Automatic generation of analogy questions for student assessment: an Ontology-based approach

Tahani Alsubait*, Bijan Parsia and Uli Sattler

School of Computer Science, The University of Manchester, Manchester, UK

(Received 13 March 2012; final version received 19 June 2012)

Different computational models for generating analogies of the form "A is to B as C is to D" have been proposed over the past 35 years. However, analogy generation is a challenging problem that requires further research. In this article, we present a new approach for generating analogies in Multiple Choice Question (MCQ) format that can be used for students' assessment. We propose to use existing high-quality ontologies as a source for mining analogies to avoid the classic problem of hand-coding concepts in previous methods. We also describe the characteristics of a good analogy question and report on experiments carried out to evaluate the new approach.

Keywords: e-assessment; ontology; analogy questions; relational similarity; vector space model; corpus-based evaluation

Introduction

Effective assessment of students is an ongoing process that should be carried out in different phases of education: planning as in diagnostic assessment, teaching and learning as in formative and self-assessment, reporting and recording as in summative assessment. At the same time, designing and implementing effective assessments, with increased numbers and higher expectations of students, is time consuming and expensive (i.e. hard). Adding an "e" prefix to assessment is not magical; interested practitioners still face some knotty problems. Typically, e-assessment refers to using technology to manage and deliver assessment. It can also provide automatic grading and instant feedback, especially with objective tests (e.g. multiple-choice). However, a major and yet unsolved problem of e-assessment is the generation of high-quality assessment items automatically (or at least semi-automatically). We argue that moving from a delivery model to a generation model is the key to the transition from e-assessment systems of today to those of the next generation. Moreover, technology-aided generation of assessment items is useful only if backed by an accepted pedagogical theory, which is usually missing in current generation models. In fact, this applies to both automatic and manual generation methods. For example, Paxton (Paxton 2001) carried out some empirical evaluations and reported that multiple-choice tests are often not well-constructed.

Part of the problem is the dearth of evaluation metrics. One possible solution is to use Item Response Theory (IRT) (Kehoe 1995; Miller, Linn, and Gronlund 2008; Mitkov, An Ha, and Karamani 2006) which describes the statistical behaviour of

*Corresponding author. Email: tmsubait@gmail.com

ISBN 978-91-977071-4-5 (print), 978-91-977071-5-2 (online)

http://dx.doi.org/10.3402/rlt.v20i0.19198

good/bad questions by following a procedure to measure three parameters: (1) possibility of guessing the correct answer, (2) tuned difficulty of test items and (3) proper discrimination between good and poor students. IRT can guide us in the evaluation of test items. However, we need a theory that guides us in the generation-process of test items that conform to the desired characteristics. A possible solution that we conjectured is to use a similarity-based approach to generate questions of different characteristics. For example, it is expected that questions with high similarity between the stem and key parts and less similarity between stem and distractors are easy questions (or perhaps guessable). Note that in MCQ terminology, the question part is called the stem, the correct answer is called the key and wrong answers are called distractors. Similarly, the question would be more difficult if the distractors were more similar to the stem compared to the key answer (e.g. lexical similarity).

To alleviate the burden of manual generation of assessment items, we propose an approach to automatically generate MCQs from Description Logics (DL) (Baader *et al.* 2007) ontologies. DL ontologies are engineering artefacts that provide formal and machine processable descriptions of the basic notions of a domain of interest. Many high-quality ontologies already exist, which suggests that mining such rich resources for assessment questions might be fruitful. Recently, a handful of studies explored the generation of MCQs from ontologies (Al-Yahya 2011; Fairon 1999; Holohan 2005, 2006; Papasalouros, Kotis, and Kanaris 2008; Zitko *et al.* 2008; Zoumpatianos, Papasalouros, and Kotis 2011), but very little research has been done on theoretical, empirical and evaluation aspects. Most of the proposed methodologies generate questions of the form "What is X?" or "Which of the following is an example of X?" based on class–subclass and/or class–individual relationships. These types of questions can be criticised as assessing lower levels only (e.g. recall) of Bloom's taxonomy of learning objectives (Bloom and Krathwohl 1956). Moreover, it is unlikely that a real test will consist of items that are all of this kind; hence, it is crucial to design approaches capable of generating questions of other kinds.

In this article, we describe the design and report on evaluation of a new approach for generating questions that require higher cognitive ability such as retrieving and mapping analogies of the form "A is to B as C is to D".

Analogy questions

Analogical reasoning is based on comparing two different types of objects and identifying points of resemblance. Hence, similarity plays a major role in analogical reasoning. In multiple-choice analogy questions, the student is given a pair of words and is asked to identify the most analogous pair of words among a set of alternative options. The required task is to recognise the relation between the pair of words in the stem and to find the pair of words that has a similar underlying relation. Multiple-choice analogy questions are used in various educational tests (e.g. college entrance tests such as SAT, GRE). As an example, see the question (GREguide 2012) in Table 1 taken from a sample of GRE verbal analogy questions:

Different computational models (Falkenhainer 1988; Gentner 1983; Larkey & Love 2003; Winston 1980) for analogy-making have been proposed over the past 35 years. These models are based typically on comparing two structured representations encoded in predicate logic statements [e.g. Structure Mapping Theory (SMT) (Falkenhainer, Forbus, and Gentner 1989; Gentner 1983)]. The SMT is more

Table 1. A Sample multiple-choice analogy question [GREguide (2012)].

Stem	CUTLERY: KNIFE::
Choices	(A) machinery: fuel (B) lumber: saw (C) furniture: chair (D) suitcase: handle
Key	(C) furniture: chair

sensitive to higher order relations (e.g. cause, imply). These models are founded on the premise that detecting analogies are useful for transferring knowledge between two domains (usually called base and target). In this article, we take a different approach: first we define *Analogy* as a function that takes two representations and returns a numerical value [0,1] representing their analogy. Examples of such functions will be discussed later. Then we show how to use this function to develop an MCQ generator that is capable of controlling the difficulty of questions. In addition, the Analogy function can be used to generate only plausible (i.e. expected to be functional) distractors. To achieve this, we use thresholds Δ_1, Δ_2, Δ_3, to parameterise our notion of analogy question (see Definition 1 below). We also define the function *Relatedness* that takes two concepts and returns their relatedness value [0,1]. This function can be used to filter the generated pairs in the stem, key and distractors according to a threshold Δ_R. Again, examples will be discussed later.

Definition 1 Let Q be an analogy question with stem $S = (A,B)$, key $K = (X,Y)$ and a set of distractors $D = \{D_i = (E_i,F_i) \mid 1 < i \leq max\}$. We assume that Q satisfies the following conditions:

(1) *The stem S, the key K, the distractor D_i are all good (i.e. Relatedness(A,B) $\geq \Delta_R$, Relatedness(X,Y) $\geq \Delta_R$, Relatedness(E_i,F_i) $\geq \Delta_R$).*

(2) *The key K is significantly more analogous to S compared to the distractors (i.e. Analogy(S,K) $\geq$ Analogy(S,D_i) $+ \Delta_1$).*

(3) *The key K is sufficiently analogous to S (i.e. Analogy(S,K) $\geq \Delta_2$).*

(4) *The distractors should be analogous to S to an extent (i.e. Analogy(S,D_i) $\geq \Delta_3$).*

(5) *Each distractor D_i is unique (i.e. Analogy(S,D_i) $\neq$ Analogy(S,D_j)).*

As an example of a *Relatedness* function, one can consider pairs of class names that are referenced together in at least one ontological axiom (perhaps in different sides of the axiom) as closely related classes. For instance, if we have an axiom in our ontology that defines X in terms of Y (e.g. $X \sqsubseteq \exists r.Y$) then *Relatedness*(X,Y) is greater than zero. For our current purposes, we designed a *Relatedness* function that captures class-subclass relations between pairs of named classes that correspond to one of the structures in Figure 1. As you might notice, we restricted our attention to those structures that have at most one change in direction and at most two steps in each direction. We also ignored some structures caused by multiple inheritances (e.g. 1d1u). These restrictions were considered to avoid too difficult (and probably confusing) questions. Also, these restrictions seem to be more aligned with human-generated analogy examples. While in the most general case, one should consider

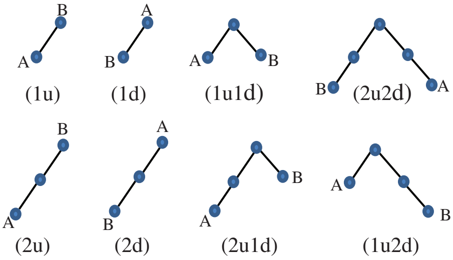

Figure 1. Closely related structures of class-subclass relations [labels represent no. of steps and direction (up or down)].

pairs with arbitrary related classes (e.g. by considering user-defined relations), for current purposes we only consider class-subclass relations. This simplifies the problem considerably in several dimensions while still generating quite a few candidate pairs (as we will see later).

As discussed above, we would like to be able to control the difficulty of the questions. According to Definition 1 and Propositions 1, 2, 3 we can control the difficulty of Q by increasing or decreasing Δ_1, Δ_2 and Δ_3.

Proposition 1 *Increasing* Δ_1 *decreases the difficulty of* Q*.*

Proposition 2 *Increasing* Δ_2 *decreases the difficulty of* Q*.*

Proposition 3 *Decreasing* Δ_3 *decreases the difficulty of* Q*.*

The *Analogy* function can be defined in different ways. For example, we can compare the number of steps between classes in each pair; pairs with similar number of steps in their representations would be more analogous. In this paper, we define the function *Analogy* in terms of similarities in number of steps and changes in direction (see Definition 2).

Definition 2 Let Analogy(x,y) be a function that takes two pairs of concepts and returns a numerical score for their analogy value [0,1]. The score is determined according to Table 2 in which values are derived from equation 1 as follows:

$$\begin{array}{lll}
Analogy(x,y) & = SS/TS \times SD/TD & (1)\\
SS & = Shared\ Steps(x,y) & (2)\\
TS & = Total\ Steps(x,y) & (3)\\
SD & = Shared\ Directions(x,y) & (4)\\
TD & = Total\ Directions(x,y) & (5)
\end{array}$$

Table 2. Values returned by the proposed function Analogy(x,y).

	1u	1d	2u	2d	1u1d	2u1d	1u2d	2u2d
1u	1	0	1/2*	0	1/4	1/9	1/9	1/16
1d	0	1	0	1/2*	1/4	1/9	1/9	1/16
2u	1/2*	0	1	0	1/3	4/9	1/6	1/4
2d	0	1/2*	0	1	1/3	1/6	4/9	1/4
1u1d	1/4	1/4	1/3	1/3	1	4/9	4/9	1/2*
2u1d	1/9	1/9	4/9	1/6	4/9	1	1/2	9/16
1u2d	1/9	1/9	1/6	4/9	4/9	1/2	1	9/16
2u2d	1/16	1/16	1/4	1/4	1/2*	9/16	9/16	1

*These values were not calculated using equation 1 but were manually coded because they correspond to similar but scaled structures.

Question generation

Our proposed approach to the generation of multiple-choice analogy questions consists of two phases: (1) extraction of interesting pairs of concepts by using the *Relatedness* function, those pairs can be used as stems, keys or distractors and (2) generation of multiple-choice questions based on the similarity between pairs which can be derived from the proposed *Analogy* function. The general algorithm is presented below (see Algorithm 1). The difficulty of the generated questions can be controlled by setting the parameters Δ_1, Δ_2 and Δ_3. In addition, the number of distractors can be controlled by setting the parameter "max". Note that avoiding non-functional (i.e. not picked by any student) is preferred (Haladyna & Downing 1993; Paxton 2001).

Algorithm 1 Generate_Analogy_Question(Ontology O, Δ_R, Δ_1, Δ_2, Δ_3, max)

```
AQ = {}; i = 0;
For each pair of classes (A,B) in O s.t. Relatedness(A,B) $\geq \Delta_R$
    For each pair of classes (X,Y) in O s.t. (A,B) $\neq$(X,Y) and Relatedness(X,Y) $\geq \Delta_R$ and_ Analogy((A,B),(X,Y)) $\geq \Delta_2$
        Q.S = (A,B);
        Q.k = (X,Y);
        For each i $\leq$max
            Get a pair of classes ($E_i$,$F_i$) in O s.t. (A,B) $\neq$($E_i$,$F_i$) and Relatedness($E_i$,$F_i$) $\geq \Delta_R$ and_ Analogy((A,B),($E_i$,$F_i$)) $\geq \Delta_3$
            If Analogy((A,B),(X,Y)) $\geq$Analogy((A,B),(Ei,Fi)) + $\Delta_1$ and Unique($E_i$,$F_i$)
                Q.D = Q.D + ($E_i$,$F_i$);
                I + +;
            End If
        Next i
        AQ = AQ + Q;
    Next (X,Y)
Next (A,B)
Return AQ;
```

We used three different ontologies to test the proposed analogy-generation engine. The three ontologies are presented in Table 3 below with some basic ontology statistics. The first ontology is the Gene Ontology which is a structured vocabulary for the annotation of gene products. It has three main parts: (1) molecular function, (2) cellular component and (3) biological role. The second and third ontologies are the People & Pets Ontology and Pizza Ontology which are very simple ontologies that were built to be used in ontology development tutorials. The table shows the number of satisfiable classes in each ontology and the number of sample questions generated by the engine (this is only a representative sample of all the generated

Table 3. Basic ontology statistics.

	No. of classes	No. of questions	% correct ~
Gene Ontology	36146	25	8%
People & Pets	58	15	67%
Pizza Ontology	97	16	88%

questions). The table also shows the percentage of questions that our proposed solver agent can correctly solve. The details of the approach used to simulate question solving are explained in the following section.

Corpus-based evaluation

In order to evaluate the proposed approach for analogy generation, we follow the method explained by Turney and Littman (Turney and Littman 2005) for evaluating analogies using a large corpus. In their study, Turney and Littman reported that their method can solve about 47% of multiple-choice analogy questions (compared to an average of 57% correct answers solved by high school students). The solver takes a pair of words representing the stem of the question and five other pairs representing the answers presented to students. Their proposed method is inspired by the Vector Space Model (VSM) of informational retrieval. For each provided answer, the solver creates two vectors representing the stem (R_1) and the given answer (R_2). The solver returns a numerical value for the degree of analogy between the stem and the given answer. Then, the answers are ranked according to their analogy value and the answer with the highest rank is considered the correct answer. To create the vectors, they proposed a table of 64 joining terms that can be used to join the two words in each pair (stem or answer). The two words and joined by these joining terms in two different ways (e.g. "X is Y" and "Y is X") to create a vector of 128 features. The actual values stored in each vector are calculated by counting the frequencies of those constructed terms in a large corpus (e.g. web resources indexed by a search engine). To improve the accuracy of their proposed method, they suggested using the logarithm of the frequency instead of the frequency itself.

In this article, we follow a similar procedure. First, we constructed a table of joining terms relevant to the relations considered in our approach (e.g. "is a", "type", "and", "or"). Based on these joining terms, we create vectors of 10 features for the stem, the key and each distractor. The constructed terms are sent as a query to a search engine (Yahoo!) and the logarithm of the hit count is stored in the corresponding element in the vector. The hit count is always incremented by one to avoid getting undefined values. Following this procedure, our proposed solver agent solved 8% of the questions generated from the Gene Ontology, 67% of the questions generated from the People and Pets Ontology and 88% of the questions generated from the Pizza Ontology. We argue that this is caused by the specific terminology used in the Gene Ontology and lack of web resources that have information regarding it compared to the other ontologies.

Conclusion and future work

In this article, we presented a new approach for generating multiple-choice analogy questions from existing ontologies. We described the design of analogy-generator and analogy-solver. The solver achieved a maximum accuracy of 88%. However, it achieved a low accuracy value of 8% when used to solve analogies generated from the Gene Ontology. We assume that the difficulty of the domain is considered as an additional dimension to our difficulty controlling model.

For future work, we are going to generalise our approach for analogy generator to include user-defined relations. To evaluate analogies generated from arbitrary relations, we suggest using Latent Relational Similarity (LRS) (Turney 2005) which has the advantage of learning relations instead of using predefined joining terms.

References

Al-Yahya, M. (2011) 'Ontoque: a question generation engine for educational assessment based on domain ontologies', *11th IEEE International Conference on Advanced Learning Technologies*, Athens, Georgia, USA, pp. 393–395.

Baader, F., *et al.* (2007) *The Description Logic Handbook: Theory, Implementation and Applications*, 2nd edn, Cambridge University Press, Cambridge, UK.

Bloom, B. S. & Krathwohl, D. R. (1956) *Taxonomy of educational objectives: the classification of educational goals by a committee of college and university examiners. Handbook 1. Cognitive domain*, Addison-Wesley, New York, NY.

Fairon, C. (1999) 'A web-based system for automatic language skill assessment: evaling', *Proceedings of Computer Mediated Language Assessment and Evaluation in Natural Language Processing Workshop*, University of Maryland, USA, pp. 62–67.

Falkenhainer, B. (1988) *Learning from Physical Analogies*, PhD thesis, University of Illinois.

Falkenhainer, B., Forbus, K. & Gentner, D. (1989) 'The structure-mapping engine: Algorithm and examples', *Artificial Intelligence*, vol. 41, pp. 1–63.

Gentner, D. (1983) 'Structure-mapping: a theoretical framework for analogy', *Cognitive Science*, vol. 7, pp. 155–170.

GREguide. (2012) 'Graduate Record Examination Guide', Available at: http://www.greguide.com.

Haladyna, T. & Downing, S. (1993) 'How many options is enough for a multiple choice test item?', *Educational & Psychological Measurement*, vol. 53, no. 4, pp. 999–1010.

Holohan, E. e. a. (2005) 'Adaptive e-learning content generation based on semantic web technology', *Proceedings of Workshop on Applications of Semantic Web Technologies for e-Learning*, Amsterdam, pp. 29–36.

Holohan, E. e. a. (2006) 'The generation of e-learning exercise problems from subject ontologies', *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*, Kerkrade, Netherlands, pp. 967–969.

Kehoe, J. (1995) 'Basic item analysis for multiple-choice tests', *Practical Assessment, Research & Evaluation*, vol. 4, no. 10, Retrieved August 15, 2012 from http://PAREonline.net/getvn.asp?v=4&n=10.

Larkey, L. & Love, B. (2003) 'Cab: Connectionist analogy builder', *Cognitive Science*, vol. 27, pp. 781–794.

Miller, M., Linn, R. & Gronlund, N. (2008) *Measurement and Assessment in Teaching*, 10th edn, Pearson, Cambridge, UK.

Mitkov, R., An Ha, L. & Karamani, N. (2006) 'A computer-aided environment for generating multiple-choice test items. Cambridge university press', *Natural Language Engineering*, vol. 12, no. 2, pp. 177–194.

Papasalouros, A., Kotis, K. & Kanaris, K. (2008) 'Automatic generation of multiple-choice questions from domain ontologies', *IADIS e-Learning 2008 conference*, Amsterdam, pp. 427–434.

Paxton, M. (2001) 'A linguistic perspective on multiple choice questioning', *Assessment & Evaluation in Higher Education*, vol. 25, no. 2, pp. 109–119.

Turney, P. (2005) 'Measuring semantic similarity by latent relational analysis', *IJCAI is the International Joint Conference on Artificial Intelligence*, Edinburgh, UK, pp. 1136–1141.

Turney, P. & Littman, M. (2005) 'Corpus-based learning of analogies and semantic relations', *Machine Learning*, vol. 60, no. 1–3, pp. 251–278.

Winston, P. (1980) 'Learning and reasoning by analogy', *Communications of the ACM,* vol. 23, pp. 689–703.

Zitko, B., *et al.* (2008) 'Dynamic test generation over ontology-based knowledge representation in authoring shell', *Expert Systems with Applications: An International Journal*, vol. 36, no. 4, pp. 8185–8196.

Zoumpatianos, K., Papasalouros, A. & Kotis, K. (2011) 'Automated transformation of swrl rules into multiple-choice questions', *FLAIRS Conference 11*, Palm Beach, FL, USA, pp. 570–575.