A HUMAN-COMPUTER COLLABORATIVE APPROACH TO THE MARKING OF FREE TEXT ANSWERS

John Sargeant, Mary McGee Wood and Stuart M. Anderson

A human-computer collaborative approach to the marking of free text answers

John Sargeant, Mary McGee Wood and Stuart M. Anderson
Department of Computer Science
University of Manchester
Kilburn Building
Oxford Road
MANCHESTER M13 9PL
johns@cs.man.ac.uk, mary@cs.man.ac.uk and
andersos@cs.man.ac.uk

Abstract

We propose the term Human-Computer Collaborative Assessment (HCCA) for a distinct and currently rather neglected sub-field of CAA. In HCCA answers are constructions rather than selections, and marking is a process of active collaboration between human marker and machine. We present the results of experience with simple tools which demonstrate significant time savings compared to traditional paper marking. Further improvements in both speed and quality of assessment are clearly possible, but require much more sophisticated tools, particularly for free text answers. We review the role which Natural Language Processing techniques can play, particularly in the light of experience from other domains. Analysis of a number of answer sets highlights key issues in HCCA as well as underlining the infeasibility of fully automatic marking in many situations.

Introduction

In this paper we are concerned with constructed answers, in particular free text, rather than multiple choice questions (MCQs). It is often preferable to ask students to construct something rather than making a choice among a fixed set of alternatives. Even if some construction does underlie the answer to an MCQ, that construction is lost. This limits their usefulness even for formative assessment, as it is difficult to give useful feedback if the reasoning behind a wrong answer is lost. Furthermore, *the same applies traditional marking*, which is primarily a process of assigning numbers within some range, i.e. answering a series of MCQs! The construct behind the assignment of a particular mark is often not recorded, because to do so systematically is very time consuming.

MCQs and similar selection-based question types are nevertheless dominant in CAA practice, because they remove the burden of marking completely. In principle we can imagine constructed answers being marked *autonomously* by

software armed with Natural Language Processing (NLP) and perhaps machine learning techniques, provided with some initial parameters and then left to “get on with it.” We are primarily interested in summative assessment, and in non-trivial content rather than style. In this context we argue that autonomous marking of free text to the level of accuracy required is an “AI-complete” problem. Similar considerations apply to other types of constructed answers such as diagrams, equations, programs etc, although these are beyond the scope of this paper.

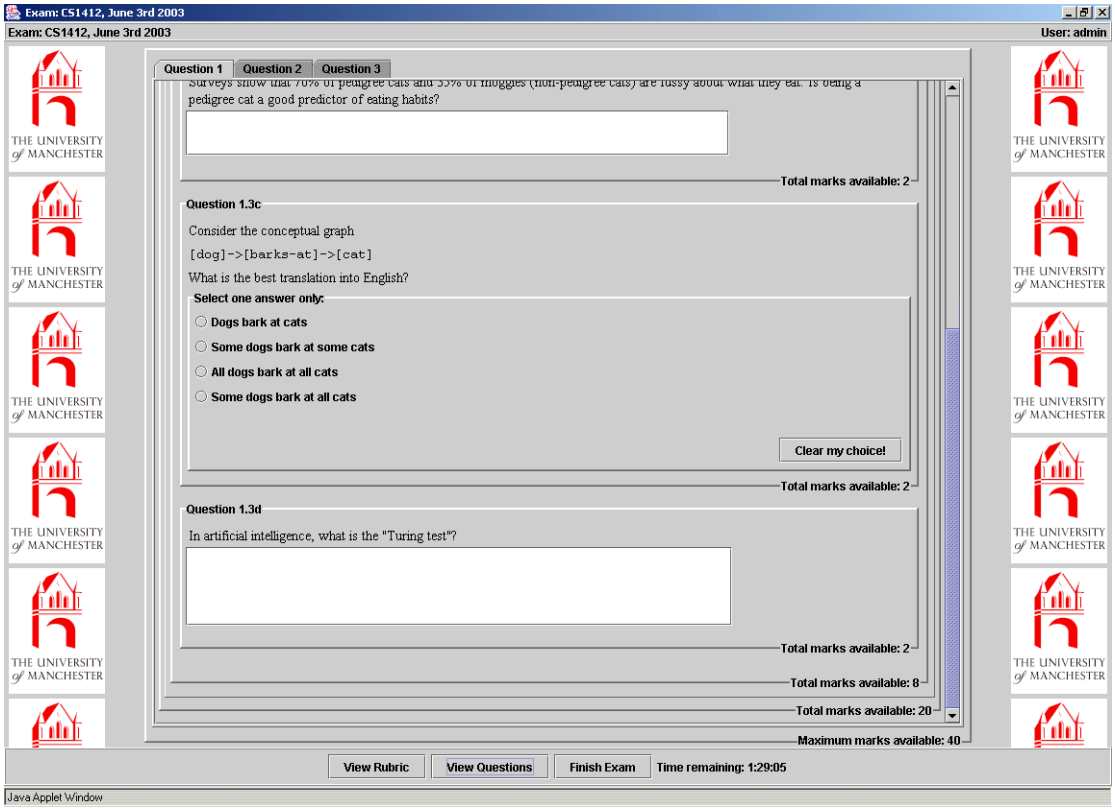
An alternative to autonomous marking is a human-computer partnership where the machine takes away much of the drudgery of the process and detects similarity between answers, while the human makes the important judgements. We call this approach Human-Computer Collaborative Assessment.

A key idea is that a representation of possible answers and their scores – the Answer Representation (AR) - is grown and refined dynamically as part of this process, not fixed in advance. The primary purpose of an AR would be to identify repeated answers or part-answers, and hence speed up the marking process. It would encode the reasoning behind marks, and hence could be used in moderation or double marking, mark justification, plagiarism detection, and formative feedback. It could also be used to measure the quality of questions: for instance, if there were too many alternative answers, the question statement could be tightened up accordingly. When reusing questions or part-questions, the existing AR, or part of it, could also be reused. An existing AR covering most new answers could be used to give automatic formative feedback. At some point it might even be usable for autonomous summative marking.

At this point it is an open question whether a single AR (based on AND/OR trees, for instance) is appropriate for a wide range of answer types, or whether ARs have to be specialised: we are investigating a variety of approaches. In this paper we focus on the evidence in favour of the HCCA approach, from the history of comparable enterprises – in particular machine translation - and from data collected in online examinations.

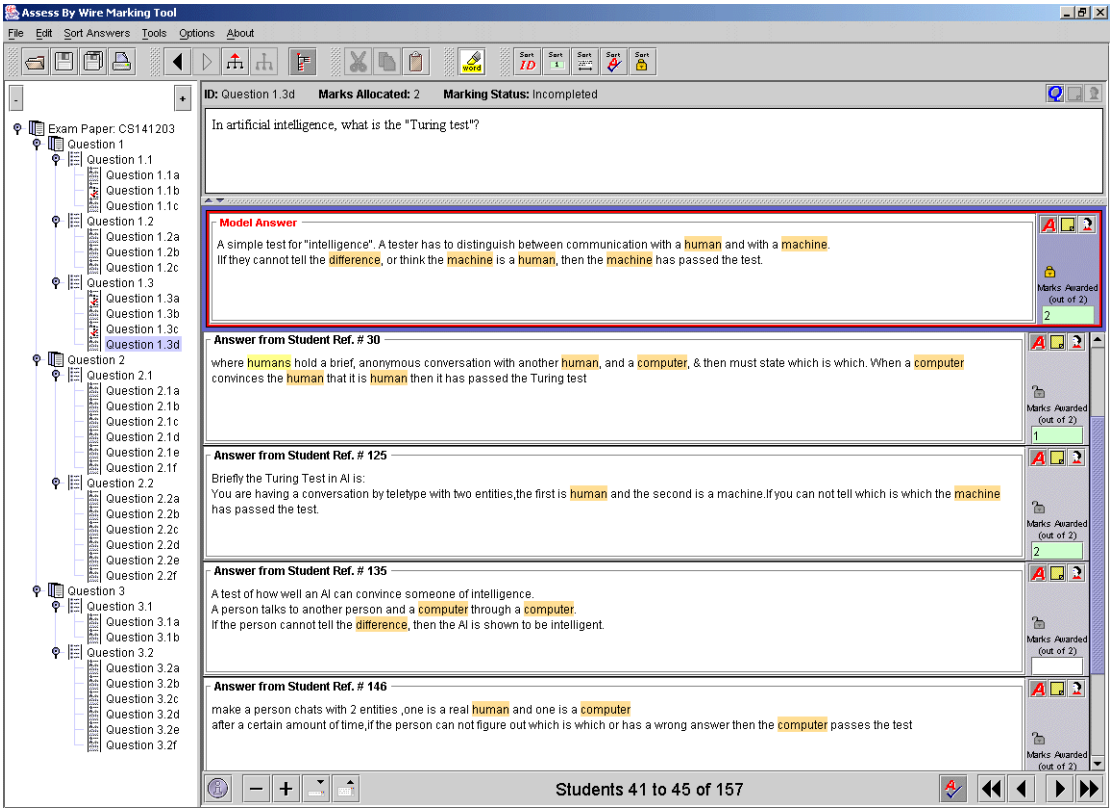
Experience with simple tools

Assess by Wire (ABW) is a set of tools for setting, administering and marking online exams. Questions can be arbitrarily nested, and can include mixtures of MCQs, text answers, and diagrammatic answers. Exams are set using a GUI setting tool. They are held (as XML) on a server, and are taken over the Internet via a Java Applet or on a local network using a Java application. In the latter case a specialised Linux environment designed for the purpose guarantees a high level of technical security. The screenshot shows the student GUI. The user interface is deliberately very simple, and has a distinctive appearance designed to aid invigilators.



The system has been used for several summative exams, most at masters level, both locally, and over the Internet for distance learning courses. The data used in this paper comes from a first year undergraduate exam in Artificial Intelligence (hereafter “the AI exam”) taken by 157 students.

The most interesting part of the system is the marking tool, developed by the third author[1]. This allows the marker to navigate through the exam, marking all answers to each part-question together, and provides various features such as the ability to highlight keywords and order the answers in various ways. Standard functionality such as automatic marking of MCQs and output of mark data is also provided. The second screenshot shows the marking tool in action.



This user interface is designed for “expert” users and shows the question tree explicitly. Elimination of handwriting deciphering, script shuffling etc. makes marking of text answers with the tool at least twice as fast as marking traditional scripts, and marking individual part-answers together, as well as genuine anonymisation, potentially improves consistency.

50 years of NLP: what have we learned?

A number of approaches to Automatic Essay Scoring (AES) are surveyed in [2]. This contains many results of the form “the system’s marks correlated better with the average of a set of human judges than the human judges did with each other”. However, this is largely because the correlations between human judges were very low, as the judgements were being made on writing style and other highly subjective factors.

When marking for specific content, we should expect much greater accuracy, making the task much more difficult. There has been success in some subject areas, such as parts of medicine [3], where precise technical terms must appear in an answer and little variation is acceptable. Even in this case manual moderation of questions is performed, potentially changing the Marking Guidelines (their Answer Representation).

The task has similarities to language translation, and the history of Machine Translation (MT) is revealing. The first practical application envisaged, by Bar-Hillel in 1951[4] for NLP was MT, and the Holy Grail of "fully automatic high-quality machine translation", or FAHQMT, was believed by some to be quickly

achievable. Experience showed otherwise, and eventually it was generally accepted that FAHQMT -- or indeed FAHQM any application of NLP -- was an "AI-complete" task which would not be achieved for years to come, if ever. In fact Bar-Hillel's initial paper already recognised the potential of human/machine partnership in the task of translation.

The early systems required a human "post-editor" to revise the output text with constant reference to the input. [5] describes the reactions of professional translators to the introduction of early commercial MT systems. These changed quickly from fear of being made superfluous to positive acceptance as they saw that the systems took over only the more literal and repetitious aspects of the task, freeing them up to concentrate on the interesting problems.

An extreme example is the METEO system [6], installed by the Canadian meteorological office in 1977 to translate weather forecasts between English and French. The task was peculiarly routine and tedious – probably not dissimilar to marking exams - and the average stay in post of the translators was six months. After the system was installed, this rose to two years. The system was still in use, processing more than 30 million words a year, twenty years later.

METEO succeeded by exploiting the limitations of a tiny, highly specialised domain and sub-language. Fully automatic translation can also be valuable for more "normal" text if high accuracy is not essential. For instance BabelFish [7] and other similar tools produce approximate translations which are good enough to support a sort of "browsing" of foreign language web pages (and are frequently good for a few laughs too).

In time it was recognized that accurate assessment of the abilities and limitations of software systems is a foundation for the design of collaborative systems in which the machine and the user share the task in such a way as to make best use of their distinct abilities.

The Ntran system, developed at UMIST from 1984 to 1988 [8], enabled a monolingual English user to translate English text accurately into Japanese. In accordance with the philosophy set out in [9], the system was interactive: it carried out as much of the task of translation as it was sure of, but stopped to ask the user for help when necessary. Thus, when an input sentence was syntactically ambiguous, the system would present paraphrases for the possible interpretations, and the user would select the one intended. If the system encountered an unknown word, it would offer the user a structured framework in which to add that word to its dictionary. The user thereby contributed carefully constrained, but essential understanding and ability to the overall performance of the collaborative human/machine system.

A branch of NLP with more direct relevance is Information Extraction (IE) [10]: extracting information from text to fill a pre-defined template. IE systems are, deliberately or otherwise, tuned to the domain for which they were first designed – for funding reasons this is usually articles from the Wall Street

Journal. For instance the MultiFlora project [11] took an existing IE system, built in the GATE NLP environment [12], and used it to parse botanical texts. A great deal of manual tuning was required to achieve even enough accuracy to make the system usable to human experts. Although, even in the domains to which they are tuned, current IE systems have precision and recall rates far below that required for autonomous summative marking, they could have a role to play in HCCA.

Evidence from students' answers

In the following sections we present a number of answers of increasing levels of complexity, drawn from the AI exam, and discuss the issues they raise for HCCA. The format of the exam was designed to be as similar as possible to the previous year's which had been done on paper. It was split into main questions, each with sub questions, and most of the sub questions were in several parts. As a result the basic "unit of answer" as very small, normally worth only one or two marks.

Eighteen percent

One sub-question was on Bayesian probability. For one of its harder parts, 0.18 or 18% was both the correct answer and the proportion of the students who got it right. The students were allowed to provide either the answer or the calculation leading to the answer, which was `(0.3/0.05)*0.03` or `(30%*3%)/5%)` etc. There were a large number of distinct answers, particularly as most of them were wrong. Treated as text strings there were 117 distinct answers out of a total of 144 attempts. The longest answer was about 200 characters long.

This shows that *even trivial free-text answers cannot be trivially marked*. Autonomous marking would require an arithmetic package able to cope with a wide range of variation. For summative purposes we would need considerable experience before such marking could be deemed safe, although as an assistant to a human marker it would be very useful.

We could constrain the form of the answer, rather than allowing arbitrary free text. The question could be phrased as an MCQ, or the user interface could be constrained to allow only numbers in a particular format. It is usually possible to require an answer in some more constrained format than free text. Whether this is desirable depends on circumstances. One issue is whether we want students to show their working or reasoning. Another that the greater the information content of an answer, the greater the chance of detecting plagiarism; text answers have more information content than more constrained types, as they are both longer and, may be far more varied.

In general, the setter of an exam should be able to choose the most appropriate form of answer on educational rather than technology grounds.

Production rules

Another question was “What are the components of a production system”, and the standard answer was “Rule memory, working memory, interpreter”. This had far less wrong answers - the average was 80% - but several students added redundant extra information. The longest answer, which got full marks, was 105 words long!

The natural approach to autonomous marking of this question is to search for the three keyphrases in the students’ answers and award a mark for each one found. If we require a literal match (ignoring case) such a process would incorrectly mark at least part of an answer wrong in 37 of the 151 cases.

One problem is mis-spellings of correct answers: this is very common, particularly among dyslexic students and non-native English writers. In this example the word most commonly mis-spelt was “interpreter”. The standard way of dealing with this is to calculate an edit distance: the minimum cost in terms of single-character edits required to convert one word to another. The definition we use assigns a cost of 1 to each insertion, deletion, or substitution of a character, .e.g. “interpretor” has an edit distance of 1. Using this measure there were 11 cases with an edit distance of 1, 1 at distance 2, 2 at distance 3, and 1 (“interaper”) at distance 4.

Suppose we consider anything within an edit distance of 2 to be correct. An autonomous marking system would therefore mark three correct answers as wrong. On the other hand in the context of HCCA this would safely allow 12 less answers to be shown to the human marker. Accepting a larger edit distance might be reasonable in this case, but often even 2 is too much – consider for instance “mode”, “model”, “modal”, “module” etc.

Sometimes we may be prepared to accept completely the wrong word. For instance the first author has accepted “patter” for “pattern” and “taxonomy” for “taxomania” in final year exam answers as the in each case context made it clear that the student understood the relevant concepts.

This leads to the second, and more difficult, problem of *context-dependent synonyms.* For instance, “inference engine” is a synonym for “interpreter”, used by 11 students. This was predictable, but the full range of synonyms which the second author accepted included several which were not:

- Working memory: knowledge base, fact memory, work space, world memory, main memory*, state memory, knowledge*, memory area*, data memory
- Rule memory: rule base, rule space, production-rule memory, rules store, rules*
- Interpreter: inference engine, inference component, rule selector, rule – selecting engine, reasoning*

The marked synonyms are particularly interesting as they were only acceptable in the context of a particular answer, not just in the context of the question or the subject. For instance, a number of students wrote simply “rules” rather than “rule memory” etc. In general this was not accepted, as it’s easy to guess that a production [rule] system contains rules. However, one student wrote “the knowledge, the rules which operate on the knowledge, and the interpreter that links these two”, and this was deemed acceptable – indeed it is arguably a better answer than the standard one as it shows understanding of the concepts rather than just remembering names.

This example is not atypical. *Context dependent synonyms are the norm* in any but the most trivial text answers, they cannot be predicted in advance, and human judgement is required to determine what they are.

An HCCA approach to marking this question, assuming that mis-spellings are automatically dealt with up to an edit distance of 2, would require a human marker to look at part ofrall of 59 of the 151 answers. Of the 92 which need not be shown, all but 2 correspond to the standard answer: the other 2 differ only in using “inference engine” for interpreter. The potential time saving is greater than these numbers suggest, as – with one bizarre exception – all the answers which include unnecessary extra information also include the required keyphrases – most of the time wrong answers are short (for this and most questions). Further improvement could be made by filtering out correct part-answers. However, this has to be done carefully, because of the dependence of keywords on their immediate context. So for this example the overall time saving of HCCA compared to paper marking is approximately a factor of 6.

A further advantage of the HCCA approach here is that it makes clear the set of choices leading to the marks – the set acceptable synonyms forms a simple Answer Representation. In principle such explicit recording of all choices could be part of a manual marking process, but without good software support this is cripplingly expensive. The first author attempted this for a manually marked final year exam paper. Although this exercise reinforced the points made here, particularly the ubiquity of context- dependent synonyms, if done systematically for the whole answer set it would have roughly tripled marking time, which was impractical.

Constraining the answer, apart from simply limiting the amount the students could type, would not make any significant difference. The reasons for this – and why converting the question to MCQ format is problematic – are left as an exercise for the reader.

The Turing test

Another question on the AI exam was “What, in artificial intelligence is the Turing Test? This represents a much greater challenge. The standard answer and a selection of student answers appear in the snapshot of the marking tool shown earlier, with keywords highlighted. The answers are a morass of

context-dependent synonyms: in particular “tester” appears *only* in the standard answer, most answers using “human” in that role, although various other synonyms such as “person” also appear. For the other role, “computer” and “program” amongst others are acceptable, whereas in most contexts we would want students to distinguish between hardware and software!

The deeper problem, in this and in most interesting cases, is that we require not just the right keywords, but a sequence of concepts to appear in a logical order. Nevertheless there is clearly scope for some automatic filtering of the answers, as phrases such as “cannot tell the difference” appear frequently. In general anybody who marks a large batch of exam papers is aware of marking “the same thing” repeatedly. We are currently working on an approach which measures both the occurrence of keywords/phrases and the distances between them, as a way of trying to capture the essential structure of answers of this type. Whatever method is used, the marking process has to take into account the possibility of unusual but creditworthy answers such as “A program which could mark this question autonomously would have passed the Turing Test”.

Conclusion

Human-computer collaborative assessment offers significant benefits over both traditional paper-based assessment and forms of CAA based on fully automatic marking:

- There is a great deal of flexibility in the way in which assessments are set. In particular the degree to which the form of answers is constrained is determined by educational rather than technological considerations.
- Marking can be considerably faster than for paper-based assessments.
- Consistency and accuracy in marking can be improved relative to either fully manual or fully automatic marking.
- The real reasons for the marks given can be explicitly recorded, and used for many purposes.

We have shown that some of these benefits can be obtained with fairly simple tools. However, to realise them in full requires much more sophisticated software, and in particular appropriate application of NLP techniques. We believe that there is a great deal of interesting and useful research to be done in this field.

References

[1] Anderson, S M, A tool to assist in the marking of structured examinations, MSc dissertation, Department of Computer Science, University of Manchester, 2002.

[2] Shermis & Burstien (eds) Automated Essay Scoring, a cross-disciplinary approach, Lawrence Erlbaum 2003

[3] Mitchell T, Aldridge N, Williamson W, Broomhead P, Computer based testing of medical knowledge, Preceedings of the 2003 CAA conference, pp 249-271.

[4] Bar-Hillel, Y. The State of Machine Translation in 1951. American Documentation, vol. 2, pp. 229-237.

[5] Lawson, Veronica. 1982. In Veronica Lawson,(ed), Practical Experience of Machine Translation

[6] B. Thouin. 1982. "*The Meteo System*". In Veronica Lawson, editor, Practical Experience of Machine Translation, pp. 39--44. North-Holland, Amsterdam, Holland.

[7] http://world.altavista.com

[8] Wood M M & Chandler, B J Machine translation for monolinguals Proceedings of Coling ’88 pp 760-763

[9] Johnson, R.L. and Whitelock, P. (1987): Machine translation as an expert task. In: Nirenburg, S. (ed.) *Machine translation: theoretical and methodological issues*. Cambridge: Cambridge Univ.Press, pp. 136-144.

[10] Cowie, J. & W. Lehnert (1996) Information Extraction. Communications of the ACM 39(1):80-91.

[11] Wood, M.M., S.J. Lydon, V. Tablan, D. Maynard, \& H. Cunningham. 2003. Using parallel texts to improve recall in IE. Proceedings of Recent Advances in Natural Language Processing, pp 505-512.

[12] Cunningham, H., D. Maynard, K. Bontcheva, & V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications.
Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, Philadelphia, USA 2002