Diagram Interpretation and e-Learning Systems

Neil Smith, Pete Thomas, and Kevin Waugh

Computing Research Centre, The Open University
Milton Keynes, MK7 6AA, UK
{n.smith, p.g.thomas, k.g.waugh}@open.ac.uk

Abstract. We describe a system capable of grading free-form diagrammatic answers. Our matches meaningful parts of a diagram with equivalents in a model solution. This is complicated by errors, omissions, and superfluous items in the student answer. The result of matching is used to calculate the grade and generate appropriate feedback; it performs at least as well as a human marker on a variety of diagram types. We describe tools that allow the easy creation of questions, marking schemes, and diagram editors suitable for embedding in a VLE quiz engine.

Key words: automatic grading, imprecise diagrams, e-learning

1 Introduction

We have produced a system that automatically assesses (summatively and formatively) student-produced diagrams. Such diagrams are normally *imprecise*: they do not match the expected diagram in some way. We handle graph-based diagrams commonly found in technical subjects (e.g. UML class and sequence diagrams). Such diagrams encode meaning in the connection of lines and boxes, adornments to diagram elements, and the spatial positioning of elements.

Our work contrasts with semi-automatic marking systems [1, 2] (which present the human marker with an ungraded answer and apply the same grade to equivalent answers) and deduction systems for diagrammatic formulae [3]. In our work, the emphasis is on extracting as much information as possible from an imprecise diagram.

Our approach compares a student *answer* diagram with one or more model *solution* diagrams. Comparison is based on identifying the *minimal meaningful units* (MMUs) in each diagram. An MMU is a partial diagram which, if any element is deleted, no longer has meaning. The first step is to identify MMUs in the answer that correspond to MMUs in the solution. However, as the student diagram is imprecise, we use a measure of similarity between MMUs, based on properties of the MMUs and, most significantly, the identifying label.

By default, we model object labels as noun phrases and relationship labels as verb phrases. Processing identifies head words in the label, which are stemmed and compared using edit distance, taking synonyms into account. This process also deals with punctuation and abbreviations.

Handling synonyms is difficult as there are many equivalents for terms occurring in the answer. Common and specialist synonyms are stated explicitly in the mark scheme. The multiple synonym problem [4] is ameliorated by stemming (to reduce words to a canonical form) and similarity measures (to estimate the closeness of words). We can capture many synonyms from corpora of student answers [5, 6].

2 The Matching Process

The similarity between two diagrams is determined by finding the best overall correspondence of MMUs. As far as possible, each MMU in the answer is matched with an MMU in the solution to give a diagram correspondence. This will be incomplete due to omissions and errors in the answer. The chosen diagram correspondence is the one which maximises the sum of the similarities of matched MMUs. MMUs unmatched after this stage are compared on the basis of their context (the set of related objects). Once a correspondence has been determined, a mark scheme, based on MMUs, is applied and feedback is given.

Thresholds are used to judge whether or not two aspects are sufficiently similar to be considered to match. Thresholds are also used in marking to determine whether to award marks for some aspect of an answer. Weights indicate the relative importance of diagram elements for both matching and marking.

Experiments performed on corpora of several hundred student answers show that our system performs at least as well as a human marker [5].

3 Supporting Applications

Question setting, answering, and marking are performed by domain-specific tools. Using another application, the teacher specifies the drawing tool (Fig. 1) by describing the domain and the amount of freedom students have to create incorrect diagrams, based on pedagogic requirements. The teacher must also specify the characteristics of the marking tool, including relevant weights and thresholds. Another application supports the teacher in developing questions, model solutions, mark schemes, and feedback. The marking, drawing, and question-setting tools are all derived automatically from generic ones, based on the specifications given.

Prototypes have been incorporated into the Open University's Moodle VLE. The VLE presents the student with a question and an applet for drawing an answer. The applet submits the finished diagram to the making engine. In formative mode, the marking engine returns feedback which is displayed in the browser and the student may make multiple attempts at the question.

4 Conclusions and Future Work

We have built an automatic marking engine for graph-based diagrams which gives good performance and good agreement with human markers. Neverthe-

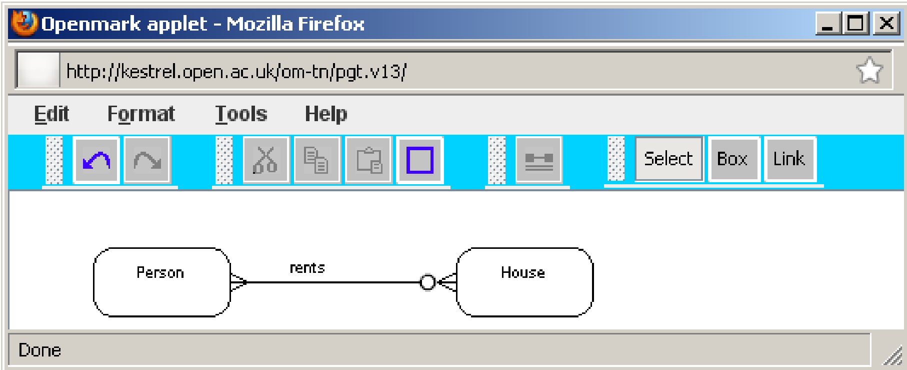

Fig. 1. The diagram drawing tool. It can run stand-alone or as an applet

less, we want to perform similar experiments with more diagram types and less constrained questions.

Our work to date has been based on developing bespoke diagram parsers generalising them to support a wider range of diagrams. While effective and useful, this approach will be limiting in future. Therefore, we will reformulate our approach as a constraint-based parser (e.g [7]), extended to use soft constraints to handle imprecision.

References

1. Batmaz, F., Hinde, C. J.: A Web-Based Semi-Automatic Assessment Tool for Conceptual Database Diagrams. In: Proceedings of the Sixth IASTED International Conference on Web-Based Education, pp.427–432 (2007)
2. Higgins, C. A., Bligh, B.: Formative Computer Based Assessment in Diagram Based Domains. In: Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE), pp.98–102 (2006)
3. Howse, J., Stapleton, G.: Visual Mathematics: Diagrammatic Formalization and Proof. In: Proceedings of International Conference on Mathematical Knowledge Management (2008)
4. Jayal, A., Shepperd, M.: The Problem of Labels in e-Assessment of Diagrams. ACM J. Educational Resources in Computing 8(4) (2008)
5. Thomas, P., Smith, N., Waugh, K.: Automatically assessing graph-based diagrams. J. Learning, Media & Technology 33(3), 249–267 (2008)
6. Jordan, S.: Assessment for learning: pushing the boundaries of computer-based assessment. Practitioner Research in Higher Education, 3(1), 11–19 (2009)
7. Henning, C.: CHR grammars. Theory and Practice of Logic Programming. 5(4–5), 467–501 (2005)