On the Automated Assessment of Short Free-Text Responses

Raheel Siddiqi
PhD student, School of Computer Science, The University of Manchester, UK
E-mail: Raheel.Sidddiqi@postgrad.manchester.ac.uk
Christopher J. Harrison
School of Computer Science, The University of Manchester, UK
E-mail: christopher.j.harrison@manchester.ac.uk

Abstract

Systems for automating the assessment of textual answers have been available commercially since the mid 1990's and some progress has been made in their application to assessing short, factual answers to purpose-written questions. However, progress in the field is hindered by a lack of qualitative information regarding the effectiveness of such systems, with little (if any) performance statistics derived from the same data sets. We evaluate two currently available systems to identify their capabilities and limitations, and to highlight areas in which future related research may be usefully directed.

In addition to our analysis of two existing systems, we propose that a common repository of standardised data sets be created and made available to researchers and system developers, possibly via some overseeing authority, in order that progress in the field can be quantified and analysed.

Key Words: Automated assessment, short textual answers, performance, standardised data sets.

2. An introduction to automated short-answer marking

Free-text questions have traditionally been absent from computerised tests because they were considered to be very difficult to mark automatically. With the advent of new technology, such as advances in the field of natural language processing and information extraction (Hearst, 2000), it is possible to incorporate certain types of free-text questions in computerised tests as their reliable automated marking is now feasible. Key benefits of automating free-text marking include time and cost savings, and the reduction in (ideally, the elimination of) errors and unfairness due to bias, fatigue (on the part of the human marker) or lack of consistency.

This paper concentrates on short-answer marking systems rather than essay marking systems. Short-answer marking systems are designed for short, factual answers where there is a clear criterion for answers being right and wrong (Sukkarieh and Pulman, 2005). The award of marks is based on content rather than style. Poor writing quality is normally tolerated. Such answers vary in length from a few words to around four or five lines of text. Not all short-answer questions are appropriate for computerised marking. Situations where short-answer questions are inappropriate for computerised marking are:

- The correct response may be expressed in a large number of ways (i.e. the short-answer question is subjective).
- Responses are complex in nature (i.e. identification of correct and incorrect answers is not clear-cut).

An example of a short-answer question that is inappropriate for computerized testing is: *"Define the term 'Democracy'"*. There are numerous standard definitions of the term 'Democracy'. Moreover, various respondents may have their own perspective about the term and may define it differently. In other words, the expected responses will likely be subjective and not simply paraphrases of a single concept. The criterion of right and wrong for an answer is also unclear.

An example of a short-answer question appropriate for computerized testing is: *"How do we terminate a statement in Java"*. The correct answer is simply: *"A Java statement is terminated using a semicolon"*. Correct student responses are expected to be paraphrases of this concept and therefore, the primary task of the assessment software is to recognise which answers are paraphrases of the correct concept and which are not.

In some cases, short-answer questions considered unsuitable for computerized tests can be modified and adapted for such use in such tests. The following are two guidelines for modifying initially unsuitable questions:

- The short-answer question should try to constrain students to an answer involving only one particular fact or concept.
- Longer response items should be broken into smaller, more specific ones.

For example, consider the following two versions of the same question. The first version is not suitable for computerised marking, but the second modified version is:

Version 1

Explain the difference between passing a primitive data type and passing a reference data type as an argument in Java.

Version 2

- How do we pass primitive data type arguments to a method in Java?
- When a primitive data type argument is passed, will the changes made to the corresponding parameter be retained after the method returns?
- What happens in computer memory when a primitive data type argument is passed to a method?
- How do we pass reference data type arguments to a method in Java?
- When the reference data type is passed, what will the passed-in reference refer to once the method call has returned?
- What happens in computer memory when a reference data type argument is passed to a method?

The following sections describe the "state-of-the-art" in short-answer marking systems, the approach to marking that such systems follow and their capabilities and limitations. The two recently developed short-answer marking systems analysed in this paper are:

1. C-rater developed by Leacock and Chodorow (2003)

2. The **I**nformation **E**xtraction (IE) based system developed by Sukkarieh et al. (2003; 2004; 2005)

Research issues with respect to these two particular systems and those pertaining to the area in general are discussed.

3. C-rater

C-rater is an automated short-answer marking engine developed by **E**ducation **T**esting **S**ervice (ETS) (Leacock and Chodorow, 2003). It is designed to score factual answers and therefore the number of possible correct answers expected from students is finite. If we consider a set consisting of all the possible correct student responses, then the c-rater scoring engine operates as a *paraphrase recognizer* that identifies members of this set. For example, consider a question: *"Why is 26th January 2001 an important date for the Indian state of Gujarat?"* Some possible correct student responses are:

- *There was an earthquake in Gujarat on that day.*
- *Many people died in an earthquake.*
- *An earth quake occurred and many people died.*
- *Thousands of people were killed as a result of an earthquake.*

A possible incorrect response is: *There was false news of an earthquake in some parts of Gujarat that panicked people across the state.* Note that there are a few words such as *earthquake* and *people* common to both the correct and incorrect responses. The task of C-rater is to identify that the first four responses are paraphrases of the correct concept while the fifth one is not.

3.1 C-rater's approach to mark student responses

A model of the correct answer has to be created by a "content expert". C-rater's task is to map the student's response on to this model and, in so doing, check the correctness of the student's response. Before this mapping can take place, the student's response is first converted to a *canonical representation* (i.e. a non-ambiguous, mutually exclusive representation of "knowledge").

In order to generate canonical representations, the variations in the students' responses have to be normalised. The designers of C-rater have identified four primary sources of variations in students' answers: syntactic variations (e.g. *"The democrats dominate the US congress"* and *"The US congress is dominated by the democrats"*); pronoun reference (e.g. *"Alan bought the cake and ate **it**"*); morphological variations (e.g. *hide, hides, hided, hidden*) and the use of synonyms and similar words (e.g. *decrease, lessen, minimise*). Spelling and typographical errors are the fifth source of variation and even though it is not considered when studying paraphrases, C-rater needs to correct these errors itself for accurate marking to be possible. A brief overview of how C-rater handles these sources of variations is given below.

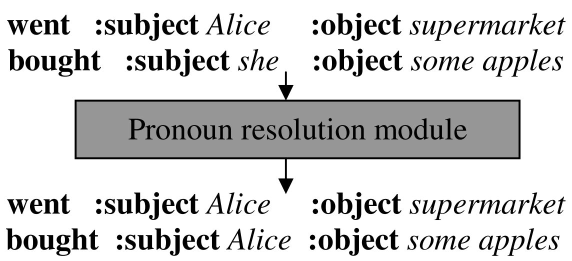
In content-based responses, the semantic domain is limited. If a student makes a typing or spelling error in their response, then that error can be automatically corrected because the correct word may easily be identified through the restricted domain. For example, consider the question: "*Why did Albert Einstein leave Germany and settle in the US in 1933?*" Now, suppose, if someone responds *"Abert Einsien"* instead of *"Albert Einstein"*, then C-rater automatically corrects the spelling.

Syntactic variation is the major source of paraphrasing. A canonical syntactic representation is created by c-rater which generates a predicate argument structure, or tuples, for each sentence of the student's response. A tuple consists of verb in each clause of a sentence together with its arguments (such as subject and object). For example, consider the question: *"What is the primary function of red blood cells in the human body?"* Table 1 below shows tuples for four possible responses to this question. The syntax of the three correct responses is different but their tuples are similar i.e. all three have "*Red blood cells*" as the subject of the main clause and "*oxygen*" as the object of main or sub-ordinate clauses. The wording of the fourth answer is similar to that of first three answers but it is marked incorrect because the object of this sentence is *"food"* rather than *"oxygen"*.

Score	Sentence and tuple
Credit	Red blood cells carry oxygen from lungs to body tissues through blood. **carry :subject** *Red blood cells* **:object** *oxygen*
Credit	Red blood cells travel through our body to deliver oxygen and remove waste. **travel :subject** *Red blood cells* **:object** *our body* **deliver :object** *oxygen* **remove :object** *waste*
Credit	Red blood cells have the important job of carrying oxygen. **have :subject** *Red blood cells* **:object** *important job* **carrying :object** *oxygen*
No credit	Red blood cells transports food to various parts of human body. **Transports :subject** *Red blood cells* **:object** *food*

Table 1. *Tuples for 4 responses.*

Pronoun resolution is the next important step. The pronoun resolution component of C-rater identifies all the noun phrases that precede the pronoun and all the noun phrases that are in the question. It then decides which noun phrase the pronoun refers to. For example, consider this sentence: "*Alice went to a supermarket where she bought some apples*". Consider next, below, the predicate-argument structure of this sentence before and after pronoun resolution:



went :subject *Alice* **:object** *supermarket*
bought :subject *she* **:object** *some apples*

Pronoun resolution module

went :subject *Alice* **:object** *supermarket*
bought :subject *Alice* **:object** *some apples*

Next, the morphological analysis component converts the inflected and derived forms of words to their base forms. For example, *adds*, *added*, *adding* and *addition* are all inflected and derived forms of the same base form *add*. Negated words are also converted to their base forms e.g. *illiterate* is converted to *literate*. But the meaning is retained as

not in the tuple. For example, consider the sentence: *"Peter is illiterate"*. Its predicate argument structure is:

be :subject *Peter* **:not :object** *literate*

The final step deals with the use of similar words and synonyms in student's responses as these words also need to be normalized. C-rater uses a *word similarity matrix* for this purpose (Leacock and Chodorow, 2003). The word similarity matrix has entries for a very large number of English words and with each word there is an associated list of similar word items. When a student's response is evaluated, C-rater tries to match each base form in the student's response with the base forms of the model answer and all the associated similar word lists. If a match is found, then the base form in the response is replaced with the word in the model answer.

Once a student's response has been converted to a normalised canonical representation, it is then compared with the canonical representation of the model answer. For each relation in the model answer's canonical representation, C-rater tries to find an equivalent relation in the canonical representation of the response.

3.2 C-rater evaluation

C-rater has been evaluated in two large-scale assessment programs (Leacock and Chodorow, 2003). The first was the **N**ational **A**ssessment of **E**ducational **P**rogress (NAEP) Math Online Project. C-rater was used to evaluate written explanations of the reasoning behind particular solutions to some Maths problems. Five such questions were used in the evaluation process. The second program was the online scoring and administration of Indiana's English 11 End of Course Assessment pilot study. In this case, C-rater was required to assess seven reading comprehension questions. The answers to these questions are more open-ended than those to the questions in NAEP Math Online Project. In the NAEP assessments, the average student response was around 15 words or 1.2 sentences long. Each student response was scored by C-rater and scored separately by two human judges. 250 to 300 student responses were used for each question. The agreement percentage between C-rater and the first human judge was 84.4% and between C-rater and the second human judge 83.6%. Flesis (1981) states that "Values greater than 0.75 or so may be taken to represent excellent agreement beyond chance". The overall agreement rate between C-rater and the two human judges was 0.84. This means that C-rater's performance was excellent in the case of the NAEP assessment.

In the Indiana pilot study, student responses were longer and the average length was around 2.8 sentences or 43 words. One hundred student responses were used for each question and were scored separately by C-rater and a human judge. Leacock and Chodorow (2003) summarized the evaluation results: "On average, C-rater and the human readers were in agreement 84% of the time". These results re-confirmed the excellent performance of C-rater according to standards set by Flesis (1981).

3.3 Limitations of C-rater and possible directions for future research

C-rater's errors fall into two categories: *misses* and *false positives*. A *miss* refers to C-rater's inability to recognise a correct concept in a response. This result in less credit awarded to the response. A *false positive,* on the other hand, occurs when a C-rater assigns too much credit for a response, i.e. credit is awarded for concept(s) that are not present in the response. The evaluation carried out by Leacock and Chodorow (2003) revealed that the rate of *misses* was much greater in the case of the Indiana assessment. The conclusion derived is that as the questions get more open-ended, the rate of *misses* increases. A possible direction of future research is to enable C-rater to effectively mark questions that are relatively more open-ended than the traditional close-ended, factual questions so that the rate of misses in such automated markings may be reduced.

False positives occur due to two reasons. C-rater has been designed to identify correct concepts in student responses but it can not detect invalid or wrong concepts. Sometimes a student expresses the required correct concept in the first sentence of the answer but then goes on to write something completely wrong or something that invalidates the correct part of the answer. C-rater awards mark on the correct part of the answer but does not deduct any mark for the wrong part. The second reason is that sometimes the student response contains the right language but it is not written in a manner that conveys the correct concept. This is mainly due to allowing for fragmentary and ungrammatical answers. Research needs to be carried out to resolve this problem of *false positives*.

4. The Information Extraction (IE) based system developed by Sukkarieh et al. (2003; 2004; 2005)

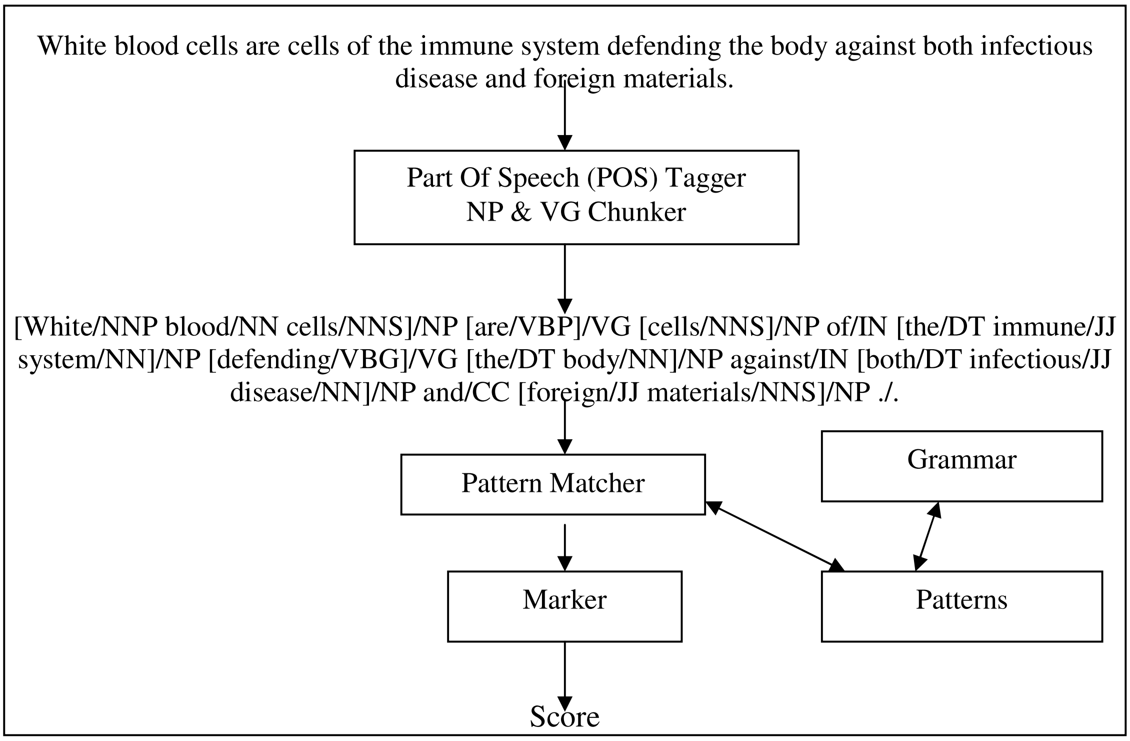
Many of the *University of Cambridge Local Examinations Syndicate (UCLES)*'s exam questions are short-answer questions which are worth one or two marks (Sukkarieh et al., 2003). Such questions are considered to be a useful and integral part of UCLES exams. Automated marking of short-answers is therefore desired by UCLES. An IE-based system is being developed at Oxford University to fulfill this need of UCLES. It is a UCLES funded project and work on this project began in summer 2002. The system's prototype has been evaluated using **G**eneral **C**ertificate of **S**econdary **E**ducation (GCSE) biology answers.

4.1 An overview of the I.E.-based system's design and performance

Information **E**xtraction (IE) techniques were adopted for use in the application. According to Sukkarieh et al. (2003), the reasons for this choice were that these techniques do not require complete and accurate parsing, they are relatively robust in the face of ungrammatical and incomplete sentences and they are also easy to implement. IE techniques are classified in to two categories: 'knowledge engineering' and 'machine learning'. The difference is that in the 'knowledge engineering' approach the information extraction patterns are discovered by a human expert while in the 'machine learning' approach the patterns are learned by the software itself.

The 'knowledge engineering' approach is more accurate and requires less training data but it requires considerable skill and time of a knowledge engineer (Appelt and Israel, 1999). On the other hand, the 'machine learning' approach is not as accurate as the 'knowledge engineering' approach. The 'machine learning' approach is suitable when no skilled knowledge engineer is available, training data is plentiful and the highest possible performance is not critical. First, the use of the 'knowledge engineering' approach in the system considered.

Figure 1 depicts the system's architecture. The student's answer is first subjected to shallow parsing by the **P**art **O**f **S**peech (POS) tagger and the **N**oun **P**hrase (NP) and **V**erb **G**roup (VG) chunker. The resulting tagged and chunked text is then used by the 'pattern matcher' which tries to match it with the hand-crafted patterns. The hand-crafted patterns must conform to the rules set out by the grammar. The result of the pattern matching process is fed to the 'marker' component which makes the final decision about the output.



White blood cells are cells of the immune system defending the body against both infectious disease and foreign materials.

[White/NNP blood/NN cells/NNS]/NP [are/VBP]/VG [cells/NNS]/NP of/IN [the/DT immune/JJ system/NN]/NP [defending/VBG]/VG [the/DT body/NN]/NP against/IN [both/DT infectious/JJ disease/NN]/NP and/CC [foreign/JJ materials/NNS]/NP ./.

Figure 1. *The system's architecture (Sukkarieh and Pulman, 2005).*

As already mentioned, a human expert discovers information extraction patterns in the 'knowledge engineering' approach. Appelt and Israel (1999) specified three crucial steps to accomplish the task of pattern writing by hand:

1. Determine all ways in which target information is expressed in a given corpus.
2. Determine all possible variants of these ways.
3. Write patterns of those ways.

Sukkarieh et al. (2003; 2004; 2005) abstracted patterns over three sets of data: (1) sample answers provided by examiners, (2) answers prepared by the evaluators themselves, and (3) student answers provided by UCLES. A pattern is basically various paraphrases collapsed into one (Sukkarieh and Pulman, 2005). It is important that pattern writers make use of the available linguistic features (i.e. the part-of-speech tags, the noun phrases and the verb groups). Consider the following example of abstracting information extraction pattern from sample answers of a question:

Question: *What is the function of white blood cells?*

Sample answers: 1. *Protect the body against disease.*
2. *Safeguard the body against infections.*
3. *Defend the body against both infectious disease and foreign materials.*
4. *Help human body fight against infections.*

Hand-written information extraction patterns:

```
{<protect>;<safeguard>;<defend>}+<body>+against+{<infection
>;<foreign material>;<virus>;<bacteria>;<disease>}
<protect>=Verb group with the content of ‘protect’
<safeguard>=Verb group with the content of ‘safeguard’ or
‘guard’
<defend>=Verb group with the content of ‘defend’
<body>={body, human body, organism}
<infection>=Noun phrase with the content of ‘infection’
<foreign material>=Noun phrase with the content of ‘foreign
material’
<virus>=Noun phrase with the content of ‘virus’
<bacteria>=Noun phrase with the content of ‘bacteria’
<disease>= Noun phrase with the content of ‘disease’
```

A set of patterns is associated with each question. This set is further divided into bags or equivalence classes. The members of an equivalence class are related by an equivalence relation i.e. a member of an equivalence class conveys the same message and/or information as other members of the same equivalence class. The marking algorithm compares student answers with equivalence classes and awards marks according to the number of matches.

The evaluation of the latest version of the system following the hand-crafted pattern writing approach was carried out using approximately 260 answers for each of the 9 questions taken from a UCLES GCSE biology exam. The full mark for these questions ranged from 1 to 4. 200 marked answers were used as the training set (i.e. the patterns were abstracted over these answers) and 60 unmarked answers were kept for the testing phase. The average percentage agreement between the system and the marks assigned by a human examiner was 84% (Sukkarieh and Pulman, 2005).

The amount of work involved in pattern writing is significant. Human expertise in both the computational linguistic and the domain of the examination are also required. Automatic customisation to new questions is therefore desirable to remove these requirements. Machine learning methods provide ways in which a short-answer marking system can be automatically customized to new questions using a training set of marked answers. A number of machine learning techniques have been tried in the system and their evaluated performances are reported by Sukkarieh et al. (2003; 2005). The machine learning techniques that have been tried are: Nearest Neighbor classification, **I**nductive **L**ogic **P**rogramming (ILP), **D**ecision **T**ree **L**earning (DTL) and **N**aïve **B**ayesian learning (NBayes). The description of these techniques is beyond the scope of this paper.

The evaluation results of the application of machine learning techniques in the short answer marking system shows that while these techniques are promising, they are not accurate enough at present to replace the hand-crafted pattern matching approach (Sukkarieh and Pulman, 2005). Currently, such techniques should be used to either aid pattern writing (Sukkarieh and Pulman, 2005) or perhaps act as complementary assessment techniques for extra confirmation.

4.2 The I.E.-based system's limitations and possible directions for future research

Existing systems' performance is unsatisfactory in cases where the required degree of inference is beyond the state-of-the-art (Sukkarieh and Pulman, 2005). The following are some situations where this may occur:

1. Need for reasoning and making inferences: for example, a student may answer with "*keep us healthy*" rather than *"protect the body from diseases"*.
2. Students sometimes use negation of a negation: for example, the answer *"it is not necessary for a female cat to give birth at a specific time"* is equal to *"a female cat can give birth at any time"*.
3. Contradictory or inconsistent information: an example of contradictory information is *needs photosynthesis/does not need photosynthesis*. An example of inconsistent information is *"red blood cells carry oxygen to different parts of human body but each human cell has a DNA molecule"*.

In order to enable systems to deal with higher levels of inference, more sophisticated techniques need to be devised. This need provides a possible direction for future research.

Currently, systems only provide summative feedback. Systems should be extended to provide useful, formative feedback. This is another possible direction for future research. There may be many ways of developing this functionality into systems; Sukkarieh et al. (2003) describe one possible approach. First a group of teachers examine a sample of student answers. A possible source of these sample student answers may be past-paper answer scripts (provided the question has appeared in a past examination). The teachers sort these sample student answers into feedback categories according to their semantic content. The number of feedback categories should not be too small; otherwise the feedback provided to the student will not be specific enough to be useful. The number of feedback categories would normally be much less than the number of student answers because many student answers would only be superficially different and would have common mistakes or misconceptions. The teachers would then write appropriate feedback for each category. The answer submitted to the system would be matched with the sample answers. The group of sample answers closest to the input would then be identified and the feedback category into which the greatest number of the group members fell would form the basis of the formative output.

5. Summary and Conclusions

Significant progress in automating the marking of short free-text responses has been made possible by developments in both hardware (specifically the availability of ever-more powerful personal computers), and also in software (often, but not exclusively, more expressive programming languages & techniques for their effective use), and latterly by new techniques (typically algorithms) for computing the accuracy (or otherwise) of an answer (typically to a question which conforms to a specific structure).

The utility of systems that are capable of marking short free-text responses accurately are intuitively obvious, i.e. the drudgery of reading and assessing possibly very large numbers of answers to questions is removed, and there is the potential for cost savings (but there are inherent overheads), greater accuracy, and the reduction, even the elimination of, bias.

Unfortunately, proprietary considerations (typically the need to keep secret the algorithms, processing techniques etc, used in commercial products), have prevented the developers of such systems from placing their research and development in the public domain. Where such work has been made available, data used in comparisons has often been different and hence the comparison is hardly effective.

A concerted effort is required to develop standardised repositories of questions and answers suitable for systematic testing of such systems, and especially new systems based on novel techniques, in order that effective comparisons and evaluations of such systems are possible. Ideally, the results of such tests should be made public, and tables produced that give detailed performance analyses.

In such a situation, the specific capabilities and limitations of systems for marking short free-text responses would be available to prospective users in advance, and users would have reasonably accurate, possibly independently audited, figures detailing the performance of systems that have been tested.

For researchers in the area, such performance figures would provide a basis for determining which capabilities need to be enhanced, and for benchmarking new systems with existing systems to determine if improvements have been made, and if so, by what amount.

References

Appelt, D. and Israel, D. (1999) *Introduction to information extraction technology.* IJCAI Tutorial.

Fleiss, J. L. (1981) *Statistical Methods for Rates and Proportions.* New York: John Wiley & Sons, pp. 212-236.

Hearst, M. A. (2000) The debate on automated essay grading. *IEEE Intelligent Systems*, September/October 2000, pp. 22-27.

Leacock, C. and Chodorow, M. (2003) C-rater: Automated Scoring of Short-Answer Questions. *Computers and the Humanities*, 37(4), pp. 389-405.

Sukkarieh, J. Z., Pulman, S. G. and Raikes, N. (2003) *Auto-marking: using computational linguistics to score short, free text responses.* Paper presented at the 29th annual conference of the International Association for Educational Assessment (IAEA), Manchester, UK.

Sukkarieh, J. Z., Pulman, S. G. and Raikes, N. (2004) *Auto-marking 2: An update on the UCLES-Oxford University research into using computational linguistics to score short, free text responses.* Paper presented at the 30th annual conference of the International Association for Educational Assessment (IAEA), Philadelphia, USA.

Sukkarieh, J. Z. and Pulman, S. G. (2005) Automatic Short Answer Marking. In: *Proceedings of the 2nd Workshop on Building Educational Applications Using NLP, June 2005*. Association for Computational Linguistics, pp. 9-16.