An Experimental Evaluation of Automatically Generated Multiple Choice Questions from Ontologies

Ghader Kurdi, Bijan Parsia, and Uli Sattler

School of Computer Science, The University of Manchester,
Kilburn Building, Oxford Road, Manchester, M13 9PL, United Kingdom

Abstract. In order to provide support for the construction of MCQs, there have been recent efforts to generate MCQs with controlled difficulty from OWL ontologies. Preliminary evaluation suggests that automatically generated questions are not field ready yet and highlight the need for further evaluations. In this study, we have presented an extensive evaluation of automatically generated MCQs. We found that even questions that adhere to guidelines are subject to the clustering of distractors. Hence, the clustering of distractors must be realised as this could affect the prediction of difficulty.

1 Introduction

Multiple Choice Questions (MCQs) are a widely adopted form of question in both paper- and electronic-based tests. A great proportion of large scale tests consist of MCQs. They have gained further importance with the advent of e-learning and Massive Open Online Courses (MOOCS) (e.g. Coursera, Future Learn, and Udacity), in which providing assessment and feedback on a large scale is challenging. However, MCQs are labour intensive, time consuming and difficult to construct. Well-constructed MCQs require a considerable time for design, writing, and revision. In order to provide support for the construction of MCQs, there have been recent efforts to generate MCQs with controlled difficulty from OWL ontologies based on the similarity theory of difficulty [1]. The similarity theory associates difficulty with the degree of similarity between the key (correct option) and the distractors (incorrect options). Despite the advances in the method, preliminary evaluation suggests that generated questions are not field ready yet and highlight the need for more extensive evaluation of the questions.

The objective of this study is to evaluate the quality of, and to categorise various problematic phenomena of, automatically generated MCQs from ontologies based on the aforementioned theory [1]. Another objective of this study is to distinguish issues that are intrinsic to similarity theory from natural language and presentation issues. The specific questions driving this study are:

1. What are the issues presented in automatically generated question? And to what degree are they prevalent?
2. Are these issues intrinsic properties of the similarity theory as opposed to natural language and presentation issues?

The main contribution of this study is the identification of a new problematic phenomenon of clustered distractors that influences the prediction of difficulty.

2 Materials and Methods

Experimental Data Our study used two domain ontologies for the evaluation: the Knowledge Acquisition (KA) ontology and the Java ontology. The KA and Java ontologies were handcrafted with the purpose of question generation in mind.[1] The reason behind choosing these two ontologies is the availability of corresponding courses provided by the School of Computer Science at the University of Manchester. The ontological statistics are provided in Table 1.

Experimental Set-up The following machine has been used to carry out the experiment presented in this study: Intel core i7 2.4GHz processor, 8 GB RAM, running Windows OS 8.1 (HP Spectre 2015 model).

Table 1. Statistics for the experimental ontologies.

Ontology	Classes	Properties	Individuals	Logical axiom
KA	151	7	0	254
Java	305	74	0	554

2.1 MCQ Generation

We used the MCQ generator developed by Alsubait et al. [1] to automatically generate MCQs using the aforementioned ontologies as inputs. The tool generates six types of questions that are explained in Appendix A. The generated questions are classified by the tool into 'easy' or 'difficult' questions. Each question consists of a stem (a text that poses the question), a key, and a non-empty set of distractors minimally containing two distractors. Different versions can be constructed from the suggested questions by selecting different subsets of the distractors. The number of generated questions is provided in Table 2. Generating questions from the Java ontology took 12 days while generating questions from the KA ontology took around 12 hours. It is clear from the table that the number of difficult questions (67 difficult questions) is low compared to the number of easy questions (2090 easy questions). The reason is that few distractors with a very high similarity to the key can be found in ontologies [1].

Although the size of the Java ontology is about double the size of the KA ontology, the number of easy questions generated from the Java ontology is about 11 times larger than the number of questions generated from the KA ontology. In addition, generating questions from the Java ontology took much more time than generating questions from the KA ontology. We expect that the magnitude

[1] For a detailed description of both ontologies, the reader is referred to [1] and [2].

of the difference between the number of questions and the generation cost in terms of time is related to the depth of the inferred class hierarchy. Looking at both ontologies, we noticed that the class hierarchy of the Java ontology is divided into eleven levels compared to five levels in the KA. In addition, many classes in the Java ontology have multiple direct subsumers while classes in KA have only one direct subsumer. To illustrate the effect of this, let us consider two classes: class (A) which is located at level 11 and has two direct subsumers throughout the hierarchy, and class (B) which is located at level 5 and has a single subsumer at each level. Taking the question category "What is X" as an example, the number of generated questions for class (A) is expected to be about $2^n - 2 = 2^{11} - 2 = 2046$ questions where n represents the number of levels. However, the number of questions for class (B) is only 30. Note that to generate the questions, similar distractors for each class must be found first. This non-linear growth suggests ontologies as a supplier that can satisfy the demand for a large number of questions since adding a few classes and submission relations increases the number of generated question significantly.

2.2 Sample Selection

Due to the large number of easy questions generated, we used a stratified sampling method in which questions were divided into groups according to the question category. With regards to easy questions, we randomly selected the questions from the different groups in proportion to their number, taking into account a 95% confidence level and 5% margin of error. As the number of difficult questions was small, we evaluated them all. The total number of evaluated questions is 506 questions (67 difficult questions and 439 easy questions), as shown in Table 2.

Table 2. Statistics for the number of generated questions. Note that the sizes of the samples of easy questions are represented between parentheses.

	Java			**KA**		
Question category	**Easy**		**Difficult**	**Easy**		**Difficult**
Generalisation: What is X	393	(66)	6	11	(8)	0
Generalisation 2: What is X2	0		0	56	(39)	8
Specification: Which is X	260	(43)	22	15	(11)	0
Specification 2: Which is X2	88	(15)	11	82	(58)	0
Definition: Which term	207	(35)	20	2	(1)	0
Recognition: Which is odd	976	(163)	0	0		0
Total	1924	(322)	59	166	(117)	8

2.3 Evaluation Criteria

We performed a preliminary evaluation of automatically generated questions and observed some problematic questions. We then referred back to the literature that discussed and suggested guidelines for developing MCQs [[3], [4]]. Haladyna

et al. [3] conducted a review of MCQ writing guidelines for assessment. In addition, Pho et al. [4] performed an analysis of multiple choice question corpus in order to define distractor characterisation. The initial criteria for our evaluation started with suggestions described in the aforementioned studies. Table 3 gives an overview of the initial set of criteria. A detailed discussion of each criterion will be provided in the associated result section for clarity. Examples of generated questions that do not adhere to guidelines can be found in Appendix B. Then, through an iterative process of evaluating the questions, we developed a new criterion for selecting distractors that was not mentioned in the literature, as will be discussed in Section 3.5.

Table 3. The predefined criteria for assessing automatically generated questions (adapted from [3], and [4]).

Quality criterion
(Q1) The question is grammatically correct.
(Q2) The question contains no clues to the key.
(Q3) Options are homogeneous in grammatical structure.
(Q4) Options are homogeneous in content.

3 MCQ Evaluation: Results and Discussion

3.1 Grammatical Correctness

The grammatical correctness of questions is an important consideration when constructing MCQs since grammatical inconsistency could give test takers without sufficient knowledge a clue to the correct answer. In order to investigate the grammatical correctness of automatically generated MCQs, we classified questions based on the level of the grammatical corrections required into:

(MIN) minor correction: involves adding appropriate articles, fixing any subject-verb disagreement and tokenising the stem and the options, including segmentation, as well as processing of camel case and underscores;
(MED) medium correction: involves inserting or deleting up to three words from the stem and the options;
(MAJ) major correction: involves rephrasing of the stem or the options.

The distribution of questions according to the level of the grammatical corrections required is shown in Table 4. Although the majority of MCQs required only minor corrections, there is a considerable number of questions requiring major corrections. Presenting questions in OWL syntax is the main reason behind the need for major grammatical corrections. However, this issue is repairable by employing one of the available ontology verbalisers. Evaluating different verbalisers in order to choose the most suitable for the purpose of question verbalisation is a part of future work. In addition, the issues of segmentation and processing of camel-case and underscore can be achieved by employing regular expressions.

The total number of questions requiring major correction was higher in the KA ontology because a higher number of questions containing sub-expressions was generated from the KA ontology (Table 11).

Table 4. Results for question evaluation in regards to the required level of grammatical corrections.

	Easy			**Difficult**		
Question category	**Minor**	**Medium**	**Major**	**Minor**	**Medium**	**Major**
What is X	70	4	0	6	0	0
What is X2	0	0	39	0	0	8
Which is X	54	0	0	22	0	0
Which is X2	0	0	73	0	0	11
Which term	36	0	0	20	0	0
Which is odd	159	0	4	-	-	-
Total	319	4	116	48	0	19

3.2 Syntactic Clues

One of the MCQ writing guidelines in regards to writing the choices is to avoid "choices identical to or resembling words in the stem" [3]. Alsubait et al. [1] identified word clues as a problem that affects the accuracy of the difficulty prediction. We have considered different possible similarities in wording between the stem and the options:

(SK) shared word(s) or phrase between the stem and the key;
(SD) shared word(s) or phrase between the stem and one or more distractors;
(SKD) shared word(s) or phrase between the stem and the options including the key and one or more distractors.
(ANT) a word in the stem has an antonym in one or more of the distractors.

The form (SK) should be avoided because it makes the key stand out as the correct answer. On the other hand, if word(s) or a phrase in the stem are repeated in the distractor(s) only, this make the distractor(s) more attractive to low information students. This form (SD) can be desirable because it improves the functionality of the clued distractor(s) and possibly the discrimination of the item. However, the attractiveness of the clued distractors tends to decrease the functionality of the other distractors. Finally, regarding the third form (SKD), there is a preference over other options for options that share similar wording with the stem, as mentioned earlier. This leads to the nonfunctionality of some of the distractors and increases the guessability of the item. However, we did not consider questions where all distractors share word(s) with the key and the stem as containing a syntactic clue. We identified another form of syntactic clue in which a word in the stem has an antonym in one or more of the distractors. This form also needs to be avoided because the distractor(s) are clued as the wrong answer(s). A lexical database such as WordNet can be used to acquire the

antonyms of concepts in the stem. The acquired terms can be associated with the stem and taken into account during the question generation.

Table 5 shows the distribution of the evaluated questions in regards to the aforementioned forms. Table 11 shows the proportion of questions that contain syntactic clues to the total number of questions in each ontology. The evaluation indicated that 25.4% and 12.5% of difficult questions generated from the Java and the KA ontologies respectively contain clues to the keys which, in turn, make the questions easy. One of the suggested solutions is to provide alternative names using OWL annotation properties which can be used by the tool if wording similarity between the stem and key is detected.

Table 5. Results for question evaluation in regards to syntactic clues.

	Easy					**Difficult**				
Question category	**SK**	**SD**	**SKD**	**ANT**	**No clue**	**SK**	**SD**	**SKD**	**ANT**	**No clue**
What is X	4	27	6	0	37	1	4	1	0	0
What is X2	13	2	5	0	19	1	0	0	0	7
Which is X	15	12	6	5	19	8	1	1	0	12
Which is X2	13	8	8	0	44	2	0	0	0	9
Which term	1	13	16	2	6	4	7	7	0	2
Which is odd	0	0	0	0	163	-	-	-	-	-
Total	42	62	48	7	274	16	12	9	0	30

3.3 Syntactic Consistency

One of the recommendations from the literature regarding the syntactic structure of the options is to "keep choices homogeneous in content and grammatical structure" [3]. Another related recommendation is to avoid "grammatical inconsistencies that cue the test-taker to the correct choice" [3]. In order to investigate to what extent automatically generated questions follow these rules, we automatically annotated the distractors with syntactic information about parts of speech (i.e. nouns (NN), verbs (VB), determiner (DT), etc.) using the Stanford part-of-speech tagger[2]. We then manually applied corrections where needed to the assigned part of speech for each distractor. We compared the key and each distractor in terms of their syntactic structures independently of their meaning as suggested in [4]. We consider the distractor and the key to be:

(GC) grammatically consistent: if their assigned parts of speech are identical,
(PC) partially consistent: if they share some parts of speech,
(IC) grammatically inconsistent: if their assigned parts of speech are totally different.

Looking at different generated questions where syntactic inconsistency presents, we concluded that grammatical inconsistency can highlight the need for modification of either the questions, or the names used in the ontology, even though this is not always associated with invalid distractors.

[2] Downloaded from: http://nlp.stanford.edu/software/tagger.shtml.

The number of questions that contain syntactic inconsistency and a detailed analysis of the number of syntactically consistent and inconsistent distractors is presented in Table 6. Table 11 show the percentage of distractors that are syntactically inconsistent with the key in each ontology. The proportion of distractors distributed over the three categories seems to be consistent in the two ontologies.

Table 6. Results of evaluating syntactic consistency. Note that the upper part reports the number of questions while the lower part reports the number of distractors.

	Easy			**Difficult**		
Question category	**GC and PC**		**IC**	**GC and PC**		**IC**
What is X	24		50	6		0
What is X2	39		0	7		0
Which is X	34		20	22		0
Which is X2	73		0	11		0
Which term	18		18	17		3
Which is odd	151		12	-		-
Total	339		100	63		3
	GC	**PC**	**IC**	**GC**	**PC**	**IC**
What is X	556	3,984	374	0	38	0
What is X2	45	74	0	12	39	0
Which is X	259	801	221	23	88	0
Which is X2	69	280	0	11	63	0
Which term	281	765	81	39	86	4
Which is odd	138	452	61	-	-	-
Total	1,348	6,356	737	85	314	4

3.4 Semantic Homogeneity

The guidelines suggest maintaining the homogeneity of options in MCQs (Q4 in Table 3). Pho et al. [4] define semantically homogeneous distractors as the alternatives that "share a common semantic type (expected by the question)". We observed that there are some questions for which the semantic type is deducible from the stem which, in turn, enforces the use of semantically homogeneous options. Otherwise, distractors are ruled out because of type mismatch between the distractors and the key. Based on this, we consider a distractor to be either:

(HOMO) homogeneous: if its type is compatible with the expected type of the key,
(HETERO) heterogeneous: if its type is not compatible with the expected type of the key.

We conducted an analysis by checking whether the expected answer type is suggested in the question either explicitly or implicitly. Then, we checked the compatibility of distractors with the expected answer type. Table 7 shows the results of investigating the compatibility of automatically generated MCQs with the semantic homogeneity rule (Q4). Table 11 shows the distribution of questions per ontology according to semantic homogeneity.

Table 7. Results for question evaluation in regards to semantic homogeneity.

	Easy			Difficult		
Category	**Homo**	**Hetero**	**Not applicable**	**Homo**	**Hetero**	**Not applicable**
What is X	3	0	71	0	0	6
What is X2	0	0	39	0	0	8
Which is X	12	12	30	5	0	17
Which is X2	0	0	73	0	0	11
Which term	17	18	1	14	6	0
Which is odd	0	0	163	-	-	-
Total	32	30	377	19	6	42

3.5 Clustered Distractors

All aforementioned flaws are regarded as linguistic or presentation issues that can be repaired by incorporating existing natural language processing and generation techniques. However, we observed an interesting phenomenon of the existence of interrelations between distractors in automatically generated questions. We called this phenomenon "clustered distractors". The following examples illustrate this phenomenon. The first two examples represent different versions of the same question where the difference is in the distractor sets. In the first version, distractors (A) and (B) are clustered because they both represent relational operators. A test taker who knows that relational operators are binary operators will easily eliminate the distractors and arrive at the correct answer. Hence, the item functions as a true-false question. Recognising one as a binary operator and the relation between the distractors gives a clue to the answer. However, in the second version, a test taker must consider each distractor and recognise it as a binary operator in order to arrive at the correct solution.

Stem: Which of the following is [a] Unary Operator

A. Less than or equal
B. Less than
C. Logical complement operator ▲ **Key**

A. Less than or equal
B. Logical OR
C. Logical complement operator ▲ **Key**

Another form of clustered distractors is presented in the following example generated from the Java ontology. Recognising that a primitive type and a scalar represent the same concept clue the test taker to select array because (A) and (B) cannot both be correct as MCQs require only one correct option. More examples are presented in the appendix.

Stem: Which of the following is [a] Reference Type?

A. Primitive type
B. Scalar
C. Array ◀ **Key**

We define clustered distractors as a subset of distractors with very high similarity among them. Our assumption is that clustered distractors make questions

easier than expected. That is, even if the question is predicted to be difficult because of the high similarity between the key and the distractors, the high similarity between the distractors draws a boundary between the key and the cluster of distractors. However, the similarity theory is blind to this fact since only the similarity between the key and distractors is considered.

The results of analysis for clustered distractors are presented in Table 8 and Table 11. The evaluation indicated that the phenomenon is dominant. A considerable number of questions in both ontologies contain clustering of distractors, with the Java ontology having a higher percentage (94.7% of easy questions and 88.1% of difficult questions). All questions in the question category "Which is odd" contain clustered distractors, which is the nature of this category of questions. One of the patterns that we noticed with regards to clustered distractors is that they represent siblings in the ontology. This is not surprising as it is expected that, in ontologies, siblings are usually very similar to each other.

Table 8. Statistics for the number of questions containing clustered distractors.

	Easy		**Difficult**	
Question category	**Clustered**	**Not clustered**	**Clustered**	**Not clustered**
What is X	71	3	6	0
What is X2	14	25	0	8
Which is X	52	2	22	0
Which is X2	43	30	11	0
Which term	26	10	13	7
Which is odd	163	0	-	-
Total	369	70	52	15

3.6 Level of Repairs

The final phase of the evaluation was to investigate the relationship between the flaws in the questions and the effort required to repair the questions. We classified questions in terms of the level of repairs required into:

- minor repair: involves minor grammatical corrections and selecting distractors if enough distractors are provided by the tool;
- medium repair: involves medium grammatical corrections and writing one distractor in order to have a question with one key and 2 distractors (3 options MCQs) if not enough distractors are provided;
- major repair: involves major grammatical correction and writing two or more distractors in order to have a question with one key and 2 distractors (3 options MCQs) if not enough distractors are provided.

The results are summarised in the following tables. It is not surprising that few questions are flawless given the fact that no natural language generation techniques were incorporated into the tool. Filtering flawed questions will result in an insufficient number of questions. Although the majority of questions contain more than one flaw, most of them are repairable by applying minor repairs. This is because a large number of distractors per question is suggested.

Table 9. Statistics for the number of flawed questions and the level of repair required.

Category	Easy				Difficult			
	Flawless	1 Flaw	≥ 2 Flaws		Flawless	1 Flaw	≥ 2 Flaws	
What is X	0	5	69		0	0	6	
What is X2	0	14	25		0	7	1	
Which is X	0	0	54		0	0	22	
Which is X2	0	13	60		0	0	11	
Which term	5	0	31		0	3	17	
Which is odd	20	130	13		-	-	-	
Total	25	162	252		0	10	57	
	None	**MIN**	**MED**	**MAJ**	**None**	**MIN**	**MED**	**MAJ**
What is X	0	63	6	5	0	4	1	1
What is X2	0	20	6	13	0	7	0	1
Which is X	0	26	5	23	0	11	4	7
Which is X2	0	53	9	11	0	9	0	2
Which term	0	23	11	2	0	14	5	1
Which is odd	25	138	0	0	-	-	-	-
Total	25	323	37	54	0	45	10	12

Table 10. The proportion of flawed questions per ontology.

Difficulty	Category	Java		KA	
		Number	Percentage	Number	Percentage
Easy	Flawless	25	7.76%	0	0
	1 Flaw	136	42.24%	26	22.22%
	≥ 2 Flaws	161	50%	91	77.78%
Difficult	Flawless	0	0	0	0
	1 Flaw	3	5.09%	7	87.50%
	≥ 2 Flaws	56	94.92%	1	12.50%
The level of repair required					
Easy	Not required	25	7.76%	0	0
	Minor	259	80.44%	64	54.70%
	Medium	19	5.90%	18	15.39%
	Major	19	5.90%	35	29.92%
Difficult	Not required	0	0	0	0
	Minor	38	64.41%	7	87.50%
	Medium	10	16.95%	0	0
	Major	11	18.64%	1	12.50%

Table 11. The proportion of questions per ontology distributed according to: A) grammatical corrections, B) syntactic clues, C) syntactic consistency, D) semantic homogeneity, and E) clustered distractors

		Java		KA	
Difficulty	**Category**	**Number**	**Percentage**	**Number**	**Percentage**
		A) Grammatical corrections			
Easy	Minor	299	92.86%	20	17.09%
	Medium	4	1.24%	0	0
	Major	19	5.90%	97	82.91%
Difficult	Minor	48	81.36%	0	0
	Medium	0	0	0	0
	Major	11	18.64%	8	100%
		B) Syntactic clues			
Easy	SK	20	6.2%	22	18.80%
	SD	50	15.5%	12	10.26%
	SKD	28	8.7%	20	17.09%
	ANT	7	2.2%	0	0
	No clue	222	68.9%	52	44.44%
Difficult	SK	15	25.4 %	1	12.5%
	SD	12	20.3%	0	0
	SKD	9	15.3%	0	0
	ANT	0	0%	0	0
	No clue	23	39%	7	87.5%
		C) Syntactic consistency (no. of questions)			
Easy	GC and PC	231	71.74%	108	92.31%
	IC	91	28.26%	9	7.69%
Difficult	GC and PC	56	94.92%	7	100%
	IC	3	5.09%	0	0
		C) Syntactic consistency (no. of distractors)			
Easy	GC	1,258	15.7%	91	17.95%
	PC	6,028	75.6%	369	72.78%
	IC	690	8.6%	47	9.27%
	Total	7,976	100%	507	100%
Difficult	GC	73	20.74%	3	9.68%
	PC	275	78.13%	28	90.32%
	IC	4	1.14%	0	0
	Total	352	100%	31	100%
		D) Semantic homogeneity			
Easy	Homogeneous	23	7.14%	9	7.69%
	Heterogeneous	30	9.33%	0	0
	Not applicable	269	83.54%	108	92.31%
Difficult	Homogeneous	19	32.20%	0	0
	Heterogeneous	6	10.17%	0	0
	Not applicable	34	57.63%	8	100%
		E) Clustered distractors			
Easy	Clustered distractors	305	94.72%	64	54.70%
	Not clustered distractors	17	5.28%	53	45.30%
Difficult	Clustered distractors	52	88.14%	0	0
	Not clustered distractors	7	11.86%	8	100%

4 Conclusion

In this study, we have presented an evaluation of automatically generated MCQs. The objective was to validate the quality of the questions and thus later be able to improve the automatic question generation process. The study confirms the need to present questions more naturally. Syntactic, and syntax-based similarity as well as semantic similarity between options must be taken into consideration when automatically selecting distractors from ontologies. Available natural language processing and generation techniques, as well as some ontology modeling guidelines, suffice to overcome the linguistic issues. Alternatively, an automatic checker would be highly valuable in highlighting problematic questions and minimising review time. We also found that even questions that adhere to guidelines are subject to the clustering of distractors. This is a significant issue that is related to the core of the generation process "the similarity theory". Although this phenomenon does not weaken the validity of the similarity theory, it highlights the need for more sophisticated application of similarity. Hence, different patterns of similarity between the options must be realised as this could affect the prediction of difficulty. We are planning to validate the effect of clustered distractors on difficulty and to develop strategies to avoid or highlight such distractors when generating questions.

Acknowledgments. The authors would like to thank Tahani Alsubait for sharing the MCQ generator code.

References

1. Alsubait, T., Parsia, B., and Sattler, U.: Generating Multiple Choice Questions From Ontologies: Lessons Learnt. In: OWLED, pp. 73-84. Chicago (2014).
2. Alsubait, T., Parsia, B., and Sattler, U.: Generating Multiple Choice Questions From Ontologies: How Far Can We Go?. In: International Conference on Knowledge Engineering and Knowledge Management, pp. 66-79. Chicago (2014).
3. Haladyna, T. M., Downing, S. M., and Rodriguez, M. C.: A review of multiple-choice item-writing guidelines for classroom assessment. Applied measurement in education. 15(3), 309-333 (2002).
4. Pho, V.-M., Andre, T., Ligozat, A.-L., Grau, B., Illouz, G., Francois, T., et al.: Multiple choice question corpus analysis for distractor characterization. In: LREC, pp. 4284-4291. Reykjavik (2014).

Appendix A: Question Categories

Table 12. Explaination of the six question categories genertaed by the MCQ generator (adapted from [1]).

Question category	Stem	Key	Distractors
Generalisation	What is X? where X is an atomic concept name	an atomic subsumer of X	atomic non-subsumers of X
Generalisation 2	What is X? where X is an atomic concept name	a complex subsumer (concept expression) of X	complex non-subsumers of X
Specification	Which is X? where X is an atomic concept name	an atomic subsumee of X	non-subsumees of X excluding subsumers and siblings of the stem
Specification 2	Which is X? where X is a complex concept	an atomic subsumee of X	non-subsumees of X excluding subsumers of the stem
Definition	Which term can be defined as 'annotation'	an atomic concept name annotated with the annotation	atomic concept names not annotated with the annotation
Recognition	Which is odd?	an atomic concept name not subsumed by X where X is a concept name	atomic concept names subsumed by X

Appendix B: Example Questions

Syntactic Clues

Examples of the form (SK)

Stem: State Transition Network ...:
A. is Produced By some Concept Map Technique
B. is Produced By some Process Map Technique
C. is Produced By some State Transition Technique ◀ **Key**

Examples of the form (SD)

Stem: Repertory Grid Stage 2 ...
A. involves Providing A Running Commentary
B. involves Repertory Grid Stage 1
C. involves Rating Concepts Against Attributes ◀ **Key**

Stem: Which of the following terms can be defined by "a Java keyword used to declare a variable that holds an 8 bit signed integer"?

A. Char
B. Short
C. Int
D. Byte ◀ **Key**

Examples of the form (ANT)

Stem: Which of the following is [a][3] Binary Operator?

A. Unary operator
B. Unary minus operator
C. Equality operator ◀ **Key**
D. Logical complement operator

Syntactic Consistency

Stem: What is [a] Book?

A. (Is)VB (A)DT
B. (Has)VB (Part)NN
C. (Concept)NN ◀ **Key**

Stem: Which of the following is [a] Java Language Feature?

A. (Recursion)NN ◀ **Key**
B. (Implementation)NN
C. (Requirement)NN (analysis)NN
D. (Throw)VB

Note that in the previous example, although (D) is inconsistent with the key, it is indeed a Java language feature.

Stem: Which of the following terms can be defined by "A binary remainder operator that produces a pure value that is the remainder from an implied division of its operands"?

A. (Divide)VB
B. (Multiply)VB
C. (Modulus)NN ◀ **Key**

In this example, both distractors (A) and (B) are inconsistent with the key but they are both plausible. By investigating the ontology, we found that this issue resulted from the inconsistent naming of concepts.

[3] This is a grammatical correction that is manually added to the question.

Semantic homogeneity

Stem: Which of the following terms can be defined by "A layout manager that allows subcomponents to be added in up to five places specified by constants NORTH, SOUTH, EAST, WEST and CENTER"?

A. Simple Object (heterogeneous)
B. Event (heterogeneous)
C. Border Layout (homogeneous) ◄ **Key**
D. Grid Layout (homogeneous)

It is clearly deduced from the previous question that the expected answer is a layout manager. As can be seen, distractors (A) and (B) are heterogeneous in relation to the key type while option (D) is homogeneous.

Clustered Distractors

Stem: Which of the following terms can be defined by "A stage in the software development process where customer needs are translated into how it could be implemented"?

A. Testing
B. Unit Testing
C. Implementation
D. Design ◄ **Key**

Distractors (A) and (B) are clustered since knowing that the answer is not testing will allow the elimination of all types of testing.

Stem: Protocol Analysis Technique ...

A. involves Repertory Grid Stage 1
B. involves Repertory Grid Stage 2
C. involves Repertory Grid Stage 4
D. involves Identifying Knowledge Objects ◄ **Key**

Stem: Which of the following is produces some Protocol?

A. Attribute Laddering
B. Process Laddering
C. Laddering
D. Semi-structured Interview ◄ **Key**