Generating Multiple Choice Questions From Ontologies: Lessons Learnt

Tahani Alsubait, Bijan Parsia, and Uli Sattler

School of Computer Science, The University of Manchester, United Kingdom
{alsubait,bparsia,sattler}@cs.man.ac.uk

Abstract. Ontologies with potential educational value are available in different domains. However, it is still unclear how such ontologies can be exploited to generate useful instructional content, e.g., assessment questions. In this paper, we present an approach to automatically generate multiple-choice questions from OWL ontologies. We describe a psychologically-plausible theory to control the difficulty of questions. Our contributions include designing a protocol to evaluate the characteristics of the generated questions, in particular, question difficulty. We report on our experience on generating questions from ontologies and present promising results of evaluating our difficulty-control theory.

1 Introduction

A question in which a set of plausible answers are offered to a student to choose from is called a Multiple Choice Question (MCQ). Providing a set of plausible answers might or might not make the question easier for the student. However, preparing reasonably good alternative answers definitely requires more time and effort from the question designer. This is why we primarily focus on developing methods to automatically generate this particular type of questions.

Developing automatic methods for Question Generation (QG) can alleviate the burden of both paper-and-pencil and technology-aided assessments. Of particular interest are large-scale tests such as nation-wide standardised tests and tests delivered as part of Massive Open Online Courses (MOOCS). Typically, these tests consists mainly of MCQs [32]. Economically speaking, it is reported [1] that a large amount of money is increasingly spent on large-scale testing (e.g., US spendings doubled from $165 million in 1996 to $330 million in 2000). Although there is no evidence that automatic QG can help to reduce these spendings, it is expected that the time spent on test preparation could be utilised in better ways. In addition, different modes of delivery (e.g., static, adaptive) can benefit from automatic QG. In fact, one of the promising applications of QG is the delivery of questions that match the estimated abilities of test takers in order to reduce the time spent by each test taker on the test [19].

Abstractly speaking, a QG system takes, as input, a knowledge source and some specifications describing the questions to be generated. As output, it produces a reasonable number of questions which assess someones understanding of that knowledge and which, of course, adhere to the given specifications. These specifications can include, for example, the format of the question and its difficulty.

Generally, questions answered correctly by 30%-90% of students are preferred to those answered by more than 90% or less than 30% of students [11]. But, a well-balanced test needs questions of all kinds of difficulties, in suitable proportions. To construct a balanced test, initially, the test developer needs to predict how certain students will perform on the items of that test. However, this is a rather difficult process [6] and there is evidence that teachers do not usually make good predictions [20]. In addition, automatic estimations of questions' difficulty can help to advance research on adaptive assessment systems which usually rely on training data to estimate the difficulty [19]. As a consequence, it is necessary for automatic QG systems to be able to control the difficulty of the generated questions, although it is rather challenging. We have addressed this challenge by developing a novel, similarity-based theory of controlling MCQ difficulty [4]. In this paper, we empirically examine whether varying the similarity between the correct and wrong answers of each question can vary the difficulty of questions.

The main goal of this study is to answer the following questions:

1. Can we control the difficulty of MCQs by varying the similarity between the key and distractors?
2. Can we generate a reasonable number of educationally useful questions?
3. How costly is ontology-based question generation, including the cost of developing/enhancing an ontology, computation cost and post-editing cost?

2 Ontology-based question generation

Two alternative sources are typically used for QG: unstructured text and ontologies.The QG workshop (2009) [30] has identified raw text as the most preferred knowledge source according to responses gathered from participants of the workshop. However, a drawback of text-based QG approaches is that they mostly generate shallow questions about explicit information as it is difficult to infer implicit relations using current NLP techniques [8, 14, 18, 22]. Many ontology-based QG approaches have been developed [9, 15, 26, 35, 36]. These approaches take advantage of reasoning services offered by ontology tools to generate questions about implicit knowledge. However, it remains to put ontology-based QG approaches into practise.

Ontologies with potential educational value are available in different domains such as Biology, Medicine, Geography, to name a few.[1] However, ontologies are not designed particularly for educational use. Thus, there is a challenge in generating useful instructional content from them. Of course, in some cases, there is a need to first build or improve an ontology for a specific subject matter before utilising it for QG. Thus, there is a trade-off between the efforts required to build and maintain an ontology and the overall advantage of single or multiple uses.

3 Multiple-choice questions

Assessment questions have been classified in a variety of ways. A common way is to classify questions to objective (e.g., MCQs or True/False questions) or subjective (e.g., essays or short answers). The essential features of objective tests

[1] For a list of ontology repositories, the reader is referred to: http://owl.cs.manchester.ac.uk/tools/repositories/

include their ability to assess a broad range of knowledge in an efficient way (compared to essays, for example). In addition, they are auto-gradable and a well-established means to test students' understanding. However, research suggests that objective questions are hard to prepare and require considerable time per question [11, 27, 31].

Definition 1 *An MCQ is a tuple $< S, K, D >$ consisting of the following parts:*

1. *Stem (S): A statement that introduces a problem to the student.*
2. *Key (K): The correct answer(s).*
3. *Distractors (D): A number of incorrect, yet plausible, answers.*

The structured format of MCQs can help in making them suitable for computerised generation. The automatic generation of MCQs in particular and assessment questions in general can help to resolve many issues in students' assessment. For example, constructing a bank of questions of known properties can help to eliminate cheating by facilitating the preparation of different tests with similar properties (e.g., item difficulty, content). Also, instead of using last years' exams as self-assessments, an instructor can generate exams that resemble original past exams with reasonable effort.

4 Running example

Consider the following ontology.

$Hospital \sqsubseteq HealthCareProvider$, $GPClinic \sqsubseteq HealthCareProvider$,
$University \sqsubseteq EducationProvider$, $School \sqsubseteq EducationProvider$,
$Registrar \sqsubseteq \exists worksIn.Hospital$, $GP \sqsubseteq \exists worksIn.GPClinic$,
$Teacher \sqsubseteq \exists worksIn.School$, $Instructor \sqsubseteq \exists worksIn.University$,
$LuckyPatient \sqsubseteq Patient \sqcap \exists marriedTo.(\exists worksIn.HealthCareProvider)$,
$Patient(Mark)$, $Instructor(Nancy)$,
$Registrar(David)$, $GP(Sara)$,
$treatedBy(Mark, Sara)$, $marriedTo(Mark, Sara)$,
$treatedBy(Nancy, Sara)$, $marriedTo(Nancy, David)$

A reasonable number of questions can be generated from the above ontology. The generation can involve two steps: (1) generating question candidates and (2) transforming the candidate question into a grammatically well-formed question. The second step is out of the scope of this research. However, for readability, we present below some stems that can be generated from the above ontology after making any necessary grammatical renderings.

1. Give an example of a health care provider.
2. What is a GP clinic?
3. Where does an instructor work?
4. Who is Mark married to?
5. Which one of the following definitions describe a lucky patient?
6. Instructor to University is as ..... to .....?
7. Nancy to David is as ..... to .....?

The above questions range from simple recall questions (e.g., 1-5) to questions that require some sort of reasoning (e.g., 6-7). For each of the above stems, it remains to specify a key and some suitable distractors. The challenge is to pick distractors that look like plausible answers to those students who do not know the actual answer. For example, for question 4, the answers are expected to be names of persons. Including other distractors, such as names of institutions, would help in making the correct answer stand out even for a low mastery student. So we need a mechanism to filter out obvious wrong answers. Moreover, the utilised mechanism should be able to select a set of distractors that make the question suitable for a certain level of difficulty.

After seeing an example to generate questions from what can be considered a toy ontology, we want to know what is the case in real ontologies which are can be big and rich. For example, the average number of axioms per ontology in BioPortal is 20,532 with a standard deviation of 115,163 and maximum number of 1,484,923 [17]. This suggests that a considerably large number of questions, but not necessarily good quality questions, can be generated from a single ontology. We investigate this by generating some questions from a selected ontologies from BioPortal. The detailed results can be found in [2, 3] suggesting that a massive number of questions can be generated from each ontology. Some of the questions that arise here are: Are these questions all good? Can we control their difficulty? And what does an ontology look like that is suitable for auto-generation of MCQs?

In short, ontology-based QG can be accomplished in different steps. First, one needs to find an ontology and possibly enhance it either on the logical level or maybe by adding annotations. Secondly, the questions can be computed. Finally, there is a need to filter the generated questions before administering them to real students.

5 Similarity-based MCQ generation

To generate pedagogically sound questions, we need a pedagogically plausible notion of similarity for selecting good distractors. Ideally, we want students' performance to correlate with their knowledge mastery (i.e., amount and quality of knowledge). This means that difficult questions are expected to be answered correctly by high mastery students only while easy questions are expected to be answered by both low and high mastery students.

The basic intuition is that offering a set of very similar answers makes it difficult to identify the correct answer; hence, decreasing the probability of knowing correct answers just because distractors are obviously wrong. Thus, to make the question more difficult, increase the degree of similarity between the key and distractors. And, to make it less difficult, decrease the degree of similarity.

For instance, we would expect the difficulty of question 4 above to increase by providing a list of GP names as distractors since the correct answer "Sara" is also a GP. So someone who knows that Mark is married to a GP would still need to know the exact name of that GP. This means that a student who knows more about the subject of the question, performs better.

The question that remains to be answered is how can we measure the similarity between the stem and distractors? The choice of similarity measures has

a direct influence on the overall QG process. However, designing similarity measures for ontologies is still a big challenge. Looking at existing similarity measures (e.g., [10, 21, 28, 29, 34]), we found that no off-the-shelf existing method satisfies our requirements. For example, some measures [28, 29, 34] are imprecise by definition as they only consider atomic subsumptions and ignore any complex subsumptions which can affect similarity computation. Other measures [10, 21] impose some requirements on the ontology that can be used for similarity computation (e.g., low expressivity, no cycles, availability of an $ABox$ or external corpus of annotated text). This has motivated us to develop a new family of similarity measures which can be used with any arbitrary OWL ontology. The basic rationale of the new measures is that similar concepts have more common and fewer distinguishing features. Details of the new similarity measures can be found in [5]. In what follows, we primarily use the measure $SubSim(\cdot)$ which, in addition to Class names, considers Subsuming class expressions that occur in the input ontology.

6 Empirical evaluation

6.1 Materials and methods

Equipment description the following machine was used for the experiments in this paper: Intel Quad-core i7 2.4GHz processor, 4 GB 1333 MHz DDR3 RAM, running Mac OS X 10.7.5. In addition to the following software: OWL API v3.4.4 [16] and FaCT++ [33].

Ontology development the Knowledge Representation and Reasoning course (COMP34512) is a third year course unit offered by The School of Computer Science at The University of Manchester. It covers various Knowledge Representation (KR) topics including Knowledge Acquisition (KA) techniques and KR formalisms. For the purposes of the experiment described in this section, a KA ontology (which models the KA part of the course) was developed from scratch.[2] This particular part of the course unit was chosen as it contains mainly declarative knowledge. Other parts of the course can be described as procedural knowledge which is not suitable to be modelled in an OWL ontology. Assessing student's understanding of declarative knowledge is an essential part of various tests.

A total of 9 hours were spent to build the first version of the ontology, excluding the time required to skim through the course contents since the ontology was not developed by the instructor who is in charge of the course unit. The *Protégé* 4 ontology editor was used for building the ontology.

Several refinements were applied to the ontology after discussing the ontology with an experienced knowledge modeller, among the authors of this paper, and getting useful feedback from her. The feedback session took around 2 hours and refinements took around 3 hours to be applied.

The resulting ontology, after applying these refinements, is an $\mathcal{SI}$ ontology consisting of 504 axioms. Among these are 254 logical axioms. Class and object property counts are 151 and 7 respectively with one inverse and one transitive object property.

[2] Can be accessed at: http://edutechdeveloper.com/MCQGen/examples/KA.owl

Question generation In [4], we have described a variety of MCQs targeting different levels of Bloom taxonomy (i.e., a classification of educational objectives) [7]. It remains here to determine which questions to be generated for the purpose of the current experiment. We choose to avoid questions which require complicated cognitive processing (not necessarily difficult ones). Questions at lower Bloom levels are more suitable for our current purposes as they require less administration time. A variety of memory-recall questions have been generated which we describe below. Questions that require higher cognitive skills (e.g., reasoning) have been examined before, for more details see Section 6.3.

A total of 913 questions have been generated from the KA ontology described above.[3] Among these are 633 easy questions and 280 difficult questions, see details below. Only 535 questions out of the 913 questions have at least 3 distractors (with 474 easy questions and 82 difficult questions). Out of these, we randomly select 50 questions for further evaluation by 3 reviewers. The 50 questions contain 5 easy and 5 difficult questions from 6 different question categories which are described below, where 2 categories contain only easy questions and one category contains a total of 5 difficult questions. The number of optimal distractors for MCQs remains debatable [13]. We choose to randomly select 3 distractors for each question. A description of each category of questions and the number of generated questions for each category is presented in Table 1.

It makes sense to have only easy questions in two of the above categories. This is because these questions where designed such that there is a concept $S1$ such that the key is not subsumed by $S1$ but the distractors are. And since similarity depends on the number of subsumers, similarity between the key and distractors should be low, especially when $S1$ is an atomic subsumer. Hence the generated distractors only fit the criteria for easy question generation.

Specifying easy vs. difficult questions using similarity measures $SubSim(\cdot)$ [5] has been used to generate most of the questions described above with the exception of using $GrammarSim(\cdot)$ [5] to generate What is X? (with class expressions as answers). This is justified by the fact that, when the answers are expressed in detail (e.g., class expressions rather than simply class names), the similarity measure should be more precise. It remains to specify the upper and lower bounds of similarity values which can be used generate appropriate distractors for easy and difficult questions. Rather than specifying a random number, we choose to calculate the average similarity values between all siblings in the ontology. This average similarity value is then used as the lower bound for generating a difficult distractor, where 1 is the upper bound. The lower bound to generate an easy distractor is set to be two thirds of the lower bound of difficult distractors.

Question review Three reviewers involved in leading the course have been asked to evaluate the 50 randomly selected questions using a web interface. For each question, the reviewer first attempts to solve the questions and then specifies whether he/she thinks that the question is (0) not useful at all, (1) useful as a seed for another question, (2) useful but requires major improvements, (3) useful but requires minor improvements or (4) useful as it is. Then, the reviewer

[3] A web-based interface for the QG tool can be accessed at http://edutechdeveloper.com/MCQGen/

predicts how certain third year students participating in the current experiment would perform on the question. To distinguish between acceptable and extreme levels of difficulty, we ask the reviewers to choose one of the following options for each questions: (1) too easy, (2) reasonably easy, (3) reasonably difficult and (4) too difficult. In what follows, we number the reviewers based on their job completion time. Hence, first reviewer refers to the reviewer who first finished the reviewing process.

Question administration A sample of the questions which have been rated by the reviewers as useful (or useful with minor improvements) by at least 2 reviewers have been administered to third year students[4] who are enrolled in the course unit for the academic year 2013/14 and who were about to sit the final exam. Two sets of the questions have been administered in two different rounds to increase participation rate. In the first round, a total of 6 questions (3 easy, 3 difficult) have been administered to 19 students using paper-and-pencils during a revision session at the end of the course. The students had 10 minutes to answer the 6 questions. In the second round, another set of 6 questions (3 easy, 3 difficult) have been administered to 7 students via BlackBoard one week before the final exam and the students were allowed to answer the questions at any time during this week.

Statistical analysis Item response theory (IRT) [24] has been used for the statistical analysis of students results. IRT studies the statistical behaviour of good/bad questions. In particular, it studies the following properties: (i) item difficulty, (ii) discrimination between good and poor students and (iii) guessability.

6.2 Analogous/prior experiments

Mitkov et. al [25] have developed an approach to automatically generate MCQs using NLP methods. They also utilise WordNet [23] to generate distractors that are similar to the key. They do not explicitly link item difficulty to similarity patterns between keys and distractors. However, they report that average item difficulty for the generated questions was above 0.5 (i.e., considered difficult) which can be explained by our similarity theory [4] since they choose to generate distractors that are similar to the key.

In an earlier study [3], we have evaluated a large set of multiple-choice analogy questions[5] which have been generated from three different ontologies. The evaluation was carried out using an automated solver which simulates a student trying to answer these questions. The use of the automated solver facilitated the evaluation of the large number of questions. The current experiment in which we recruit a group of students in real class settings confirms the results of study carried earlier using the automated solver.

[4] This study has been approved by the ethics committee in the School of Computer Science, The University of Manchester (approval number: CS125).

[5] In an analogy question, a pair of concepts of the form "A is to B" is presented to the student who is asked to identify the most similar pair of concept out of a set of pairs provided as alternative answers to the question.

6.3 Results and discussion

Overall cost. As mentioned earlier, the cost of QG might, in some cases, include any costs of developing a new ontology or reviewing/editing an existing one. For the current experiment, we experienced the extreme case of having to build an ontology from scratch for the purpose of QG. A total of 14 hours were spent to develop the ontology described above. For computation time, we need to consider both the time required to compute pairwise similarity for the underlying ontology and the time required to compute the questions. Computing pairwise similarity for all sub-concept expressions (including concept names) in the KA ontology took 22 minutes. This includes time required to compute similarities using both $SubSim(\cdot)$ and $GrammarSim(\cdot)$ [5] for a total of 296 sub-concept expressions. Computing a total of 913 questions took around 21 minutes. Computing "which is the odd one out?" questions took 17 minutes out of the 19 minutes while computing all other questions took less than 4 minutes.

Finally, we also have to consider any time required to review the questions (possibly including post-editing time). As the reviewers were allowed to review each item in an unrestricted manner, it is difficult to determine the exact time that each reviewer has spent on each item. For example, for a set of questions, a reviewer might start looking at a question on a given day and then submits the review on the next day after getting interrupted for any reason. We exclude questions for which the recorded time was more than 60 minutes as this clearly shows that the reviewer was interrupted in the middle of the reviewing process. The first reviewer spent between 13 and 837 seconds to review each of the 50 questions, including time for providing suggestions to improve the questions. The second reviewer spent between 12 and 367 seconds. And the third reviewer spent between 17 and 917 seconds. Note that these times include the time required to attempt to answer the question by the reviewer.

Usefulness of questions. A question is considered "useful" if it is rated as either "useful as it is" or "useful but requires minor improvements" by a reviewer. 46 out of 50 questions were considered useful by at least 1 reviewer. 17 out of the 46 questions were considered useful by at least 2 reviewers. The first reviewer rated 37 questions as being useful while the second and third reviewer rated 8 and 33 questions as useful respectively. Note that the third reviewer is the main instructor of the course unit during the academic year in which the experiment has been conducted while the second reviewer taught the course unit in the previous year. The first reviewer has not taught this course unit before but has general knowledge of the content.

Usefulness of distractors. A given distractor is considered "useful" if it has been functional (i.e., picked by at least one student). For the 6 questions which were administered on paper, at least 2 out of 3 distractors were useful. In 5 out of the 6 questions, the key answer was picked more frequently than the distractors. Exceptionally, in 1 question, a particular *linguistically unique* distractor was picked more frequently than the key. The course instructor justified this by pointing out that this question was not covered explicitly in class. For the 6 questions which have been administered on BlackBoard, at least one distractor was useful except for one question which has been answered correctly by all 7 students.

Item discrimination. We used Pearson's coefficient to compute item discrimination to show the correlation between students' performance on a given

questions and the overall performance of each student on all questions. The range of item discrimination is [-1,+1]. A good discrimination value is greater than 0.4 [12]. For the 6 questions administered on paper and 4 out of 6 questions administered via BlackBoard, item discrimination was greater than 0.4. For one question administered via BlackBoard, item discrimination could not be calculated as 100% of students answered that question correctly. Finally, item discrimination was poor for only one question. The third reviewer pointed out that this question is highly guessable.

Item difficulty. One of the core functionalities of the presented QG tool is to be able to control item difficulty. To evaluate this functionality, we examine tool-reviewers agreement and tool-students agreement. As described above, the tool generates questions and labels them as easy or as difficult. Each reviewer can estimate the difficulty of a question by choosing one of the following options: (1) too easy, (2) reasonably easy, (3) reasonably difficult and (4) too difficult. A question is too difficult for a particular group of students if it is answered correctly by less than 30% of the students and is too easy if answered by more than 90% of the students [11]. In both cases, the question needs to be reviewed and improved. Accordingly, we consider a question to be difficult if answered correctly by 30-60% and easy if answered correctly by 60-90% of the students.

Before discussing tool-reviewers agreement, it is worth to note agreements among reviewers. We distinguish between loose agreements and strict agreements. A loose agreement occurs when two reviewers agree that a question is easy/difficult but disagree whether it is too easy/difficult or reasonably easy/difficult. Table 2 summarises agreements among reviewers. Each reviewer agrees with the tool on 31 (not necessarily the same) questions.

With regard to the 6 questions delivered on paper, 2 questions were reasonably difficult and 2 were reasonably easy for the students. These 4 questions were in line with difficulty estimations by the QG tool. 1 out of the 6 questions was too difficult for the students. Most of the students picked a linguistically unique distractor rather than the key. Remarkably, the tool and the three reviewers have rated this item as easy. Finally, 1 question was too easy for the students however it was rated as difficult by the tool. This is due to having a clue in the stem. Similarly, for BlackBoard questions, 1 question was reasonably difficult and 1 question was reasonably easy for the students; just in line with tool estimations. 1 out of the 6 questions was too easy for the students (100% correct answers). This question was rated as easy by the tool. Again, 1 question was rated as difficult by the tool but was easy for the students due to having a clue in the stem. 2 questions were not in line with tool estimations but were in line with at least 2 reviewers estimations.

7 Conclusion and future research directions

With the growing interest in ontology-based QG approaches, there is a clear need of putting these approaches into practise. We have examined the pedagogical characteristics of questions generated from a handcrafted ontology. The results are promising with regard to both usefulness of questions and difficulty prediction. For future work, we will experiment with more ontologies, in particular, existing ontologies rather than handcrafted ones. Also, it remains to incorporate NLP methods to account for linguistic/lexical similarities..

Question	description	Total	Total-Easy	Total-difficult	3+dist	3+dist-easy	3+dist-difficult
What is the following definition describing?	Stem: string + Annotation, answers: class names	27	17	10	21	16	5
Which is the odd one out?	Stem: string, answers: class names (key is not subsumed by S1 while distractors are subsumed by S1, S1 is an atomic)	94	94	0	75	75	0
What is X?	X: atomic, answers: class names (key: subsumers of X, distractors: non-subsumers of X excluding subsumers of the key)	133	77	56	101	71	30
What is X?	X: class name, answers: class expressions (key: subsumers of X, distractors: non-subsumers of X excluding subsumers of the key to avoid generating very similar but trivial distractors)	259	162	97	133	106	27
Which of these is X?	X: class names, answers: class names (key: subsumees of X, distractors: non-subsumees of X excluding subsumers and siblings of the stem)	55	54	1	41	41	0
Which of these is X?	X: class expression, answers: class names (key: subsumees of X, distractors: non-subsumees of X excluding subsumers of the stem)	345	229	116	185	165	20

Table 1: The number of generated questions from KA ontology according to 6 different categories

	1st & 2nd	1st & 3rd	2nd & 3rd
Loose agreements	31	26	33
Strict agreements	19	15	15

Table 2: Loose and strict agreements between the three reviewers

References

1. Achieve. Testing: Setting the record straight. Technical report, Washington, DC: Author, 2000.
2. T. Alsubait, B. Parsia, and U. Sattler. Mining ontologies for analogy questions: A similarity-based approach. In *OWLED*, 2012.
3. T. Alsubait, B. Parsia, and U. Sattler. Next generation of e-assessment: automatic generation of questions. *International Journal of Technology Enhanced Learning*, 4(3/4):156–171, 2012.
4. T. Alsubait, B. Parsia, and U. Sattler. A similarity-based theory of controlling mcq difficulty. In *Second International Conference on e-Learning and e-Technologies in Education (ICEEE)*, pages 283–288, 2013.
5. T. Alsubait, B. Parsia, and U. Sattler. Measuring similarity in ontologies: How bad is a cheap measure? In *27th Inernational Workshop on Description Logics (DL-2014)*, 2014.
6. I. Bejar. Subject matter experts' assessment of item statistics. *Applied Psychological Measurement*, 7:303–310, 1983.
7. B. S. Bloom and D. R. Krathwohl. *Taxonomy of educational objectives: The classification of educational goals by a committee of college and university examiners. Handbook 1. Cognitive domain*. New York: Addison-Wesley, 1956.
8. J. Brown, G. Firshkoff, and M. Eskenazi. Automatic question generation for vocabulary assessment. In *Proceedings of HLT/EMNLP*, pages 819–826, Vancuver, Canada, 2005.
9. M. Cubric and M. Tosic. Towards automatic generation of e-assessment using semantic web technologies. In *Proceedings of the 2010 International Computer Assisted Assessment Conference*, University of Southampton, July 2010.
10. C. d'Amato, N. Fanizzi, and F. Esposito. A dissimilarity measure for ALC concept descriptions. *Proceedings of the 21st Annual ACM Symposium of Applied Computing, SAC2006, Dijon, France. ACM.*, 2:16951699, 2006.
11. B. G. Davis. *Tools for Teaching*. San Francisco, CA: Jossey-Bass, 2001.
12. R.L. Ebel. Procedures for the analysis of classroom tests. *Educational and Psychological Measurement*, 14:352–364, 1954.
13. T.M. Haladyna and S.M. Downing. How many options is enough for a multiple choice test item? *Educational & Psychological Measurement*, 53(4):999–1010, 1993.
14. M. Heilman. *Automatic Factual Question Generation from Text*. PhD thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2011.
15. E. Holohan, M. Melia, D. McMullen, and C. Pahl. The generation of e-learning exercise problems from subject ontologies. In *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*, pages 967–969, 2006.
16. M. Horridge and S. Bechhofer. The OWL API: A Java API for working with OWL 2 ontologies. In *In Proceedings of the 6th International Workshop on OWL: Experiences and Directions (OWLED)*, 2009.
17. M. Horridge, B. Parsia, and U. Sattler. The state of biomedical ontologies. In *BioOntologies 2011 15th-16th July, Vienna Austria*, 2011.
18. A. Hoshino and H. Nakagawa. Real-time multiple choice question generation for language testing: a preliminary study. In *Proceedings of the Second Workshop on Building Educational Applications using Natural Language Processing*, pages 17–20, Ann Arbor, US, 2005.
19. A. Hoshino and H. Nakagawa. Predicting the difculty of multiple-choice cloze questions for computer-adaptive testing. In *Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics*, Romania, March 21-27 2010.

20. J. Impara and B. Plake. Teachers' ability to estimate item difficulty: A test of the assumptions in the angoff standard setting method. *Journal of Educational Measurement*, 35(1):69–81, 1998.
21. K. Lehmann and A. Turhan. A framework for semantic-based similarity measures for ELH-concepts. *JELIA 2012*, pages 307–319, 2012.
22. C. Liu, C. Wang, Z. Gao, and S. Huang. Applications of lexical information for algorithmically composing multiple-choice cloze items. In *Proceedings of the Second Workshop on Building Educational Applications using Natural Language Processing*, pages 1–8, Ann Arbor, US, 2005.
23. G. MILLER. WordNet: A lexical database for English. *COMMUNICATIONS OF THE ACM*, 38(11), November 1995.
24. M. Miller, R. Linn, and N. Gronlund. *Measurement and Assessment in Teaching, Tenth Edition*. Pearson, 2008.
25. R. Mitkov, L. An Ha, and N. Karamani. A computer-aided environment for generating multiple-choice test items.cambridge university press. *Natural Language Engineering*, 12(2):177–194, 2006.
26. A. Papasalouros, K. Kotis, and K. Kanaris. Automatic generation of multiple-choice questions from domain ontologies. In *IADIS e-Learning 2008 conference*, Amsterdam, 2008.
27. M. Paxton. A linguistic perspective on multiple choice questioning. *Assessment & Evaluation in Higher Education*, 25(2):109–119, 2001.
28. R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. In *IEEE Transaction on Systems, Man, and Cybernetics*, volume 19, page 1730, 1989.
29. P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th international joint conference on Artificial intelligence (IJCAI95)*, volume 1, pages 448–453, 1995.
30. V. Rus and A. Graesser. Workshop report: The question generation task and evaluation challenge. Institute for Intelligent Systems, Memphis, TN, ISBN: 978-0-615-27428-7, 2009.
31. J. T. Sidick, G. V. Barrett, and D. Doverspike. Three-alternative multiple-choice tests: An attractive option. *Personnel Psychology*, 47:829–835, 1994.
32. M. Simon, K. Ercikan, and M. Rousseau, editors. *Improving Large Scale Education Assessment: Theory, Issues, and Practice*. Routledge, New York, 2013.
33. D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR)*, 2006.
34. Z. Wu and MS. Palmer. Verb semantics and lexical selection. In *Proceedings of the 32nd. Annual Meeting of the Association for Computational Linguistics (ACL 1994)*, page 133138, 1994.
35. B. Zitko, S. Stankov, M. Rosic, and A. Grubisic. Dynamic test generation over ontology-based knowledge representation in authoring shell. *Expert Systems with Applications: An International Journal*, 36(4):8185–8196, 2008.
36. K. Zoumpatianos, A. Papasalouros, and K. Kotis. Automated transformation of SWRL rules into multiple-choice questions. In *FLAIRS Conference'11*, 2011.