contribution 5: Analyse the effect of personalised feedback on students' performance and learning gain in the assessment test

Chapter 6 presents the experiment carried out in Moodle VLE to evaluate the Ontology-based Personalised Feedback Generator (OntoPeFeGe). Both the personalised feedback algorithm and the auto-generated feedback were evaluated.

The results revealed that the personalised feedback algorithm in OntoPeFeGe only improved students' performance when the personalised feedback started by providing students with Response Contingent feedback. This suggests that students who received feedback with the highest level of information tend to have a statistically significant increase in their performance compared to students who received personalised feedback which starts with Bugs-related feedback or Topic Contingent feedback.

Moreover, Chapter 6 investigated the effect of personalised feedback and KOR feedback on students' performance and learning gain at questions designed to assess students at different levels in Bloom's taxonomy. The results revealed that both the personalised feedback and KOR feedback have the same effect on students' performance and learning gain when provided to students after answering questions designed to assess them at the knowledge and comprehension levels in Bloom's taxonomy. However, the effect of personalised feedback and KOR feedback on students' performance and learning gain differed when provided to students after answering questions designed to assess them at the application and analysis levels in Bloom's taxonomy. While the personalised feedback had no statistically significant effect on students' performance and learning gain when provided to students after answering questions designed to assess them at the application level, the KOR feedback improved students' performance significantly. On the other hand, the personalised feedback improved students' learning gain significantly when provided to students after answering questions designed to assess them at the analysis level in Bloom's taxonomy and KOR feedback decreased their performance significantly.

Chapter 6 also studied the relationship between students' background and the personalised feedback. The results obtained were similar to the results reported by Arroyo *et al.* (Arroyo et al., 2011; Woolf et al., 2010) as students with low background knowledge benefit more from the personalised feedback compared to students with high background knowledge. Moreover, students with low background knowledge had an increase in their performance after receiving the personalised feedback while students who received KOR feedback had no change in their performance.

Students and teachers' satisfaction regarding the ontology-based auto-generated feedback was also investigated. The results revealed that 72.92% of the students in the experimental group agreed that the feedback is useful, 70.83% agreed that the feedback generated was easy to read, and 68.75% agreed that the formative feedback helped them answer other questions in the assessment test. However, students in the control group had approximately similar results and one-third of students in both groups were not satisfied. Therefore, further investigation was carried out and the results revealed that students in the experimental group who had an improvement in their performance tend to agree more that the feedback was easy to read. Moreover, students with low background knowledge tend to agree more than students with high background knowledge that the feedback helped in answering some of the upcoming questions in the tests.

The teachers (three domain experts) were also satisfied about the auto-generated feedback, as they agreed that the questions and feedback were easy to read and the OntoPeFeGe provides students with different information at different levels of details. Moreover, they agreed that the feedback pedagogical content is reasonable and related to the question.

7.3 Future Directions

The proposing of the Ontology-based Personalised Feedback Generator (OntoPeFeGe) system and the Terminological ONtology Evaluator (TONE), and the results achieved so far suggests several paths for future research:

Support more types of questions which aim to assess students at different levels in Bloom's taxonomy

OntoPeFeGe supports the generation of true and false, multiple choice, and short answer questions. In future work, many other types of question such as matching or essay questions could be integrated into the system. Moreover, the OntoPeFeGe auto-generates assessment questions which assess students at the knowledge, comprehension, application, and analysis levels in Bloom's taxonomy. The stem templates could be extended in the future work to support the synthesis and evaluation levels in Bloom's taxonomy.

Supporting more types of questions which assess students at different levels of Bloom's taxonomy allows further analysis on the quality of the auto-generated questions and how the questions' quality could be affected by the ontology-based generation strategy, the level of the question in Bloom's taxonomy, and the type of question generated.

Support more types of formative feedback

OntoPeFeGe supports the auto-generation of Knowledge Of Response feedback, Knowledge of Correct Response feedback, Bugs Related (BR) feedback, Topic contingent (TC) feedback, and Response Contingent (RC) feedback, which teachers usually provide to students in VLEs immediately after answering an assessment question. These types of feedback could be enhanced in future work. In addition, there are many other types of feedback which teachers could provide to students in VLEs (e.g., hint feedback) and could be integrated into OntoPeFeGe in future work (Mason and Bruning, 2001; Shute, 2008).

Improve the stem templates (interrogative sentences) using natural language processing techniques

Chapter 3 illustrated the stem templates used to assess students at different levels in Bloom's taxonomy (Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013). The stem templates are instantiated during the generation process and sometimes they are syntactically wrong. In this thesis, the generated questions were corrected manually by a domain expert in the field of computer networks. However, this process could be automated by applying natural language processing techniques.

Enhance the Terminological ONtology Evaluator (TONE)

TONE could be enhanced by creating a metric which assesses the capability of the candidate domain ontology to:

1. Auto-generate knowledge, comprehension, application, and analysis questions.
2. Auto-generate *the reason why the correct answer is correct* and *the reason why the incorrect answer is incorrect* pedagogical contents.

Examine the importance of TONE to developers and researchers working in the field of ontology-based question and feedback generators and to teachers in learning environments

TONE (TONE-R and TONE-A) had lower variance in the Coverage Recall, Precision and Fmeasure Scores compared to Rospocher's approach (Rospocher et al., 2012). Moreover, TONE had a higher agreement in the F-measure Score and Semantic Richness Score compared to Rospocher's approach. As future work, the importance of having low variance in the ontology evaluation metrics and high agreement between the rankings of the ontology evaluation metrics to developers, researchers and teachers should be examined.

Further investigation of the Item response theory invariance assumption

The Item Response Theory invariance assumption in Section 5.2 was only investigated by dividing the groups of students into low ability and high ability. The invariance assumption could also be tested on groups of students divided randomly or according to their gender, year of study, and culture.

Further investigation of the effect of the ontology-based generation strategies on the question's difficulty and discrimination

Chapter 5 investigated the effect of the three main ontology-based generation strategies (class, terminology and property) on the question difficulty and discrimination indices. The effect of the ontology-based generation sub-strategies was not investigated due to limitations in the experiment design, as the experiment contained only 44 questions which do not cover all the ontology-based generation sub-strategies. Further investigation could be carried out by generating questions using all the ontology-based sub-strategies, testing the questions on students, and studying the ontologybased generation strategies effect on the question difficulty and discrimination.

Build prediction models to predict the question difficulty and discrimination

Prediction models could be built using machine learning techniques such as multiple linear regression (James et al., 2014) to predict the question difficulty (very difficult, moderately difficult, moderately easy, very easy) and the question discrimination (low, medium, high) using the following three features: the ontology-based generation strategy, the level of the question in Bloom's taxonomy, and the type of question. This aims to provide guidance for developers and researchers working in the field of ontology-based question generators and help them save time and effort in terms of testing the auto-generated questions on real students.

Improve the personalised feedback algorithm by applying machine learning techniques

The personalised feedback algorithm implemented and evaluated in this thesis adopts Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001) which adapts the types of feedback based on the student and task characteristics. The task's characteristics only considered the level of the question in Bloom's taxonomy. Future work may include considering other task characteristics which were analysed in Chapter 5 such as the question difficulty index, the question discrimination index, and the type of question (true and false, multiple choice and short answer), and applying machine learning techniques to provide students with the appropriate type of feedback which aims to improve their performance.

Compare the ontology-based auto-generated feedback in OntoPeFeGe with feedback hard-coded by teachers

The ontology-based auto-generated feedback in OntoPeFeGe was evaluated by assessing the effect of feedback on students' performance and examining students' and teachers' satisfaction regarding the auto-generated feedback. The evaluation approach was similar to the evaluations carried out by the ontology-based feedback generators introduced in Section 2.3.3. Future work may include carrying out further experiments to investigate the effect of the different types of feedback auto-generated in OntoPeFeGe on students' performance, and how the effect differs from the different types of feedback hard-coded by the teachers.

7.4 Chapter Summary

This chapter presents a review of the work carried out in this thesis and the significance of the major contributions. In summary, the review carried out in this chapter revealed that this thesis met its primary aim through TONE and OntoPeFeGe, which were proposed in Chapters 3 and 4 respectively. OntoPeFeGe contributes to the research carried out in the personalised feedback frameworks (Mason and Bruning, 2001; Narciss et al., 2014; Arroyo et al., 2000; Arroyo et al., 2001; Arroyo et al., 2011; Woolf et al., 2010), the ontology-based formative feedback generators (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013; Duboc et al., 2011; Al-Yahya, 2014; Frutos-Morales et al., 2010; S et al., 2012; Castellanos-Nieves et al., 2011) and the ontology-based question generators (Papasalouros et al., 2008; Papasalouros et al., 2011; Cubric and Tosic, 2011; Grubisic, 2012; Grubisic et al., 2013; Al-Yahya, 2011; Al-Yahya, 2014). OntoPeFeGe is the first to auto-generate different types of feedback which are domain independent and provide students with personalised feedback. In addition, OntoPeFeGe allowed analysing the quality of the ontology-based auto-generated questions which were designed to assess students at different levels in Bloom's taxonomy quantitatively for the first time.

In addition to OntoPeFeGe, this thesis presented TONE which contributes to the field of terminological data-driven ontology evaluation approaches (Brank et al., 2005; Maedche and Staab, 2002; Lozano-Tello and Gmez-Prez, 2004; Porzel and Malaka, 2004; Brewster et al., 2004; Jonquet et al., 2010; Martnez-Romero et al., 2012; Martnez-Romero et al., 2014; Rospocher et al., 2012; Jones and Alani, 2006). TONE is the first ontology evaluation approach which evaluates the candidate domain ontologies coverage (Recall, Precision, and Fmeasure) and semantic richness. In addition, it is the first to select dynamically a cut-off point in the list of terms extracted from the corpus when different term extraction tools and terms recognition algorithms are used. TONE was compared with other terminological data-driven ontology evaluation approaches and the results revealed that TONE had lower variance in the Semantic Richness Score compared to Rospocher's (Rospocher et al., 2012) and Jones and Alani's (Jones and Alani, 2006) data-driven approaches and higher agreement compared to Rospocher's approach (Rospocher et al., 2012). Moreover, TONE had lower variance and higher agreement in the F-measure Scores compared to Rospocher's approach (Rospocher et al., 2012). This helps the user obtain approximately similar ontology evaluation metrics regardless of the list of terms extracted from the corpus.

This chapter also discussed several paths for future research, which could be applied to OntoPeFeGe and TONE.

Appendix A

Preliminary Studies

This appendix presents two preliminary studies carried out prior to the design and implementation of the Ontology-based Personalised Feedback Generator (OntoPeFeGe) system explained in Chapter 4. The first study aimed to investigate the importance of building a system that focuses on formative feedback which students receive in Virtual Learning Environments (VLEs), specifically feedback provided to students immediately after answering an assessment question. The importance of building such a system was examined by assessing if the formative feedback in VLEs helps students understand and learn the educational material and if students use the formative feedback during exam revision.

The second study investigates the different types of formative feedback the teachers provide to students immediately after answering an assessment question. In addition, the study examines if the different types of feedback teachers provide to students differ across educational courses. These investigations aim to:

1. Justify the adoption of Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001) in OntoPeFeGe where the feedback is provided to students immediately after answering an assessment question. Mason and Bruning's framework provides students with personalised Bugs- Related (BR), Topic Contingent (TC), and Response Contingent (RC) feedback by considering the student and the task characteristics (Mason and Bruning, 2001). Therefore, it is essential to examine how frequent teachers use the BR, TC, and RC types of feedback before adopting their framework in OntoPeFeGe.

2. Determine the types of feedback OntoPeFeGe should focus on.

Several types of feedback which vary from verifying students answer (e.g., correct/incorrect) towards explaining to students the reason why the correct answer is correct and the incorrect answer is incorrect were investigated.

The two preliminary studies are illustrated in Section A.1 and Section A.2 respectively.

A.1 Study 1: Do students benefit from formative feedback?

This study aims to examine the importance of formative feedback students receive immediately after answering assessment questions. In general, to students in VLEs, and in particular to students in the field of computer networks where the OntoPeFeGe evaluation experiments were carried out. The study investigated the following questions:

Q1: Does the formative feedback in VLEs help students understand and learn the educational material?

Q2: Do students use the formative feedback in VLEs during exam revision?

Several experimental studies have been carried out to answer both questions in the medical and chemistry domains. Dermo and Carpenter performed a study in Blackboard VLE (Blackboard, 1997) on seventy eight students in the medical domain. Their results revealed that 88.7% of students agreed that the formative feedback helped them understand and learn the educational material, and 22% of students agreed that they used the formative feedback during examination and revision period (Dermo and Carpenter, 2011). Similarly, Bedford and Price carried out a study in Moodle VLE (Moodle, 2011) on ninety-eight students in the chemistry domain. Their results revealed that 73% of students agreed that the feedback provided to them in Moodle VLE helped them understand and learn the educational material, and 64% of students agreed that they used the feedback during exam revision (Bedford and Price, 2007).

Both studies revealed that some students in the medical and chemistry domains disagreed that the feedback helped them understand and learn the educational material and that they used the feedback during the exam revision. Dermo and Carpenter (Demaidi, 2015a) did not investigate this. However, Bedford and Price investigated students disagreement and stated that the main percentage of students who disagreed were high performing, well motivated, and independent learners that did not need to

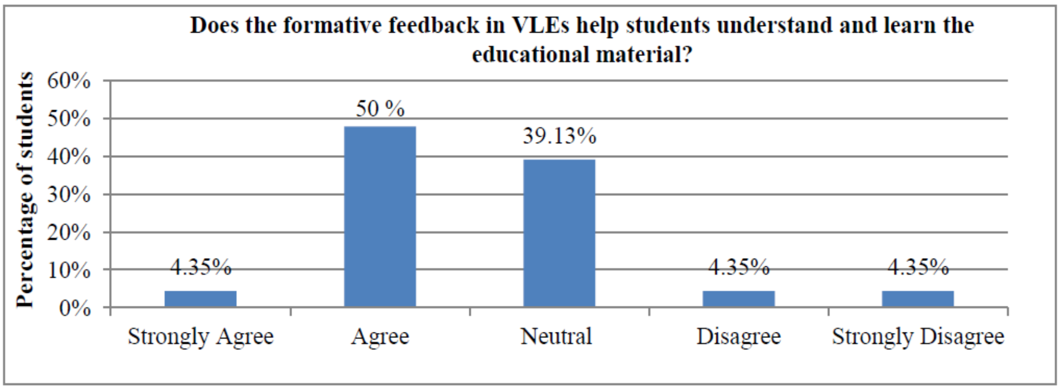


Figure A.1: Distribution of student responses to Q1

use the feedback to either understand the educational material or during exam revision (Demaidi, 2014) classified as surface learners disagreed because they did not make the connection between the feedback and the educational material (Demaidi, 2014).

The following section presents the experiment carried out to investigate **Q1** and **Q2** on students in the field of computer networks.

Experiment

In 2012, twenty-three third year undergraduate students enrolled in the Digital Wireless Communications and Networks course (COMP38512) (of Manchester, 2012b) at the School of Computer Science, the University of Manchester volunteered to take part in the study.

Moodle VLE was used to provide students with the Digital Wireless Communications and Networks course materials and assignments. Therefore, a 5-point Likert scale (1: strongly disagree, to 5: strongly agree) questionnaire was provided to students in Moodle VLE one month prior to the exams period. Students were asked to answer the questionnaire by signing into Moodle VLE.

Fig A.1 shows that more than half (54.35%) of the students in the computer networks domain agreed (the average ranking score was 3.43) that the immediate feedback provided in Moodle and Blackboard VLEs helped them understand and learn the educational material. However, the percentage is low compared to students in the medical (88.7%) and chemistry (73%) domains.

The results also revealed that 52.18 % of students agreed (the average ranking score was 3.5) that the feedback provided to them was useful during exam revision period (see Fig A.2).

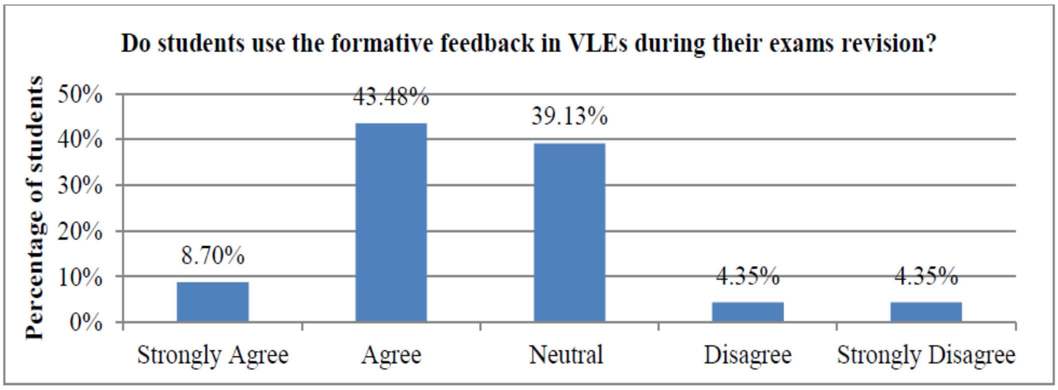


Figure A.2: Distribution of student responses to Q2

Even though, more than half of the students agreed that the formative feedback helped them understand the educational material and that they used the formative feedback during exam revision, 39.13% of the students had a neutral opinion and 8.7% of the students disagreed. To investigate this, students' overall performance in the Digital Wireless Communications and Networks course was mapped to their responses and the results obtained were similar to Bedford and Price's study, as students who disagreed were students who had the highest average performance scores in the course. The analysis also revealed that students who had a neutral opinion had the lowest average performance scores in the course.

In summary, students from different educational domains agree that the formative feedback provided to them immediately after answering assessment questions help them understand and learn the educational material. Moreover, they agree that they benefited from the feedback as they intended to use it during the exams revision. These observations suggest the importance of building a OntoPeFeGe, which focuses on the formative feedback students receive immediately after answering an assessment question.

A.2 Study 2: Observe types of feedback provided to students

This study aims to justify adopting Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001) in OntoPeFeGe and to determine the types of feedback the OntoPeFeGe should focus on by answering the following questions:

Q1: Which types of formative feedback do teachers usually use in Learning Environments?

Q2: Do types of feedback teachers provide to students differ across educational courses?

In order to answer the questions above, the formative feedback content provided to students in learning environments should be analysed to identify which types of feedback teachers use. Brown and Glover proposed a qualitative coding system which categorises the types of feedback (Mason and Bruning, 2001; Narciss et al., 2014; Shute, 2008; Narciss, 2013; Bangert-Drowns et al., 1991; Mory, 2004) according to the depth of detail provided in each type into the following three main categories (Brown et al., 2003; Glover and Brown, 2006):

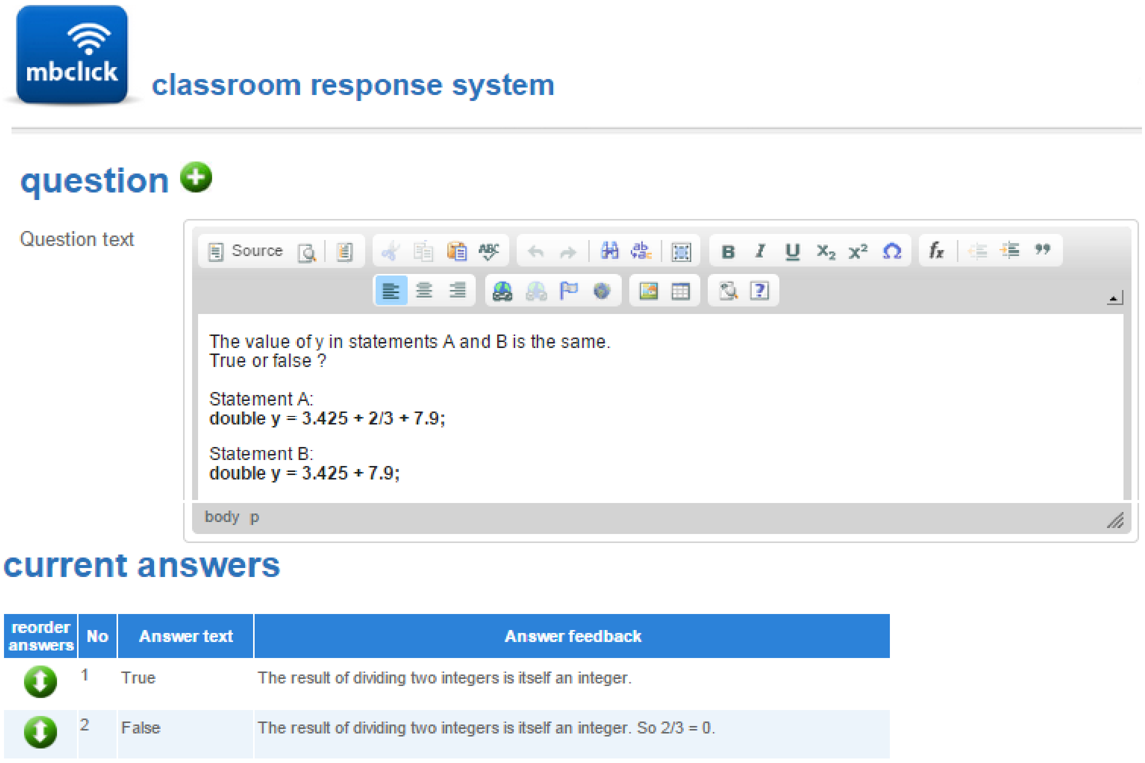
1. Indication feedback which notifies students if the provided answer is correct or incorrect. This category contains the Knowledge Of Result (KOR) feedback.
2. Correction feedback which provides students with the correct answer. This category contains the Knowledge of Correct Response (KCR) feedback.
3. Explanation feedback, which provides students with explanation relevant to their answers. For example, students who fail to answer the assessment question receive feedback which explains to them the reason why their answer is incorrect. The explanation feedback defined by Brown and Glover contains the Bugs-related (BR), the Topic Contingent (TC) and the Response Contingent (RC) types of feedback that were introduced in Section 2.1.

In this study, Brown and Glover's coding system is used to answer **Q1** and **Q2** and investigate the usage of formative feedback provided to students immediately after answering an assessment question.

Experiment

Three teachers volunteered to take part in the experiment from the following schools at the University of Manchester: the School of Electrical and Electronic Engineering (EEE), the School of Social Science, and the School of Chemistry.

The teachers used the mbclick (Rubner, 2012; of Manchester, 2012a) assessment system which is an electronic voting system developed by the University of Manchester to assess students during a lecture session (Rubner, 2012). The system provides



question

Question text

The value of y in statements A and B is the same.
True or false ?

Statement A:
double y = 3.425 + 2/3 + 7.9;

Statement B:
double y = 3.425 + 7.9;

current answers

reorder answers	No	Answer text	Answer feedback
⬇	1	True	The result of dividing two integers is itself an integer.
⬇	2	False	The result of dividing two integers is itself an integer. So 2/3 = 0.

Figure A.3: A true and false question in mbclick and the associated feedback comments

teachers with a web-based VLE to create true and false, multiple choice, and short answer questions. It also provides teachers with the facilities to associate hard-coded feedback, which is called the feedback comment to each question's option. Fig A.3 is a screen shot of a true and false question created in the mbclick system. It shows the two formative feedback comments created by the teacher for the question's true and false options.

Students used their mobile phones to access the mbclick web-based environment and answer the questions. After students have submitted their answers, mbclick provides them with immediate feedback related to their selected option (Rubner, 2012).

In this study, the feedback comments the three teachers provided to students using mbclick were analysed. Table A.1 shows the educational courses, the number of students, the level of students, the number of questions and the number of feedback comments analysed in this study.

Table A.1: Number of questions and feedback comments analysed in mbclick

Discipline	**Course**	**Teacher ID**	**Number of students**	**Year of study**	**Number of questions**	**Number of feedback comments**
EEE	Java Programming [153]	A	109	Second year undergraduates	25	88
EEE	Data Networking [154]	A	64	Third year undergraduates	12	47
Social Sciences	Introductory Mathematics [155]	B	218	First year undergraduates	4	17
Chemistry	Introductory Chemistry [156]	C	225	First year undergraduates	9	36

Brown and Glover's feedback coding system (Brown et al., 2003; Glover and Brown, 2006) was used to analyse the KOR, KCR and explanation feedback comments[1] teachers provided to students in each of the courses presented in Table A.1. After that, further analysis was carried out to investigate the percentage of feedback comments in each of the explanation feedback categories: BR, TC, and RC types of feedback. The study also investigated other types of feedback, which teachers could provide to students in learning environments such as the hint the and Answer Until Correct (AUC) feedback.

Q1: Which types of formative feedback do teachers usually use in Learning Environments?

Several studies used Brown and Glover's coding system to address Q1, however, their analysis focused on delayed formative feedback which was provided to students three weeks after submitting their assignments (Glover and Brown, 2006; Fernndez-Toro et al., 2013; Walker, 2009). Brown and Glover analysed the types of feedback provided on 112 student assignments in the bioscience, chemistry and physics courses taught at the UK's Open University (Brown et al., 2004; Brown and Glover, 2005). They analysed 2150 feedback comments and the results are shown in Fig A.4. 60% of the feedback comments provided to students were KCR feedback, 13% were KOR feedback, and 27% were explanation feedback. Walker (Walker, 2009) carried out a similar study on three courses taught at the Faculty of Engineering (Engineering the Future, Information and Communication Technologies, Computer and Processors). She analysed 3000 feedback comments and the results obtained were approximately similar to Brown and Glover's, as 71% of the feedback comments were KCR feedback (see Fig A.4). In another study, Fernãndez-Toro *et al.* (Fernndez-Toro et al., 2013) followed the same approach and analysed 4000 feedback comments provided to students in two Spanish language courses taught at the UK's Open University and their results were similar to the previous findings, as KCR feedback was the highest type of feedback used (47%).

These studies revealed that KCR feedback is highly used compared to both KOR and explanation feedback when delayed formative feedback is provided to students. This could be due to the fact that delayed feedback provides students with the opportunity to think about their answers and self correct their misconceptions prior to receiving

[1] Each question consisted from two to five options and each option was associated with a feedback comment.

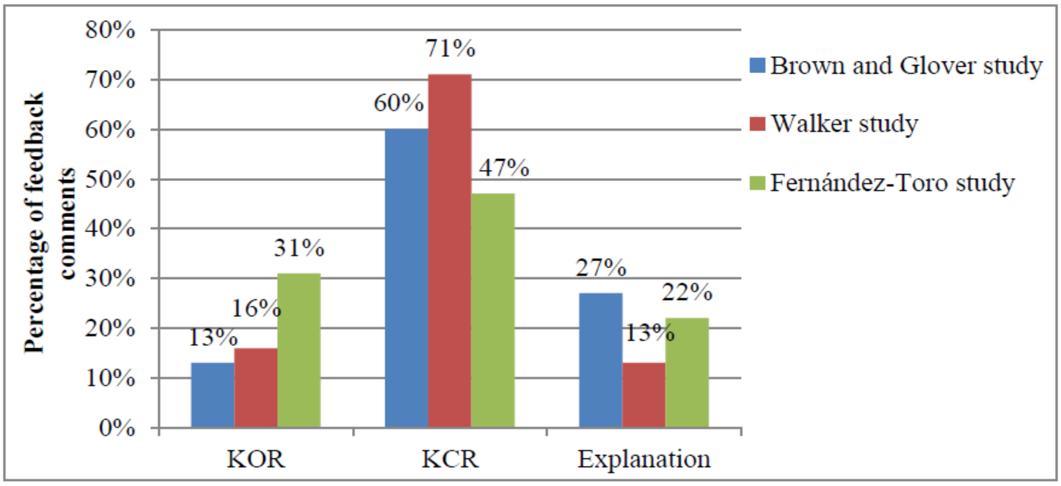


Figure A.4: Percentage of feedback comments obtained using Brown and Glover's coding system

the feedback. Moreover, Fernãndez-Toro (Demaidi, 2015b) stated that KCR feedback is highly used in the Spanish language assignments as most of the errors are usually language errors and teachers tend to annotate and correct the errors rather than giving explanation to students. Glover (Demaidi, 2015c) added that this could be related to the teachers involved in the course who do not have time to write explanation feedback on each assignment as they have hundreds of assignments to mark.

This study follows a similar approach to the studies mentioned above. However, Brown and Glover's coding system is used to analyse the feedback comments provided to students immediately after answering the assessment questions in the four educational courses shown in Table A.1. Fig A.5 shows that the percentages of KOR feedback comments used in the four educational courses ranged between [11%-34%] and were approximately similar to the results obtained by Brown and Glover, Walker, and Fernãndez-Toro studies. However, the percentages of KCR feedback comments in the four educational courses tend to be low (percentages of KCR feedback comments ranged between [6%-19%]) compared to the results obtained by Brown and Glover, Walker, and Fernãndez-Toro studies (percentages of KCR feedback comments ranged between [47%-71%]). Moreover, the percentages of explanation feedback comments were much higher in the four educational courses [47%-83%] compared to Brown and Glover, Walker, and Fernãndez-Toro studies [13%-27%].

It is obvious that the teachers in learning environments provide students with KOR, KCR, and explanation types of feedback. However, the analysis carried out in this

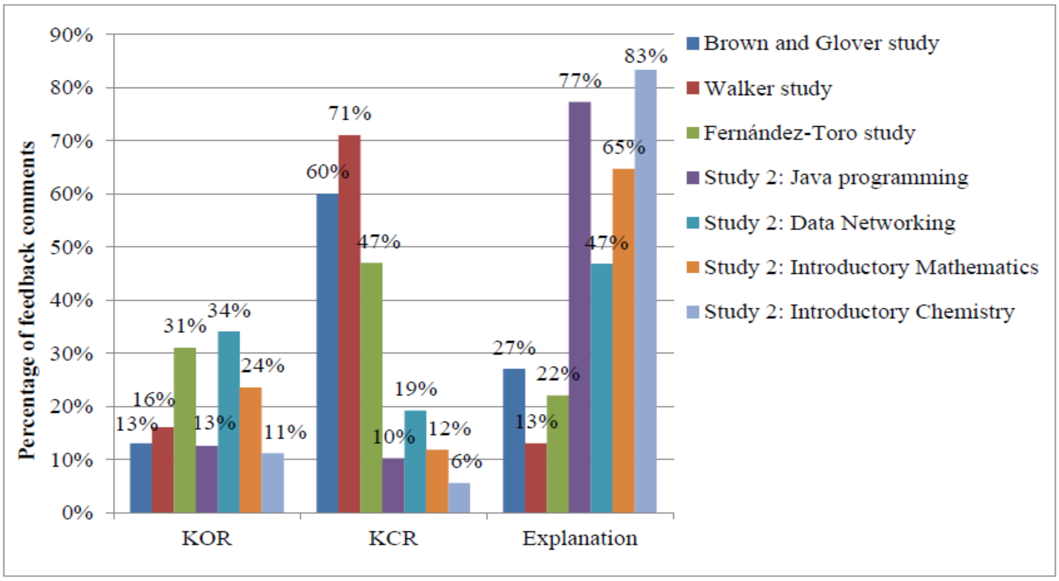


Figure A.5: Percentage of types of feedback provided to students in learning environments

study revealed that there is a difference in the percentages of KCR and explanation feedback comments when feedback is provided to students immediately or after some delay. While KCR feedback is highly used in Brown and Glover, Walker, and Fernãndez-Toro studies, explanation feedback is highly used when the formative feedback is provided to students immediately after answering an assessment question. A follow-up with one of the teachers who participated in the experiment revealed that the teacher believed that explanation feedback helps students gain and close the gap in their knowledge more than other types of feedback, especially when the feedback is provided to students immediately after answering an assessment question. The teacher also agreed with Glover that providing students with explanation feedback costs a lot of time and he suggested that it would be useful to auto-generate the feedback.

More detailed analysis was carried out to investigate the percentages of feedback comments in:

1. The explanation feedback categories: BR, TC, and RC.

2. The hint feedback.

3. The AUC feedback.

Fig A.6 shows that teachers used the BR, TC, and RC feedback comments in the four educational courses. However, hint and AUC feedback were not used.

In summary, using Brown and Glover's coding system revealed that teachers in learning environments provide students with KOR, KCR, and explanation feedback. The results also revealed that with mbclick where feedback is provided to students immediately after answering an assessment question, teachers tend to provide students with more explanation feedback compared to the studies carried out by Brown and Glover, Walker, and Fernãndez-Toro where students received delayed feedback. Further analysis was carried out to investigate the usage of BR, TC, and RC feedback in the four educational courses. The results revealed that teachers used BR, TC, and RC feedback.

The results obtained in this study justifies adopting Mason and Bruning's personalised feedback framework in the OntoPeFeGe, which aims to provide students with personalised feedback immediately after answering an assessment question. Moreover, the results suggest that OntoPeFeGe should focus on auto-generating KOR, KCR, BR, TC, and RC feedback, which were used in the four educational courses. Even though the study helped identify the types of formative feedback which teachers use or do not use in learning environments, it is important to keep in mind that the number of feedback comments analysed in this study is small compared to the feedback comments analysed by Brown and Glover, Walker, and Fernãndez-Toro.

Q2: Do types of feedback teachers provide to students differ across educational courses?

Fig A.6 shows that the three teachers in this study provided students with KOR, KCR, and explanation feedback. The highest percentage of feedback comments in the four educational courses were the explanation feedback with percentages in the range 47% to 83% and the KCR feedback was the lowest type of feedback used in the four educational courses with percentages in the range 6% to 19%. Despite the small sample of feedback comments analysed in the four educational courses, we can conclude that in our test cases the three teachers followed the same pattern in feedback usage regardless of the level of the students (first year, second year, third year) and the educational course, i.e., the explanation feedback was the highest type of feedback used, followed by the KOR feedback, and then the KCR feedback.

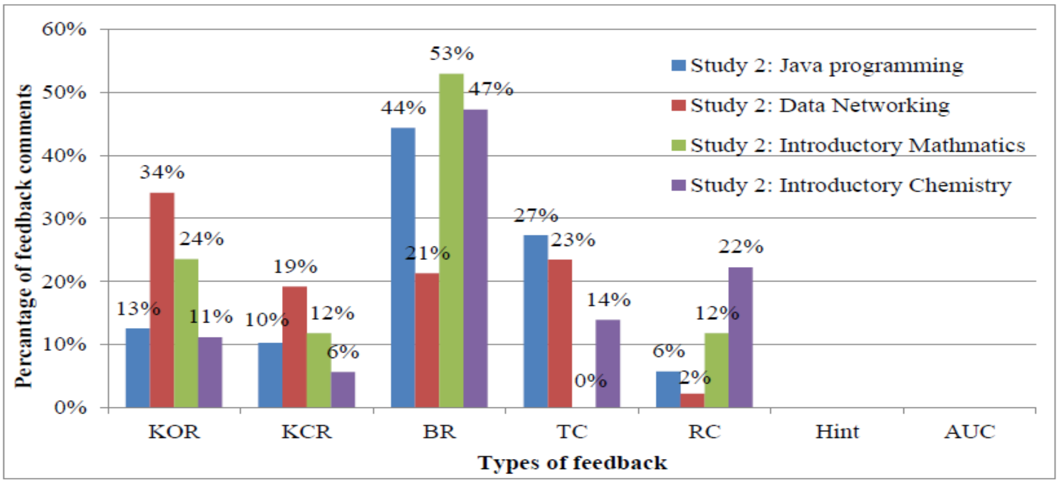


Figure A.6: Percentage of types of feedback provided to students in mbclick

A.3 Summary

This appendix presented two preliminary studies carried out prior to the design and implementation of the Ontology-based Personalised Feedback Generator (OntoPeFeGe) system explained in Chapter 4. Study 1 investigated the importance of building a system focusing on formative feedback which students receive in Virtual Learning Environments (VLEs), specifically the feedback provided to them immediately after answering an assessment question. The importance of building such a system was examined by assessing if the formative feedback in VLEs helps students understand and learn the educational material and if students use the formative feedback during exam revision. The results obtained support the importance of building the OntoPeFeGe as students from different educational domains (medicine, chemistry, and computer networks) agreed that the feedback in VLEs helped them understand and learn the educational material, and they use the formative feedback during their exams revision.

Study 2 investigated the different types of formative feedback the teachers provide to students immediately after answering an assessment question. In addition, Study 2 examined if the different types of feedback teachers provide to students differ across educational courses. These investigations aimed to justify the adoption of Mason and Bruning's personalised feedback framework (Mason and Bruning, 2001) in OntoPeFeGe and to determine the types of feedback OntoPeFeGe should focus on.

The feedback comments provided to students in four educational courses taught at the University Of Manchester were analysed using Brown and Glover's coding system and the results revealed that teachers provided students with KOR, KCR, and explanation feedback. Moreover, the results revealed that with mbclick teachers tend to provide students with more explanation feedback compared to the studies carried out by Brown and Glover, Walker, and Fernãndez-Toro. These results suggest the importance of explanation feedback in learning environments where feedback is provided to students immediately after answering an assessment question.

Despite the difference in the explanation feedback usage between learning environments where feedback is provided to students immediately or after some delay, researchers and teachers agreed that providing students with explanation feedback is time consuming (Demaidi, 2015c; Demaidi, 2013). This suggests the importance of building the OntoPeFeGe, which auto-generate explanation feedback to students and save teachers' time and effort.

Further analysis was carried out in the Study 2 to investigate the usage of BR, TC, and RC types of feedback, which are part of the explanation feedback category according to Brown and Glover's coding system. The results revealed that teachers used BR, TC, and RC feedback in the four educational courses. Moreover, the analysis revealed that the three teachers followed the same pattern in feedback usage regardless of the level of students and the educational course (i.e., the explanation feedback was the highest type of feedback used, followed by the KOR feedback, and then the KCR feedback). The results obtained justify adopting Mason and Bruning's personalised feedback framework in the OntoPeFeGe.

Study 2 also justifies the need for a new feedback generator which auto-generates KOR, KCR, BR, TC, and RC feedback. Several feedback generators were surveyed in Section 2.3.3. These generators focused on generating specific types of feedback. Kazi *et al.* (Kazi et al., 2012; Kazi et al., 2010; Kazi et al., 2013) auto-generated hint feedback, which was not used by the teachers in Study 2 due to the nature of the assignment (question) provided to students. Students in groups of six to eight using Kazi *et al.* feedback generator were asked to answer an essay question by drawing causal links between concepts in the medical domain. Therefore, providing students with hint feedback helped them discuss, review and re-submit their answers. However, in the four educational courses investigated in Study 2 teachers provided students with several assessment questions and students were asked to answer the questions individually. In addition, students were not allowed to re-submit their answers. Therefore, teachers focused on providing students with explanation feedback which they believed would help students close the gap in their knowledge more than other types of feedback

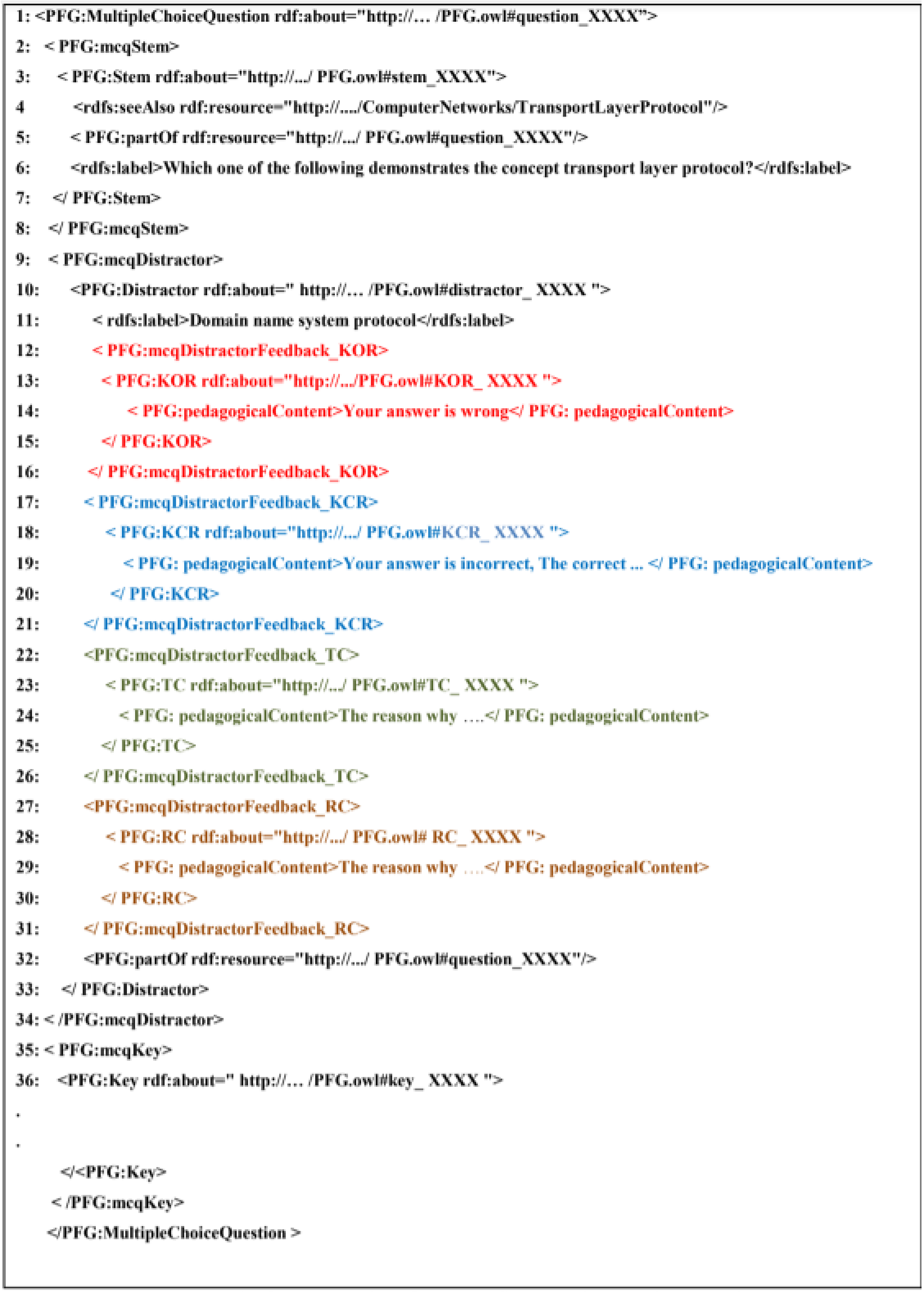
(Demaidi, 2013). S$\tilde{a}$nchez-Vera *et al.* (S et al., 2012; Castellanos-Nieves et al., 2011) feedback generator auto-generated KOR and KCR feedback and ignored the explanation feedback (BR, TC, and RC) which was highly used by the three teachers in Study 2. Duboc *et al.* (Duboc et al., 2011; Duboc, 2013) feedback generator auto-generated more types of feedback. However, their generator focused on auto-generating KCR, BR, and TC feedback and ignored both KOR and RC feedback.

Appendix B

Part of OntoPeFeGe Domain Ontology

Fig B.1 shows part of the Ontology-based Personalised Feedback Generator (OntoPeFeGe) question and feedback domain ontology OWL file in RDF/XML format (W3C, 2004b) which is mapped to Moodle XML format (Moodle, 2012). The example file contains a multiple choice question (MCQ) individual auto-generated in OntoPeFeGe and the different types of formative feedback individuals which are associated with the question's *Key* and *Distractors*.

Fig B.1 shows the multiple choice question individual (line 1) which has a *Stem* individual (line 3), a *Distractor* individual (line 10), and a *Key* individual (line 36). Fig B.1 also shows that the *Stem* individual (line 3) has a label property (line 6) that captures the stem template which is fulfilled by traversing the domain ontology (e.g., the Computer Networks ontology created by Sakathi (Murugan.R et al., 2013)). The *Distactor* individual (line 10) shown in Fig B.1 also has a label property (line 11), KOR feedback individual (line 13), KCR feedback individual (line 18), TC feedback individual (line 23), and RC feedback individual (line 28). Each type of formative feedback has a *pedagogical content* property which captures the auto-generated formative feedback from the domain ontology (see Section 4.1.4). The *pedagogical content* property for the KOR feedback individual (line 14) is *Your answer is wrong* because the KOR feedback individual is associated with the *Distractor* individual.



```
<PFG:MultipleChoiceQuestion rdf:about="http://… /PFG.owl#question_XXXX">
 < PFG:mcqStem>
   < PFG:Stem rdf:about="http://.../ PFG.owl#stem_XXXX">
     <rdfs:seeAlso rdf:resource="http://..../ComputerNetworks/TransportLayerProtocol"/>
     < PFG:partOf rdf:resource="http://.../ PFG.owl#question_XXXX"/>
     <rdfs:label>Which one of the following demonstrates the concept transport layer protocol?</rdfs:label>
   </ PFG:Stem>
  </ PFG:mcqStem>
  < PFG:mcqDistractor>
     <PFG:Distractor rdf:about=" http://… /PFG.owl#distractor_ XXXX ">
        < rdfs:label>Domain name system protocol</rdfs:label>
        < PFG:mcqDistractorFeedback_KOR>
          < PFG:KOR rdf:about="http://.../PFG.owl#KOR_ XXXX ">
             < PFG:pedagogicalContent>Your answer is wrong</ PFG: pedagogicalContent>
          </ PFG:KOR>
        </ PFG:mcqDistractorFeedback_KOR>
       < PFG:mcqDistractorFeedback_KCR>
          < PFG:KCR rdf:about="http://.../ PFG.owl#KCR_ XXXX ">
            < PFG: pedagogicalContent>Your answer is incorrect, The correct ... </ PFG: pedagogicalContent>
           </ PFG:KCR>
       </ PFG:mcqDistractorFeedback_KCR>
       <PFG:mcqDistractorFeedback_TC>
          < PFG:TC rdf:about="http://.../ PFG.owl#TC_ XXXX ">
            < PFG: pedagogicalContent>The reason why ….</ PFG: pedagogicalContent>
         </ PFG:TC>
       </ PFG:mcqDistractorFeedback_TC>
       <PFG:mcqDistractorFeedback_RC>
          < PFG:RC rdf:about="http://.../ PFG.owl# RC_ XXXX ">
            < PFG: pedagogicalContent>The reason why ….</ PFG: pedagogicalContent>
         </ PFG:RC>
       </ PFG:mcqDistractorFeedback_RC>
       <PFG:partOf rdf:resource="http://.../ PFG.owl#question_XXXX"/>
    </ PFG:Distractor>
< /PFG:mcqDistractor>
< PFG:mcqKey>
  <PFG:Key rdf:about=" http://… /PFG.owl#key_ XXXX ">
.
.
    </<PFG:Key>
  < /PFG:mcqKey>
 </PFG:MultipleChoiceQuestion >
```

Figure B.1: Part of OntoPeFeGe output file (multiple choice question)

Appendix C

Stem Templates in OntoPeFeGe

The appendix illustrates the stem templates integrated in OntoPeFeGe.

Table C.1: The stem templates integrated in OntoPeFeGe

Question Number	Stem template	Bloom's level	Type of question	Generation strategy	Literature
1	Which of the following definitions describes the concept Class A?	Knowledge	Multiple choice	Property-based	Cubric and Tosic (Cubric and Tosic, 2011)
2	Read the paragraph and decide which one of the following concepts it defines?	Knowledge	Multiple choice	Property-based	Cubric and Tosic (Cubric and Tosic, 2011)
3	Are Class A and Class B directly connected?	Knowledge	True and false	Terminology-based	Grubisic (Grubisic et al., 2013)
4	Are Class A and Class B directly connected with Property?	Knowledge	True and false	Property-based	Grubisic (Grubisic et al., 2013)
5	Is Class A subclass of Class B?	Knowledge	True and false	Terminology-based	Grubisic (Grubisic et al., 2013)
6	Is Class A superclass of Class B?	Knowledge	True and false	Terminology-based	Grubisic (Grubisic et al., 2013)
7	What directly connects Class A and Class B?	Knowledge	Multiple choice	Property-based	Grubisic (Grubisic et al., 2013)
8	Which one of the following response pairs relates in the same way as: Class A Property Class B	Comprehension	Multiple choice	Property-based	Cubric and Tosic (Cubric and Tosic, 2011)
9	Are Class A and Class B indirectly connected?	Comprehension	True and false	Terminology-based	Grubisic (Grubisic et al., 2013)
10	Which sub-concept is directly connected by Property with Class A?	Comprehension	Multiple choice	Property-based	Grubisic (Grubisic et al., 2013)
11	Which super-concept is directly connected by Property with Class A?	Comprehension	Multiple choice	Property-based	Grubisic (Grubisic et al., 2013)
12	Which one of the following demonstrates the concept Class A?	Application	Multiple choice	Class-based	Cubric and Tosic (Cubric and Tosic, 2011)
13	Class A is —— of Class B	Application	Multiple choice	Property-based	Grubisic (Grubisic et al., 2013)
14	How many concepts is Class A connected with?	Application	Short answer	Property-based	Grubisic (Grubisic et al., 2013)
15	Analyse the following text and decide which one of the following words is a correct replacement for the blank space in the text?	Analysis	Multiple choice	Property-based	Cubric and Tosic (Cubric and Tosic, 2011)
16	Read the paragraph and decide which one of the following concepts generalise the concept defined by it	Analysis	Multiple choice	Property-based and Class-based	Cubric and Tosic (Cubric and Tosic, 2011)
17	Class A and Class B are directly connected by —-?	Analysis	Short answer	Property-based	Grubisic (Grubisic et al., 2013)
18	Is Property of Subject Object?	Analysis	True and false	Property-based	Grubisic (Grubisic et al., 2013)
19	What is Property of Class A?	Analysis	Multiple choice	Property-based	Grubisic (Grubisic et al., 2013)
20	Whose Property is Class A?	Analysis	Multiple choice	Property-based	Grubisic (Grubisic et al., 2013)

Appendix D

Assessment Tests

This appendix presents the link to the three tests provided to students in Moodle Virtual Learning Environment. Please click on the link below to see the assessment tests. The tests could be displayed in a web browser.

`https://drive.google.com/open?id=0B25z6hoT8MGnNmx4VzZrTXA3RzA`

Appendix E

Item Response Theory (IRT) Assumptions

E.1 Unidimensionality

The Principal Component Analysis (PCA) is used in this thesis to analyse the unidimensionality of the assessment tests in Chapter 5 by:

1. Finding the correlation matrix between students' responses to the assessment questions in the assessment test.

2. Measuring the eigenvalues for the correlation matrix to determine the amount of variance captured by each component. Only components with an eigenvalue above one are maintained according to Kaisar-Guttman criterion (Yeomans and Golder, 1982). Kaisar-Guttman states that each question contributes one unit to the total variance and an eigenvalue less than one is less than the amount of variance contributed by one question, as a result it should be ignored.

If PCA results in more than one component, then the test is not unidimensional and more than one ability parameter (θ) is underlying the assessment test. PCA outputs a table with the number of components (ability parameters) underlying the test, and the amount of data variance captured by each component. Moreover, it produces a Scree plot (Cattell, 1966) to give a visual representation of the eigenvalues. Each eigenvalue larger than one represents a component (ability parameter) underlying the data.

E.2 Local Independence

Local dependence chi-square (LD x^2)) test (Chen and Thissen, 1997) is used to assess the IRT local independence assumption. It is applied for each pair of question in the assessment test. For example, the local independence chi-square *null hypothesis* states that *question x and question y are independent and students' answers to question x are independent of their answers to question y*. The LD x^2 test

starts by observing the following:

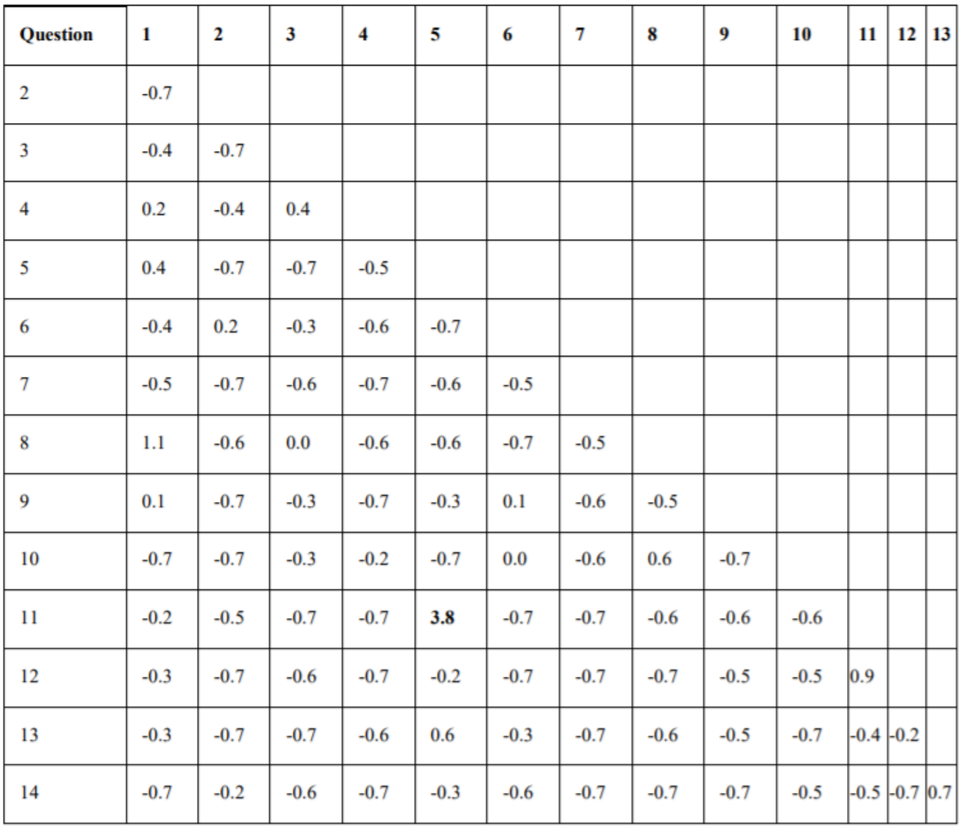
- The number of students who answer question x and question y right.
- The number of students who answer question x right and question y wrong.
- The number of students who answer question x wrong and question y right.
- The number of students who answer question x and question y wrong.

After that, the expected number of students for each of the above cases is found using the IRT models. Finally, chi-square (x^2) is measured between the observed and expected values to determine whether a question violates the null hypothesis or not. If the value obtained for x^2 is greater than 3.841 at a degree of freedom (DF) equals one, then according to the chi-square distribution table, this reflects a local dependence issue.

Appendix F

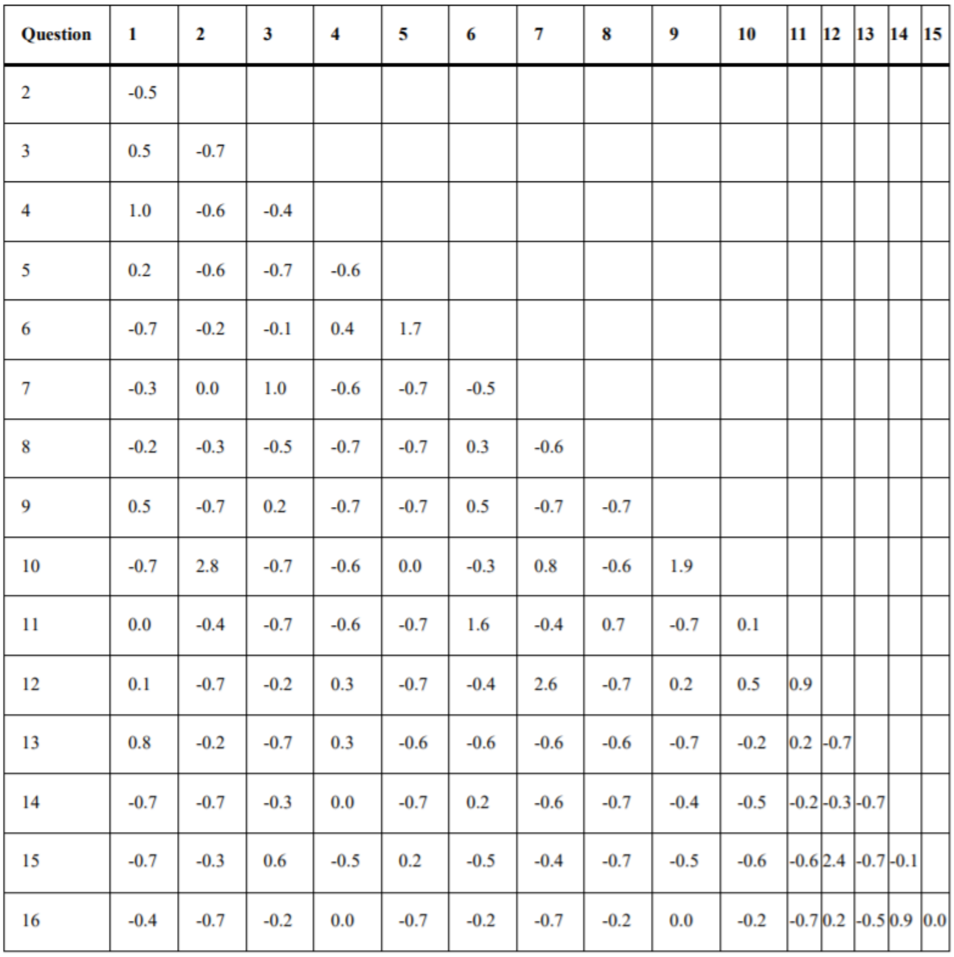
Local Dependence Chi-square Test

This appendix illustrates the Local Dependence chi-square (LD x^2) test for test-one, test-two and test-three.



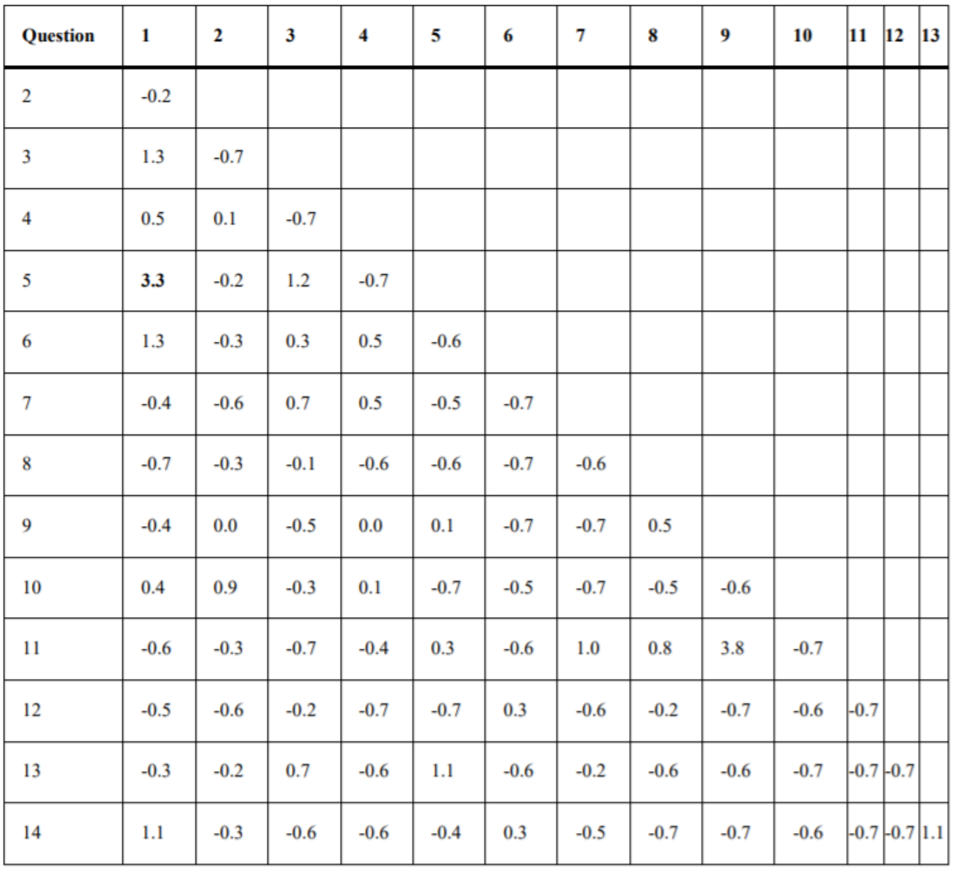
Question	1	2	3	4	5	6	7	8	9	10	11	12	13
2	-0.7												
3	-0.4	-0.7											
4	0.2	-0.4	0.4										
5	0.4	-0.7	-0.7	-0.5									
6	-0.4	0.2	-0.3	-0.6	-0.7								
7	-0.5	-0.7	-0.6	-0.7	-0.6	-0.5							
8	1.1	-0.6	0.0	-0.6	-0.6	-0.7	-0.5						
9	0.1	-0.7	-0.3	-0.7	-0.3	0.1	-0.6	-0.5					
10	-0.7	-0.7	-0.3	-0.2	-0.7	0.0	-0.6	0.6	-0.7				
11	-0.2	-0.5	-0.7	-0.7	**3.8**	-0.7	-0.7	-0.6	-0.6	-0.6			
12	-0.3	-0.7	-0.6	-0.7	-0.2	-0.7	-0.7	-0.7	-0.5	-0.5	0.9		
13	-0.3	-0.7	-0.7	-0.6	0.6	-0.3	-0.7	-0.6	-0.5	-0.7	-0.4	-0.2	
14	-0.7	-0.2	-0.6	-0.7	-0.3	-0.6	-0.7	-0.7	-0.7	-0.5	-0.5	-0.7	0.7

Figure F.1: LD x^2 statistics for 14 questions in test-one (Degree of Freedom =1)



Question	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	-0.5														
3	0.5	-0.7													
4	1.0	-0.6	-0.4												
5	0.2	-0.6	-0.7	-0.6											
6	-0.7	-0.2	-0.1	0.4	1.7										
7	-0.3	0.0	1.0	-0.6	-0.7	-0.5									
8	-0.2	-0.3	-0.5	-0.7	-0.7	0.3	-0.6								
9	0.5	-0.7	0.2	-0.7	-0.7	0.5	-0.7	-0.7							
10	-0.7	2.8	-0.7	-0.6	0.0	-0.3	0.8	-0.6	1.9						
11	0.0	-0.4	-0.7	-0.6	-0.7	1.6	-0.4	0.7	-0.7	0.1					
12	0.1	-0.7	-0.2	0.3	-0.7	-0.4	2.6	-0.7	0.2	0.5	0.9				
13	0.8	-0.2	-0.7	0.3	-0.6	-0.6	-0.6	-0.6	-0.7	-0.2	0.2	-0.7			
14	-0.7	-0.7	-0.3	0.0	-0.7	0.2	-0.6	-0.7	-0.4	-0.5	-0.2	-0.3	-0.7		
15	-0.7	-0.3	0.6	-0.5	0.2	-0.5	-0.4	-0.7	-0.5	-0.6	-0.6	2.4	-0.7	-0.1	
16	-0.4	-0.7	-0.2	0.0	-0.7	-0.2	-0.7	-0.2	0.0	-0.2	-0.7	0.2	-0.5	0.9	0.0

Figure F.2: LD x^2 statistics for 16 questions in test-two



Question	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
2	-0.2												
3	1.3	-0.7											
4	0.5	0.1	-0.7										
5	**3.3**	-0.2	1.2	-0.7									
6	1.3	-0.3	0.3	0.5	-0.6								
7	-0.4	-0.6	0.7	0.5	-0.5	-0.7							
8	-0.7	-0.3	-0.1	-0.6	-0.6	-0.7	-0.6						
9	-0.4	0.0	-0.5	0.0	0.1	-0.7	-0.7	0.5					
10	0.4	0.9	-0.3	0.1	-0.7	-0.5	-0.7	-0.5	-0.6				
11	-0.6	-0.3	-0.7	-0.4	0.3	-0.6	1.0	0.8	3.8	-0.7			
12	-0.5	-0.6	-0.2	-0.7	-0.7	0.3	-0.6	-0.2	-0.7	-0.6	-0.7		
13	-0.3	-0.2	0.7	-0.6	1.1	-0.6	-0.2	-0.6	-0.6	-0.7	-0.7	-0.7	
14	1.1	-0.3	-0.6	-0.6	-0.4	0.3	-0.5	-0.7	-0.7	-0.6	-0.7	-0.7	1.1

Figure F.3: LD x^2 statistics for 14 questions in test-three

Appendix G

Students' instruction Sheet

I am inviting everybody registered in Computer Networks (COMP28411) and Data Networking (EEEN30024) courses to take part in a research study aimed at providing better formative feedback in Moodle Virtual Learning Environment. Students who volunteer to be part of the study will be asked to answer three short quizzes; after you finish them, you will receive an email to confirm that you have finished the experiment.

If you would like to participate, please read the description below: Students who volunteer to be part of the study will be asked to do the following:

1. Go to `http://trove.cs.man.ac.uk/moodle/`
2. Sign in using your university username (e.g., mbaxxxxx) and Password
3. Go to the Transport Layer course
4. Answer the following three tests:
   (a) Answer ”Transport Layer test one”.
   (b) Answer ”Transport Layer test two”. The test will be available online after you finish test one.
   (c) Answer ”Transport Layer test three”. The test will be available online after you finish test two.

Appendix H

Students’ Questionnaire

This questionnaire aims to assess students’ perspective about the auto-generated feedback usefulness and clarity.

Q1: I found the provided feedback useful

- Agree
- Neutral
- Disagree

Q2: I found the provided feedback easy to read

- Agree
- Neutral
- Disagree

Q3: The provided feedback in test two helped me answer questions in test three

- Agree
- Neutral
- Disagree

Appendix I

Teachers' Questionnaire

This questionnaire is a 5-point likert scale and it aims to assess teachers perspective about the questions clarity, the feedback clarity, and the feedback pedagogical content and it can be found in the following:

Q1: The generated questions are easy to read.

Q2: The generated feedback is easy to read.

Q3: I found the feedback pedagogical content reasonable and related to the question (The feedback pedagogical content is related to the questions asked).

Q4: The feedback is providing students with different information at different level of detail. I can see different types of feedback.

Q5: I believe that feedback auto-generation saves time in terms of writing different feedback content to students with different background knowledge and performance.

Q6: I believe that feedback auto-generation saves time in terms of writing different feedback content to different MCQ options.

Q7: I believe that feedback auto-generation saves time in terms of writing different feedback content to different expected answers. This is for True/false and short answer questions.

Q8: In general, the feedback provided is useful to students.

References

Abedalaziz, N. and Leng, C. H. (2013). The relationship between ctt and irt approaches in analyzing item characteristics.

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723.

Al-Yahya, M. (2011). Ontoque: a question generation engine for educational assesment based on domain ontologies. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, pages 393–395. IEEE.

Al-Yahya, M. (2014). Ontology-based multiple choice question generation. *The Scientific World Journal*, 2014.

AlYahya, M., Al-Khalifa, H., Bahanshal, A., Al-Odah, I., and Al-Helwah, N. (2010). An ontological model for representing semantic lexicons: an application on time nouns in the holy quran. *Arabian Journal for Science and Engineering*, 35(2):21–35.

Al-Yahya, M., George, R., and Alfaries, A. (2015). Ontologies in e-learning: Review of the literature. pages 67–84.

Alagumalai, S. and Curtis, D. D. (2005). *Classical test theory*, pages 1–14. Springer.

Allemang, D. and Hendler, J. (2011). *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier.

Alsubait, T., Parsia, B., and Sattler, U. (2014). Generating multiple choice questions from ontologies: lessons learnt. In *The 11th OWL: Experiences and Directions Workshop (OWLED2014)*.

Ana, G. S. and Macario, D.-S. J. (2009). Designing feedback to support language acquisition using the ingenio authoring tool. *Procedia-Social and Behavioral Sciences*, 1(1):1239–1243.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., and Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207.

Anderson, L. W. and Sosniak, L. A. (1994). Bloom's taxonomy: a forty-year retrospective. ninety-third yearbook of the national society for the study of education, pt. 2.

Arnold, K., Gosling, J., and Holmes, D. (1996). *The Java programming language*, volume 2. Addison-wesley Reading.

Arroyo, I., Beck, J. E., Beal, C. R., Wing, R., and Woolf, B. P. (2001). Analyzing students response to help provision in an elementary mathematics intelligent tutoring system. In *Papers of the AIED-2001 Workshop on Help Provision and Help Seeking in Interactive Learning Environments*, pages 34–46. Citeseer.

Arroyo, I., Beck, J. E., Woolf, B. P., Beal, C. R., and Schultz, K. (2000). Macroadapting animalwatch to gender and cognitive differences with respect to hint interactivity and symbolism. In *Intelligent Tutoring Systems*, pages 574–5831. Springer.

Arroyo, I., Woolf, B. P., Cooper, D. G., Burleson, W., and Muldner, K. (2011). The impact of animated pedagogical agents on girls' and boys' emotions, attitudes, behaviors and learning. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, pages 506–510. IEEE.

Baker, F. B. (2001). *The basics of item response theory*. ERIC.

Bangert-Drowns, R. L., Kulik, C.-L. C., Kulik, J. A., and Morgan, M. (1991). The instructional effect of feedback in test-like events. *Review of educational research*, 61(2):213–238.

Bedford, S. and Price, G. (2007). A study into the use of computer aided assessment to enhance formative assessment during the early stages of undergraduate chemistry courses.

Bishop, P., Downes, J., Netcoh, S., DeMink-Carthew, J., Farber, K., LeGeros, L., and Stokes, T. (2018). Middle grades teachersdispositions in personalized learning environments. *Preparing Middle Level Educators for 21st Century Schools*, page 229.

Black, P. and Wiliam, D. (1998). *Inside the black box: Raising standards through classroom assessment*. Granada Learning.

Black, P. and Wiliam, D. (2009). Developing the theory of formative assessment. *Educational Assessment, Evaluation and Accountability (formerly: Journal of Personnel Evaluation in Education)*, 21(1):5–31.

Blackboard (1997). Blackboard. Accessed 10/2011.

Bloom, B. S., Examiners, C. o. C., and University (1956). *Taxonomy of educational objectives*, volume 1. David McKay New York.

Bock, R. and Aitkin, M. (1982). Marginal maximum likelihood estimation of item parameters. *Psychometrika*, 47(3):369–369.

Bodenreider, O. (2004). The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32:267–270.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

Borst, W. N. (1997). *Construction of engineering ontologies for knowledge sharing and reuse*. Universiteit Twente.

Boyce, S. and Pahl, C. (2007). Developing domain ontologies for course content. *Journal of Educational Technology & Society*, 10(3).

Brank, J., Grobelnik, M., and Mladeni, D. (2005). A survey of ontology evaluation techniques.

Brank, J., Grobelnik, M., and Mladeni, D. (2007). *Automatic evaluation of ontologies*. Springer.

Brewster, C., Alani, H., Dasmahapatra, S., and Wilks, Y. (2004). Data driven ontology evaluation.

Brickley, D. and Guha, R. V. (2000). Resource description framework (rdf) schema specification 1.0: W3c candidate recommendation 27 march 2000.

Brown, E., Gibbs, G., and Glover, C. (2003). Evaluation tools for investigating the impact of assessment regimes on student learning. *Bioscience Education*, (2).

Brown, E. and Glover, C. (2005). Refocusing written feedback. In *13th Improving Student Learning Symposium, Imperial College, London, UK*, pages 5–7.

Brown, E. and Glover, C. (2006). Evaluating written feedback. *Innovative assessment in higher education*, pages 81–91.

Brown, E., Glover, C., Freake, S., and Stevens, V. (2004). Evaluating the effectiveness of written feedback as an element of formative assessment in science. In *Proceedings of the Improving Student Learning: Diversity and Inclusivity Symposium*.

Brown, J. D. (1996). *Testing in language programs*. Prentice Hall Regents New Jersey.

CAA (2002). Assessment of higher-order skills.

Cai, L., Thissen, D., and du Toit, S. (2011). Irtpro for windows [computer software]. *Lincolnwood, IL: Scientific Software International*.

Castellanos-Nieves, D., Fernndez-Breis, J. T., Valencia-Garca, R., Martnez-Bjar, R., and Iniesta-Moreno, M. (2011). Semantic web technologies for supporting learning assessment. *Information Sciences*, 181(9):1517–1537.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate behavioral research*, 1(2):245–276.

Chen, W.-H. and Thissen, D. (1997). Local dependence indexes for item pairs using item response theory. *Journal of Educational and Behavioral Statistics*, 22(3):265–289.

Chou, Y.-T. and Wang, W.-C. (2010). Checking dimensionality in item response models with principal component analysis on standardized residuals. *Educational and Psychological Measurement*, 70(5):717–731.

Cocea, M. (2011). *User modelling and adaptation in exploratory learning*. Thesis, Birkbeck (University of London).

Cohen, L., Manion, L., and Morrison, K. (2013). *Research methods in education*. Routledge.

Comer, J. S. and Kendall, P. C. (2013). *The Oxford Handbook of Research Strategies for Clinical Psychology*. Oxford University Press.

Cooper, H., Hedges, L. V., and Valentine, J. C. (2009). *The handbook of research synthesis and meta-analysis*. Russell Sage Foundation.

Corkins, J. (2009). *The Psychometric Refinement of the Materials Concept Inventory (MCI)*. ProQuest.

Crocker, L. and Algina, J. (1986). *Introduction to classical and modern test theory*. ERIC.

Cronbach, L. J. and Shavelson, R. J. (2004). My current thoughts on coefficient alpha and successor procedures. *Educational and psychological measurement*, 64(3):391–418.

Csomai, A. and Mihalcea, R. F. (2006). *Creating a Testbed for the Evaluation of Automatically Generated Back-of-the-book Indexes*, pages 429–440. Springer.

Cubric, M. and Tosic, M. (2011). Towards automatic generation of e-assessment using semantic web technologies. *International Journal of e-Assessment*, pages 1–9.

Dancey, C. P. and Reidy, J. (2004). *Statistics Without Maths for Psychology: Using Spss for Windows*. Prentice-Hall, Inc.

De Ayala, R. J. (2009). *Theory and practice of item response theory*. Guilford Publications.

Demaidi, M. (2013). Interview with rubner, g.

Demaidi, M. (2014). Email to bedford, s.

Demaidi, M. (2015a). Email to carpenter, liz.

Demaidi, M. (2015b). Email to fernndez-toro, m.

Demaidi, M. (2015c). Email to glover, c.

Demaidi, M. and Filer, N. (2012). Wireless ray tracing educational land. In *Proceedings of INTED2012 Conference*, pages 0402–0411. International Association for Technology, Education and Development (IATED).

DeMars, C. E. (2012). Confirming testlet effects. *Applied Psychological Measurement*, 36(2):104–121.

Dermo, J. and Carpenter, L. (2011). e-assessment for learning: can online selected response questions really provide useful formative feedback. In *Proceedings of CAA 2011 International Conference, Southampton*.

Ding, L. and Beichner, R. (2009). Approaches to data analysis of multiple-choice questions.

Ding, L., Chabay, R., Sherwood, B., and Beichner, R. (2006). Evaluating an electricity and magnetism assessment tool: Brief electricity and magnetism assessment. *Physical review special Topics-Physics education research*, 2(1).

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J. (2004). Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 652–659. ACM.

Doran, R. L. (1980). *Basic measurement and evaluation of science instruction*. National Science Teachers Association Washington, DC.

Downes, S. (2007). Learning networks in practice.

Duboc, J.-R. (2013). *Dynamic feedback generation in virtual patients using semantic web technologies*. Thesis, University of Southampton, Southampton, UK.

Duboc, J.-R., Weal, M., Davis, H., and Choi, S. (2011). Automatic feedback generation in virtual patients using semantic web technologies. pages 1–11.

Ebel, R. (1979). *Essentials of Educational Measurement*. Prentice-Hall.

Erguven, M. (2014). Two approaches to psychometric process: Classical test theory and item response theory. *Journal of Education*, 2(2):23–30.

Fancher, E. L. (2013). *Comparison of Methods of Analysis for Pretest and Posttest Data*. Thesis, University of Georgia.

Felder, R. M. and Brent, R. (1997). Objectively speaking. *Chemical Engineering Education*, 31:178–179.

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

Ferguson, G. (1949). On the theory of test development. *Psychometrika*, 14:61–68.

Fernndez-Toro, M., Truman, M., and Walker, M. (2013). Are the principles of effective feedback transferable across disciplines? a comparative study of written assignment feedback in languages and technology. *Assessment & Evaluation in Higher Education*, 38(7):816–830.

Field, A. (2013). *Discovering statistics using IBM SPSS statistics*. Sage.

Fotaris, P., Mastoras, T., Mavridis, I., and Manitsaris, A. (2010). *Extending LMS to support IRT-based assessment test calibration*, pages 534–543. Springer.

Foundation, A. S. Jena ontology api. Accessed 12/2011.

Foundation, A. S. (2010). Welcome to apache opennlp. Accessed 12/2010.

Frantzi, K., Ananiadou, S., and Mima, H. (2000). Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130.

Frutos-Morales, F., Snchez-Vera, M., Castellanos-Nieves, D., Esteban-Gil, A., Cruz-Corona, C., Prendes-Espinosa, M., and Fernndez-Breis, J. (2010). An extension of the oele platform for generating semantic feedback for students and teachers. *Procedia-Social and Behavioral Sciences*, 2(2):527–531.

Ganapathi, G., Lourdusamy, R., and Rajaram, V. (2011). Towards ontology development for teaching programming language. In *World Congress on Engineering*.

GLOBAL, I. (2012). Ims question & test interoperability. Accessed 1/2012.

Glover, C. and Brown, E. (2006). Written feedback for students: too much, too detailed or too incomprehensible to be effective? *Bioscience Education*, (7).

Goguadze, G. (2009). *Representation for interactive exercises*, pages 294–309. Springer.

Goldin, I., Narciss, S., Foltz, P., and Bauer, M. (2017). New directions in formative feedback in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 27(3):385–392.

Gouli, E., Gogoulou, A., Papanikolaou, K. A., and Grigoriadou, M. (2006). An adaptive feedback framework to support reflection, guiding and tutoring. *Advances in web-based education: Personalized learning environments*, pages 178–202.

Group, O. W. (2004). Owl web ontology language. Accessed 11/2011.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220.

Grubisic, A. (2012). *Adaptive students knowledge acquisition model in elearningsystems*. Thesis, University of Zagreb, Zagreb, Croatia.

Grubisic, A., Stankov, S., and itko, B. (2013). Stereotype student model for an adaptive e-learning system. In *ICIIS 2013: International Conference on Information and Intelligent Systems*.

Gler, N., Uyank, G. K., and Teker, G. T. (2014). Comparison of classical test theory and item response theory in terms of item parameters. *European Journal of Research on Education*, 2(1):1–6.

Haladyna, T. M. (1994). *Developing and validating multiple-choice test items / Thomas M. Haladyna*. Hillsdale, NJ ; Hove, UK : Erlbaum, Hillsdale, NJ ; Hove, UK.

Hambleton, R. K. (1991). *Fundamentals of item response theory*, volume 2. Sage publications.

Hambleton, R. K. and Swaminathan, H. (1985). *Item response theory: Principles and applications*, volume 7. Springer.

Hankins, M. (2007). Questionnaire discrimination:(re)-introducing coefficient . *BMC Medical Research Methodology*, 7(1).

Hardouin, J.-B. and Mesbah, M. (2004). Clustering binary variables in subscales using an extended rasch model and akaike information criterion. *Communications in Statistics-Theory and Methods*, 33(6):1277–1294.

Hasmy, A. (2014). Compare unidimensional & multidimensional rasch model for test with multidimensional construct and items local dependence. *Journal of Education and Learning (EduLearn)*, 8(3):187–194.

Hattie, J. and Timperley, H. (2007). The power of feedback. *Review of educational research*, 77(1):81–112.

Hooshyar, D., Ahmad, R. B., Yousefi, M., Fathi, M., Horng, S.-J., and Lim, H. (2016). Applying an online game-based formative assessment in a flowchart-based intelligent tutoring system for improving problem-solving skills. *Computers & Education*, 94:18–36.

Huggett, M. and Rasmussen, E. (2013). Using digital book metrics for navigation and browsing.

James, G., Witten, D., and Hastie, T. (2014). An introduction to statistical learning: With applications in r.

jatetoolkit (2011). jatetoolkit. Accessed 5/2012.

Jeong, D. H., Ziemkiewicz, C., Ribarsky, W., Chang, R., and Center, C. V. (2009). Understanding principal component analysis using a visual analytics tool. *Charlotte Visualization Center, UNC Charlotte*.

Jones, A. (2009). *Using the right tool for the job: An analysis of item selection statistics for criterion-referenced tests*. ProQuest.

Jones, M. and Alani, H. (2006). Content-based ontology ranking.

Jonquet, C., Musen, M. A., and Shah, N. H. (2010). Building a biomedical ontology recommender web service. *J Biomed Semantics*, 1(Suppl 1).

Kang, T. and Cohen, A. S. (2007). Irt model selection methods for dichotomous items. *Applied Psychological Measurement*, 31(4):331–358.

Kazi, H., Haddawy, P., and Suebnukarn, S. (2010). Leveraging a domain ontology to increase the quality of feedback in an intelligent tutoring system. In *Intelligent Tutoring Systems*, pages 75–84. Springer.

Kazi, H., Haddawy, P., and Suebnukarn, S. (2012). Employing umls for generating hints in a tutoring system for medical problem-based learning. *Journal of biomedical informatics*, 45(3):557–565.

Kazi, H., Haddawy, P., and Suebnukarn, S. (2013). Clinical reasoning gains in medical pbl: an umls based tutoring system. *Journal of Intelligent Information Systems*, 41(2):269–284.

Keuning, H., Jeuring, J., and Heeren, B. (2016). Towards a systematic review of automated feedback generation for programming exercises. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 41–46. ACM.

Kline, P. (1986). *A handbook of test construction: Introduction to psychometric design*. Methuen.

Kline, P. (2013a). *Handbook of psychological testing*. Routledge.

Kline, P. (2013b). *Personality: The psychometric view*. Routledge.

Knoth, P., Schmidt, M., Smrz, P., and Zdrahal, Z. (2009). Towards a framework for comparing automatic term recognition methods.

Kouneli, A., Solomou, G., Pierrakeas, C., and Kameas, A. (2012). *Modeling the knowledge domain of the java programming language as an ontology*, pages 152–159. Springer.

Krathwohl, D. R. (2002). A revision of bloom's taxonomy: An overview. *Theory into practice*, 41(4):212–218.

Kurose, J. F. and Ross, K. W. (2013). *Computer Networking: A Top-Down Approach*. Pearson Education, 6th edition.

Lee, M.-C., Ye, D. Y., and Wang, T. I. (2005). Java learning object ontology. In *Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on*, pages 538–542. IEEE.

Library, P. O. (2013). Protege ontology library. Accessed 5/2013.

Lopes, L., Vieira, R., Finatto, M. J., and Martins, D. (2010). Extracting compound terms from domain corpora. *Journal of the Brazilian Computer Society*, 16(4):247–259.

Lord, F. (1952). A theory of test scores. *Psychometric monographs*.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems*. Routledge.

lorecommender (2013). lorecommender. Accessed 9/2013.

Lozano-Tello, A. and Gmez-Prez, A. (2004). Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, 2(15):1–18.

Maedche, A., Motik, B., Silva, N., and Volz, R. (2002). *Mafraa mapping framework for distributed ontologies*, pages 235–250. Springer.

Maedche, A. and Staab, S. (2002). *Measuring similarity between ontologies*, pages 251–263. Springer.

Manning, C. D., Raghavan, P., and Schutze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Martnez-Romero, M., Vzquez-Naya, J. M., Pereira, J., and Pazos, A. (2012). *A multi-criteria approach for automatic ontology recommendation using collective knowledge*, pages 89–103. Springer.

Martnez-Romero, M., Vzquez-Naya, J. M., Pereira, J., and Pazos, A. (2014). Bioss: A system for biomedical ontology selection. *Computer methods and programs in biomedicine*, 114(1):125–140.

Mason, B. J. and Bruning, R. H. (2001). Providing feedback in computer-based instruction: What the research tells us. Accessed 10/2011.

Matuszek, C., Cabral, J., Witbrock, M. J., and DeOliveira, J. (2006). An introduction to the syntax and content of cyc. In *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pages 44–49. Citeseer.

Mealling, M. and Denenberg, R. (2002). Report from the joint w3c/ietf uri planning interest group: Uniform resource identifiers (uris), urls, and uniform resource names (urns): Clarifications and recommendations.

Meyer, C. M. and Gurevych, I. (2012). *Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography*. Oxford University Press.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mitkov, R., An Ha, L., and Karamanis, N. (2006). A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(02):177–194.

Mitkov, R. and Ha, L. A. (2003). Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*, pages 17–22. Association for Computational Linguistics.

Mitkov, R., Ha, L. A., Varga, A., and Rello, L. (2009). Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 49–56. Association for Computational Linguistics.

Mitrovic, A. (2012). Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User Modeling and User-Adapted Interaction*, 22(1-2):39–72.

Mitrovic, A. and Martin, B. (2000). Evaluating the effectiveness of feedback in sql-tutor.

Mitrovic, A. and Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language.

Mitrovi, A. (1998). Experiences in implementing constraint-based modeling in sql-tutor. In *Intelligent Tutoring Systems*, pages 414–423. Springer.

Moodle (2011). Moodle. Accessed 10/2011.

Moodle (2012). Moodle xml format. Accessed 1/2012.

Mory, E. H. (2004). Feedback research revisited. *Handbook of research on educational communications and technology*, 2:745–783.

Murugan.R, S., Bala, P., and Dr.G.Aghila (2013). An ontology for exploring knowledge in computer networks. *International Journal on Computational Sciences & Applications (IJCSA)*, 3(4):13–21.

Nakagawa, H. (2001). Experimental evaluation of ranking and selection methods in term extraction. *Bourigault D, LHomme M.-C., Jacquemin C.(d.), Recent advances in computational terminology, John Benjamins Publishing Company*, pages 303–326.

Narciss, S. (2013). Designing and evaluating tutoring feedback strategies for digital learning environments on the basis of the interactive tutoring feedback model. *Digital Education Review*, (23):7–26.

Narciss, S. and Huth, K. (2004). How to design informative tutoring feedback for multimedia learning. *Instructional design for multimedia learning*, pages 181–195.

Narciss, S. and Huth, K. (2006). Fostering achievement and motivation with bug-related tutoring feedback in a computer-based training for written subtraction. *Learning and Instruction*, 16(4):310–322.

Narciss, S., Sosnovsky, S., Schnaubert, L., Andrs, E., Eichelmann, A., Goguadze, G., and Melis, E. (2014). Exploring feedback and student characteristics relevant for personalizing feedback strategies. *Computers & Education*, 71:56–76.

of Manchester, U. (2012a). Classroom response system. Accessed 1/2012.

of Manchester, U. (2012b). Digital wireless communications and networks course. Accessed 1/2012.

of Manchester, U. (2013a). Computer networks course. Accessed 5/2013.

of Manchester, U. (2013b). Computer networks course. Accessed 1/2013.

of Manchester, U. (2013c). Data networking course. Accessed 4/2013.

of Manchester, U. (2013d). Data networking lectures slides. Accessed 6/2013.

OpenCyc (2011). Opencyc. Accessed 11/2011.

Paek, I. and Han, K. T. (2013). Irtpro 2.1 for windows (item response theory for patient-reported outcomes). *Applied Psychological Measurement*, pages 242–252.

Pahl, C. and Holohan, E. (2009). Applications of semantic web technology to support learning content development. *Interdisciplinary Journal of E-Learning & Learning Objects*, 5.

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

Papasalouros, A., Kotis, K., and Kanaris, K. (2011). Automatic generation of tests from domain and multimedia ontologies. *Interactive Learning Environments*, 19(1):5–23.

Pardo, A., Jovanovic, J., Dawson, S., Gašević, D., and Mirriahi, N. (2017). Using learning analytics to scale the provision of personalised feedback. *British Journal of Educational Technology*.

Passier, H. and Jeuring, J. (2004). Ontology based feedback generation in design-orientated e-learning systems. *Proceedings of the IADIS International Conference on e-Society*, pages 992–996.

Pazienza, M. T., Pennacchiotti, M., and Zanzotto, F. M. (2005). *Terminology extraction: an analysis of linguistic and statistical approaches*, pages 255–279. Springer.

Porzel, R. and Malaka, R. (2004). A task-based approach for ontology evaluation. In *ECAI Workshop on Ontology Learning and Population, Valencia, Spain*. Citeseer.

Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.

Price, M., Handley, K., Millar, J., and O'Donovan, B. (2010). Feedback: all that effort, but what is the effect? *Assessment & Evaluation in Higher Education*, 35(3):277–289.

Reckase, M. D. (2009). *Multidimensional item response theory*. Springer.

Rector, A., Rogers, J., Zanstra, P., and Van Der Haring, E. Opengalen: open source medical terminology and tools. In *AMIA Annual Symposium Proceedings*, volume 2003, page 982. American Medical Informatics Association.

Romero, M. M., Vzquez-Naya, J. M., Munteanu, C. R., Pereira, J., and Pazos, A. (2010). *An approach for the automatic recommendation of ontologies using collaborative knowledge*, pages 74–81. Springer.

Rospocher, M., Tonelli, S., Serafini, L., and Pianta, E. (2012). Corpus-based terminological evaluation of ontologies. *Applied Ontology*, 7(4):429–448.

Rubner, G. (2012). mbclick-an electronic voting system that returns individual feedback. In *Wireless, Mobile and Ubiquitous Technology in Education (WMUTE), 2012 IEEE Seventh International Conference on*, pages 221–222. IEEE.

S, M. D. M., Nchez-Vera, Tom, J., Fern, S., Ndez-Breis, Castellanos-Nieves, D., Frutos-Morales, F., and Prendes-Espinosa, A. P. (2012). Semantic web technologies for generating feedback in online assessment environments. *Know.-Based Syst.*, 33:152–165.

Sadler, D. R. (1989). Formative assessment and the design of instructional systems. *Instructional science*, 18(2):119–144.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Schmidt, K. M. and Embretson, S. E. (2003). Item response theory and measuring abilities. *Handbook of psychology*.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Shute, V. J. (2008). Focus on formative feedback. *Review of educational research*, 78(1):153–189.

Stanek III, E. J. (1988). Choosing a pretest-posttest analysis. *The American Statistician*, 42(3):178–183.

Stanford (2012). Pizza ontology. Accessed 4/2012.

Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1):161–197.

Tarp, S. (2009). *Lexicography at a crossroads: dictionaries and encyclopedias today, lexicographical tools tomorrow*, volume 90. Peter Lang.

Tarus, J. K., Niu, Z., and Mustafa, G. (2018). Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, 50(1):21–48.

Thorndike, R. L. and Hagen, E. (1961). Measurement and evaluation in psychology and education.

Toland, M. D. (2014). Practical guide to conducting an item response theory analysis. *The Journal of Early Adolescence*, 34:120–151.

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley, Reading, Mass. ; London.

Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(02):93–136.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221.

Vasilyeva, E., De Bra, P., Pechenizkiy, M., and Puuronen, S. (2008a). Tailoring feedback in online assessment: influence of learning styles on the feedback preferences and elaborated feedback effectiveness. In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference on*, pages 834–838. IEEE.

Vasilyeva, E., Pechenizkiy, M., and De Bra, P. (2008b). Analysis of feedback authoring possibilities in web-based learning systems. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, volume 2008, pages 3981–3986.

W3C (2004a). Owl web ontology language. Accessed 2/2012.

W3C (2004b). Rdf/xml syntax specification (revised). Accessed 11/2011.

W3C (2012). W3c xml schema definition language. Accessed 12/2011.

W3C (2014). Rdf schema 1.1. Accessed 11/2011.

Walker, M. (2009). An investigation into written comments on assignments: do students find them usable? *Assessment & Evaluation in Higher Education*, 34(1):67–78.

Wong, W., Liu, W., and Bennamoun, M. (2008). Determining the unithood of word sequences using a probabilistic approach. *arXiv preprint arXiv:0810.0139*.

Woolf, B. P., Arroyo, I., Muldner, K., Burleson, W., Cooper, D. G., Dolan, R., and Christopherson, R. M. (2010). The effect of motivational learning companions on low achieving students and students with disabilities. In *Intelligent Tutoring Systems*, pages 327–337. Springer.

Yeomans, K. A. and Golder, P. A. (1982). The guttman-kaiser criterion as a predictor of the number of common factors. *The Statistician*, pages 221–229.

Zhang, F. and Lidbury, B. A. (2013). Evaluating a genetics concept inventory. *Bioinformatics: Concepts, Methodologies, Tools, and Applications*, pages 29–41.

Zhang, Z., Iria, J., Brewster, C. A., and Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms.