c-rater:Automatic Content Scoring for Short Constructed Responses

Jana Z. Sukkarieh and **John Blackmore**
Educational Testing Service
Princeton
NJ 08541

Abstract

The education community is moving towards constructed or free-text responses and computer-based assessment. At the same time, progress in natural language processing and knowledge representation has made it possible to consider free-text or constructed responses without having to fully understand the text. c-rater is a technology at Educational Testing Service (ETS) used for automatic content scoring for short, free-text responses. This paper describes some of the major developments made in c-rater recently.

Introduction

The education community is moving towards constructed or free-text responses[1]. Also, it is moving towards widespread computer-based assessments. At the same time, progress in natural language processing (NLP) and knowledge representation (KR) has made it possible to consider free-text responses without having to fully understand the text. c-rater (Leacock & Chodorow 2003) is a technology at Educational Testing Service (ETS) for automatic content scoring for short, free-text responses. This paper describes some of the major developments made in c-rater recently. Unlike most automatic content scoring systems, c-rater considers analytic-based content. This means that a c-rater item consists of (in addition to a prompt and an optional stimulus) a set of clear, distinct, predictable, main/key points or concepts, and the aim is to score students' answers automatically for evidence of what a student knows vis-à-vis these concepts. See items in Table 1 for examples. For each item, there are corresponding concepts in the right-hand column that are denoted by C_1, C_2, ... C_n where n is a concept number. These are also separated by semicolons for additional clarity. The number of concepts, N, is included in the heading **Concepts:**N. The scoring guide for each item is based on those concepts. Note that we deal with items whose answers range from few words up to around 100 words each.
In the section "c-rater in a Nutshell", we describe c-rater's task in terms of NLP and KR, and how c-rater works, that is, 'the solution' we undertake for that task. In the section "How it Works", we describe each part of 'the solution' in more detail and emphasize the recent major enhancements. In the section "Evaluation", we describe a pilot study we conducted in 2008 and the results on existing ETS items. We then discuss some of our limitations and consequentially the need to introduce deeper semantics and an inference engine into c-rater. Before we conclude, we briefly summarize others' work on automatic content scoring.

Table 1: Sample items in Biology and Reading Comprehension

Statement of the item	**Rubric**
Item 1. Full Credit is 2 Identify TWO common ways the body maintains homeostasis during exercise	**Concepts:3** C_1: sweating; C_2: dilation of blood vessels; C_3: increased circulation rate;
Scoring Guide	2 points for 2 or more concepts 1 point for 1 concept else 0
Item 2. Full Credit is 1 According to the text (a text has been given), what was one SIMILARITY between robes with abstract designs and robes with life scenes?	**Concepts:3** C_1: They both used bulky hides; C_2: They were both flatly painted; C_3: They were both painted by Plains Indians;
Scoring Guide	1 point for 1 concept
Item 3. Full credit is 2 (a stimulus is given) Explain what you think the delegates may be trying to persuade the Native Americans to believe or to do. Then, name an example of how the delegates attempt to persuade through their speech.	**Concepts: 11** C_1: to understand the conflict with England; C_2: to take their side against England; C_3: to appeal to the Native Americans; C_4: by calling them as brothers; C_5: use authoritative language; 6 other concepts
Scoring Guide	1 point for a 'what' concept 1 point for a 'how' concept

c-rater in a Nutshell

c-rater's Task

We view c-rater's task as a textual entailment problem (TE). We use TE here to mean either:

- a paraphrase
- an inference up to a context

For example, consider item 3 in Table 1. An answer like *"take the colonists' side against England"* is the same as C2, an answer like *"the dispute with England is understood"* is a paraphrase of C1 and an answer like *"The colonists address the crowd. They say Oh Siblings!"* implies C4. Note that in this case the word siblings is acceptable while an answer like *"My sibling has a Y Chromosome"* for the concept *"My brother has a Y chromosome"* is not

[1]We use the term responses and answers interchangeably.

acceptable. The context of the item is essential in some cases in determining whether an answer is acceptable; hence, we say **up to a context** in the definition above.

c-rater's task is reduced to a TE problem in the following way:

> **Given:** a concept, C, (for example, *"body increases its temperature"*) **and** a student answer[2], A, (for example either *"the body raise tempreture"*, *"the bdy responded. His temperature was* 37° *and now it is* 38°*"* or *"Max has a fever"*) **and** the context of the item,
> **the aim is** to check whether C is an inference or a paraphrase of A (in other words A implies C and A is true)

How it Works

Given a c-rater item (a test question) and a sample of student data, there are four major processes in c-rater.

1. **Model building:** This simply means that given the concepts in the item, one or more model answers for this item are created. A model answer is a set of model sentences where each sentence corresponds to a certain concept. This correspondence means that the model sentence entails the concept. The process of model building consists mainly of 3 tasks:
   (a) Generating model sentences i.e., sentences that are paraphrases or imply a concept. This is done for each concept.
   (b) Selecting required or essential words (or multi-word lexicon) in the above sentences which is nothing but eliminating stop words and irrelevant or noisy lexicon.
   (c) Selecting similar lexicon to the above required words from various resources like Rogets Thesaurus, Dekang Lin's similar lexicon databases (http://www.cs.ualberta.ca/ lindek/downloads.htm), and WordNet.

   For example, Table 2 shows a set of model answers for item 1 above. For each sentence, the required lexicon (RL) are given in bold and their similar lexicon are denoted by $Sim(RL)$. For the results in this paper, the process of model building is done in a knowledge-engineering way (i.e., a human performs the above tasks). We have since automated the process. However, this will be left for another paper.
2. **Linguistic processing:** Model answers and students' answers are automatically processed using the same Natural language processing (NLP) tools.
3. **Recognizing the main points or concepts:** The linguistic features obtained from the previous step are used to automatically determine whether the response contains the concepts expected in a student answer or not.
4. **Scoring:** Finally, based on 3), scoring guidelines are applied accordingly to produce a score. In addition to a score, recently we enhanced the output to obtain concept-based feedback and a confidence measure. The feedback states which concept(s) the student got right; hence, it gives a transparent justification for the score. The confidence measure is c-rater's way of self-assessment. When c-rater is not confident about its score for a certain answer it flags the answer for a human for further review.

[2]This may contain misspellings and grammatical errors.

Table 2: A Set of Model Answers for Item 1 in Table 1

Concept 1: sweating	
	Model sentence 1: **sweating**
	Sim(sweat): {perspire}
	Model sentence 2: to **release moisture**
	Sim(release): {discharge}
	Sim(moisture): {water, wetness}
	Model sentence 3: **exuding droplets** while moving
	no similar lexicon chosen
Concept 2: dilation of blood vessels	
	Model sentence 1: **dilation** of **blood vessels**
	no similar lexicon chosen
Concept 3: increased circulation rate	
	Model sentence 1: **increased circulation rate**
	Sim(increase): {raise, rise, augment }
	Model sentence 2: **rate** of **blood flow increases**
	same similar lexicon chosen for increase

Model Building

In the past, c-rater's model building process depended on holistic scores provided by two humans. Recently, we modified the process to depend on concept or analytic scores. To this end, among other things, we defined what we call concept-based scores, designed and implemented a concept-based scoring or annotation tool, and modified the statistical analysis we perform in terms of kappa calculation measuring agreement between the two humans, and agreement between each human and c-rater. The motivation behind concept-based scoring has many aspects. First, trying to have a one-to-one correspondence between human analytic annotations (will be described next) and a concept will minimize the noise in the data; hence, increase the accuracy of a model. This should simplify automating model building which is currently laborious and time-consuming, when done in a knowledge-engineering approach. Further, we expect better accuracy with which the matching algorithm decides about whether Concept C is a TE of Answer A since it is learning from a much more accurate set of linguistic features about the TE task than it does without this correspondence. A similar idea has been used in the OXFORD-UCLES system (Sukkarieh & Pulman 2005) where a Naïve Bayes learning algorithm applied only to the lexicon in the answers produced a high-quality model from a tighter correspondence between a concept and the portion of the answer that deserves a credit.

Concept-Based Scoring Given a scoring form containing the students' answers, we ask the human raters to annotate the answers of the items. By annotation, we mean that for each concept, we ask the raters to quote the portion from a student's answer that says the same thing as, or implies, the concept in the context of the question at hand. For example, assume a student answers Item 1 with *This is an easy process. The body maintains homeostasis during exercise by releasing water and usually by increasing blood flow*. For C_1: sweating, the human rater quotes *releasing water*. For

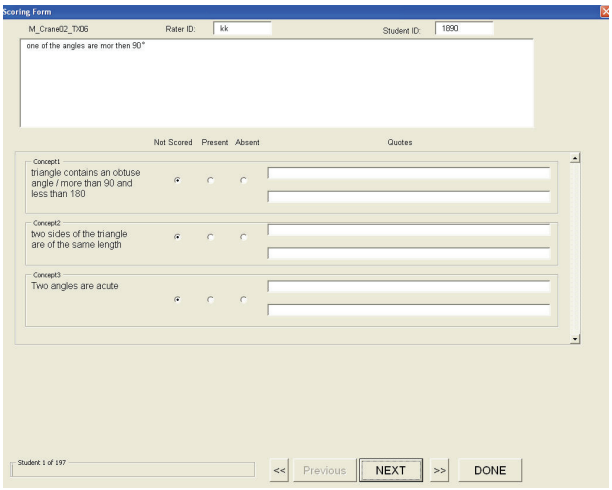

Figure 1: Concept-Based Scoring or Annotation Tool

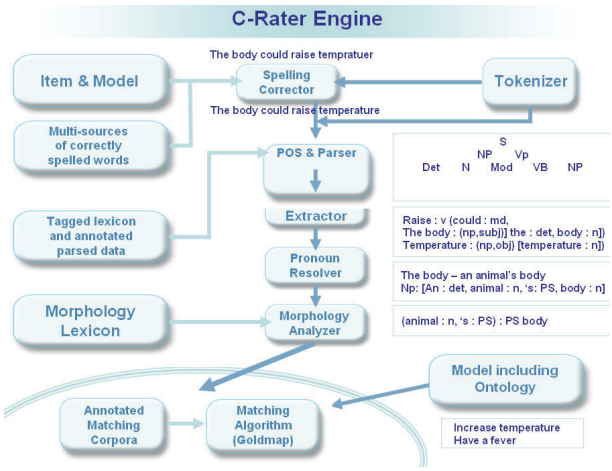

Figure 2: c-rater's Engine: Linguistic Analysis and Matching Module

C_3: increased circulation rate, the rater quotes *increasing blood flow*.

For every item, a scoring form was built. An example for a mathematics item with 3 concepts is shown in Figure 1.

The concepts corresponding to the item were listed in the form. For each answer, the human rater clicks on *Absent* when a concept is absent, *Present* when a concept is present or *Negated* when a concept is negated or refuted[3]. This is done for each concept. The default is *Not scored*. $Absent, Present, Negated$ are what we call **analytic** or **concept-based scores** and not the actual scores according to the scoring rules. When a concept is present or negated, the raters are asked to include a quote extracted from the student's answer to indicate the existence or the negation of the concept. Basically, the raters are asked to extract the portion of the Text T that is a paraphrase or implies the Concept, C, (when the concept is present) and the portion of Text T such that $T = neg(C)$ (when the concept is negated). We call a quote corresponding to Concept C **positive evidence** or **negative evidence** for $Present$ and $Negated$, respectively. Note that portions corresponding to one piece of **evidence** (positive or negative) do not need to be in the same sentence and could be scattered over a few lines. Sometimes there is more than one piece of **evidence** for a particular concept. Further, due to the nature of the task some cases are subjective (no matter how objective the concepts are, deciding about an implication in a context is sometimes tricky). Hence, the annotation of some of the answers of some items may be challenging.

Linguistic analysis: c-rater and NLP

The major modification in the linguistic processing was to replace a partial parser by a deeper parser with a constituent-based tree output, and enhance and enrich the linguistic features that are now extracted from the constituent-based parse tree using rules that we have written ourselves using what we call feature extractor (the original module was called chunker). In the following, we describe the linguistic analysis in more detail.

Figure 2 contains the architecture of the linguistic analysis and the matching module, Goldmap, in c-rater's application. Student data is noisy; that is, it is full of misspellings and grammatical mistakes. Any NLP tool that we depend on should be robust enough towards noise. In the following, we describe the stages that a student answer and a model answer go through in terms of processing in c-rater. Spelling correction is performed as a first step in an attempt to decrease the noise for subsequent NLP tools.

In the next stage, parts of speech tagging and parsing are performed. c-rater used to utilize a partial parser, Cass, (Robert Berwick & Tenny 1991), which uses a chunk-and-clause parsing approach where ambiguity is contained; e.g. a prepositional phrase (PP) attachment is left unattached when it is ambiguous. Cass has been designed for large amounts of noisy text. However, we observed that the degree of noise varies from one set of data to another (in our space of data) and a large subset of our data is not as noisy as it was originally thought (e.g. we noted the difference between text written by native and non-native English speakers). Hence, in an attempt to gain additional linguistic features, a deeper parser was introduced (OpenNLP parser, Baldridge and Morton, http://opennlp.sourceforge.net/) instead of Cass.

In the third stage, a parse tree is passed to a feature extractor. We apply rules that we have written ourselves to extract features from the parse tree. The result is a flat structure representing phrases, predicates, and relations between predicates and entities. Each phrase is annotated with a label indicating whether it is independent or dependent. Each entity is annotated with some syntactic and semantic role. Until now, we have used around 25 different labels with various granularity that include among others { *verb*, *complement*, *object*, *indirect_obj*, *agent*, *passive_verb*, *subject*, *object_ of_a_preposition*, *negation*, *dependent_clause*,

[3] In the example shown, we did not include the $Negated$ option.

Table 3: Goldmap Features or Functions

Feature	Description
nmw	Number of missing words, refers to the number of required words that were not matched by the response sentence.
argsMismatch	Required argument does not match.
argRoleIncompatible	Found term(s) of required argument, but the type of matching role is incompatible with type of required argument role.
polarityMismatch	Required role and matching role do not agree on negation; either required role is negated and matching role is not, or vice-versa.
vpPOSMismatch	To identify cases where we match adjectives to VBN_verbs and vice-versa.

independent_clause, *relative_clause*, *subordinate_clause*, *modal_verb*, *ergative_verb*, *phrasal_verb*, *ditransitive_verb* ...}. The structure also indicates the links between various clauses and distributes links when necessary. For example, if there is a conjunction with a distributive verb like *Eleanor cleaned the first module and the second module*, a link is established between the two conjuncts and the verb is distributed to represent *Eleanor cleaned the first module and Eleanor cleaned the second module*.

The next stage is an attempt to resolve pronouns to either an entity in the text of an answer or the question. Finally, a morphological analyzer reduces words to their lemmas. We assume that the reader knows what these last two stages do and will not go into any details here.

The culmination of all the above tools reside in one representation including all the above linguistic features waiting to be used by the matching algorithm, Goldmap, that we describe next.

Goldmap or Concept Detection

In the past, c-rater's matching algorithm, Goldmap, was a rule-based pattern-matching algorithm giving a 0/1 match. Though rule-based approaches are more transparent and easier to track, they are not flexible. Any amount of "uncertainty" (which is the case when extracting linguistic features from a text; let alone noisy text) will always imply failure on the match. A probabilistic learning approach, on the other hand, is "flexible". That said, a probabilistic learning approach is not as transparent, and it lends itself to the usual questions about which threshold, in the space of probabilities, should be considered and whether heuristics should be used i.e. should a probability above 0.5 qualify for a match or another value? We will not go further into this here. For the study in this paper, we use a probability greater or equal to 0.5 to decide a match.

Maximum Entropy Modeling for Concept Detection Maximum entropy (ME) modeling is a probability distribution estimation technique with a closed world principle i.e., the technique models **what is known** and assumes nothing else. **What is unknown** is modeled as if ME is a uniform distribution (Ratnaparkhi 2003). Maximizing the uncertainty, randomness or entropy[4] ensures the least-biased distribution. This also makes the algorithm resistant to noise (Goldwater & Johnson 2003). **What is known** is usually a set of observations or facts that one makes about a sample of the data at hand (or training data) and writes these observations in terms of constraints (or functions/features with constraints over their values). The constraints may have a ranking order too.

In c-rater, ME is used to output a probability on whether some sentence in a student's answer is a paraphrase or implies a concept. The training data from which Goldmap learns its model consists of: a 1000 sentence pairs which are item-independent, with a label of 0 or 1 for each pair for {no_match, match} respectively (manually labelled). In the following, we list some of the observations that the constraints we define in Goldmap take into consideration. Before that we just explain some notation. Let $Sim(X)$ denote the set of similar lexicon of a lexical entity denoted by X and let P^{-1} denote the passive voice of a predicate denoted by P :

1. Two sentences with no common required lexicon or similar lexicon are unlikely to match[5]
2. A sequence of lexicon including morphology in one sentence matches the exact sequence of lexicon including morphology in another sentence
3. A predicate, P, with subject S and object O in one sentence matches a predicate, P, or one of its similar lexicon $Sim(P)$ with a subject S or S' where $S' \in Sim(S)$ and an object O or O' where $O' \in Sim(O)$
4. A predicate, P, with subject S and object O in one sentence matches P^{-1} with subject O or O' where $O' \in Sim(O)$ and object S or $S' \in Sim(S)$
5. A negated role does not match a positive role
6. A past participle (VBN_verb) could be used as an adjective; hence, its similar lexicon could be adjectives
7. Complement of an auxiliary (a noun phrase, an adjective or a prepositional phrase) could be replaced with another complement
8. Ergative verbs need special rules: when ergative verbs have a subject but no object, we consider the subject as the object in our matching. For example, "The pollution decreased the fish populations" and "The fish populations decreased"
9. A relative pronoun could be replaced with its corresponding role in the independent clause that the relative clause depends on
10. Students may write an interrogative utterance for a statement and still be considered correct

The observations are data-driven, i.e., observed in our students' data. In addition to real students' data, we are also using the syntactic-based variations described in (Vanderwende & Dolan 2006) for some guidance. See Table 3 for some of the features or functions that we use to represent these observations. To sum up, Goldmap's features and the constraints applied on their values depend on the linguistic features that are obtained in the linguistic analysis in the

[4] hence, the name.

[5] Note that it is not impossible that a similar lexicon is replaced by a gloss, a definition or an inference.

Table 4: An Example Showing a Student's Answer Compared to a Model Sentence.

MODEL SENTENCE: Peter threw the ball.
REQUIRED WORDS: ball, peter, threw (no similar words for any)
MODEL SENTENCE AFTER PRE-PROCESSING: Peter threw the ball.
MODEL SENTENCE PARSE:
(TOP (S (NP (NNP Peter))(VP (VBD threw)
(NP (DT the) (NN ball)))(. .)))
MODEL SENTENCE LINGUISTIC ANALYSIS OUTPUT:
Independent_clause throw :subj peter :obj ball
STUDENT'S RESPONSE: Peter wis thrown by theball.
RESPONSE AFTER PRE-PROCESSING: Peter was thrown by the ball.
RESPONSE PARSE:
(TOP (S (NP (NNP Peter))
(VP (VBD was)(VP (VBN thrown)
(PP (IN by)(NP (DT the) (NN ball)))))(. .)))
RESPONSE LINGUISTIC ANALYSIS OUTPUT:
Independent_clause be throw :psubj peter :by :pagent ball
GOLDMAP FEATURES:
<PROBABILITY: 0.3685>
nmw=0
argsMismatch:subj
argRoleIncompatible:$subj \longrightarrow psubj$
argsMismatch:obj
argRoleIncompatible:$obj \longrightarrow pagent$

previous process, the required lexicon and their similar lexicon. When a student's answer is given, each sentence in that answer is compared to each model sentence under one concept. A probability on that match is obtained for each sentence pair $< Model\_sentence, Answer\_sentence >$ for each concept. The highest probability is considered. An example is given in Table 4. In the example, one can see how the value of $argRoleIncompatible$ feature captures the observation we made about passives earlier.

Evaluation

The study that we conducted can be summarized as follows. We considered 12 questions: 7 reading comprehension questions and 5 mathematics questions whose answers are textual short answers written in English. The examinees were 7th and 8th graders in Maine. The sample size of data that was used to build a model ranged from 130-150 answers. Two human raters were asked to annotate and score the data. For each item, once a concept-based model was built, the unseen or blind data was scored. The size of the blind data ranged from 61-114 answers. Table 5 shows the results, in terms of unweighted kappa and percentage agreement showing agreement between the two humans (H1-H2), and the average agreements between c-rater and each human (c-H1/H2). The results are very promising.

The reasons we observed for the failure of a match (and consequently a lower agreement) varied from:

- Some concepts were not distinct. For example, in one mathematics item there were 7 concepts that should have been collapsed into 3 e.g., a concept *Crane 1 is like Crane 2 just smaller* is the same as another concept *Crane 2 is the same as Crane 1 just bigger*.
- Uncorrected spelling mistakes (or sometimes corrected to an unintended word)
- Unexpected similar lexicon, unexpected variations that a model did not predict
- A similar lexicon is not enough for e.g. a definition or a gloss of a word is needed instead of a synonym

Table 5: Concept-Based Scoring Results

Item	# Training (Blind)	H1-H2	c-H1/H2
R02	150 (114)	1.0 (100%)	0.94 (98%)
R08	150 (113)	0.76 (91)	0.69 (88)
R12B	150 (107)	0.96 (98)	0.87 (92)
R21A	150 (66)	0.77 (84)	0.71 (80.5)
RU05	130 (60)	0.71 (81)	0.58 (75)
RU19	130 (61)	0.71 (83)	0.73 (83.5)
RU27	130 (61)	0.87 (91)	0.55 (69)
M02B	130 (67)	0.71 (89)	0.6 (84)
M02D	130 (67)	0.8 (91)	0.71 (86)
M02F	130 (67)	0.86 (94)	0.76 (89)
M03	130 (67)	0.87 (95)	0.82 (93)
M05	130 (67)	0.77 (87)	0.63 (80)

- Linguistic phenomena that we do not deal with
- The need for a reasoning/inference module
- The fact that some model sentences are too general and have generated false positives (negative evidence was used as guidance to minimize this occurence)
- Inconsistency in the concept-based scoring due to either an error on behalf of a human rater or the fact that for some answers of some items, annotation was challenging

Semantics and Inferences

We stated above that we consider the problem to be a TE problem, and this will require extracting more semantics as well as the use of world knowledge. Until now, we have depended on syntax and lexical semantics (mainly similar lexicon) and simple semantic roles. Even with lexical semantics, we need to include many more enhancements. Sentences like *the British prevented them from owning lands* will not match *not owning land* unless the implicit negation in the word *prevent* is stated clearly. In addition to semantics and world knowledge, what distinguishes the task of automatic content scoring from other textual entailment tasks is that the context of the item needs to be taken into consideration. Also, students' answers are full of misspellings and grammatical mistakes that, as far as we know, no other textual entailment system has dealt with so far.

Further, one main limitation in Goldmap is that it is trained with sentence pairs and not answer-concept pairs. It currently not only favours poorly written long sentences over short discrete sentences, but may miss the entailment if it is over more than one sentence. Finally, we will always face the challenge of detecting negative evidence that students include in their answers unless that negated evidence is pointed out, in advance, by test developers or human raters.

Automatic Content Scoring:Others' work

In the last few years, a keen interest in automatic content scoring of constructed-response items has emerged. Several systems for content scoring exist. We name a few, namely, TCT (Larkey 1998), SEAR (Christie 1999), Intelligent Essay Assessor (Foltz, Laham, & Landauer 2003), IEMS (Ming, Mikhailov, & Kuan 2000), Automark (Mitchell *et al.* 2002) , C-rater (Leacock & Chodorow 2003), OXFORD-UCLES (Sukkarieh, Pulman, & Raikes 2003), Carmel (Rosé *et al.* 2003), JESS (Ishioka & Kameda 2004), etc. The techniques used vary from latent semantic analysis (LSA)

or any variant of it, to data mining, text clustering, information extraction (IE), BLEU algorithm or a hybrid of any of the above. The languages dealt with in such systems are English, Spanish, Japanese, German, Finnish, Hebrew, or French. However, the only four systems that deal with both **short answers** and **analytic-based content** are Automark at Intelligent Assessment Technologies , c-rater at Educational Testing Service(ETS), the Oxford-UCLES system at the University of Oxford and CarmelTC at Carnegie Mellon University, all dealing with answers written in English. Though Automark, c-rater and OXFORD-UCLES were developed independently, their first versions worked very similarly using a sort of knowledge-engineering information extraction approach taking advantage of shallow linguistic features that ensure robustness against noisy data (i.e., misspellings and grammatical errors). Later on, OXFORD-UCLES used data mining techniques similar to the ones in CarmelTC. Though these latter techniques prove very promising in categorizing students' answers into classes (a class is either a main point expected in an answer or "none"), the models of most of these techniques are not transparent, an issue that researchers who use data mining techniques for educational purposes need to address.
Unfortunately, there is no evaluation benchmark to compare results between c-rater, Automark, Carmel and OXFORD-UCLES. We would like to develop a benchmark set since we believe that this will contribute to and help automatic content scoring research but IP issues on items and their answers currently prevent us from doing so.

Conclusion

In this paper, we have described c-rater, ETS's technology for automatic content scoring of short constructed responses. We have also reported results for a pilot study with the recent version of c-rater. The results are promising, but more work needs to be done. In the near future, we will be concentrating on improving and adding tools that will help us obtain additional **linguistic features** in order to perform a more informed TE task. In particular, more than one parsing mechanism is to be included. On the one hand, we would like to take advantage of output obtained from dependency-tree parsing and on the other one parsing mechanism could be used as a fallback strategy to the other (when deeper-parsing results are deemed unreliable) and potentially a semantic representation will be added to the output of the parser.

Currently, there are many linguistic phenomena we cannot deal with and do not include in our observations for Goldmap, work will continue in that direction to increase Goldmap's accuracy. In addition, enlarging the set of training data from which Goldmap learns its model is underway. Finally, we are conducting a more thorough evaluation for some of c-rater's modules and categorizing, linguistically, what we can and cannot do. Using these linguistic categories, we will better determine the benefits and tradeoffs of the modifications we have made recently.

Acknowledgments

Special thanks to Tom Morton.

References

Christie, J. 1999. Automated essay marking for both content and style. In *Proceedings of the 3rd International Computer Assisted Assessment Conference*.

Foltz, P.; Laham, D.; and Landauer, T. 2003. Automated essay scoring. In *Applications to educational technology*.

Goldwater, S., and Johnson, M. 2003. Learning ot constraint rankings using a maximum entropy model. In *Stockholm Workshop on Variation within Optimality Theory*.

Ishioka, T., and Kameda, M. 2004. Automated japanese essay scoring system: Jess. In *In Proceedings of the 15th International Workshop on database and Expert Systems applications*.

Larkey, L. 1998. Automatic essay grading using text categorization techniques. In *In proceedings of the 21st annual international ACM SIGIR Conference on Research and Development in Information Retrieval*.

Leacock, C., and Chodorow, M. 2003. C-rater: Automated scoring of short-answer questions. *Computers and Humanities* 389–405.

Ming, Y.; Mikhailov, A.; and Kuan, T. L. 2000. Intelligent essay marking system. Technical report, Learner Together NgeANN Polytechnic, Singapore.

Mitchell, T.; Russel, T.; Broomhead, P.; and Aldrige, N. 2002. Towards robust computerised marking of free-text responses. In *In Proceedings of the 6th International Computer Assisted Assessment Conference*.

Ratnaparkhi, A. 2003. A simple introduction to maximum entropy models for natural language processing. Technical report, University of Pennsylvania.

Robert Berwick, S. A., and Tenny, C., eds. 1991. *Principle-Based Parsing*. Kluwer Academic Publishers, Dordrecht. chapter Parsing by Chunks, 257–278.

Rosé, C. P.; Roque, A.; Bhembe, D.; and VanLehn, K. 2003. A hybrid text classification approach for analysis of student essays. In *Building Educational Applications Using NLP*.

Sukkarieh, J. Z., and Pulman, S. G. 2005. Information extraction and machine learning: Auto-marking short free text responses to science questions. In *Proceedings of the 12th International Conference on AI in Education*.

Sukkarieh, J. Z.; Pulman, S. G.; and Raikes, N. 2003. Auto-marking: using computational linguistics to score short, free text responses. In *Presented at the 29th IAEA*.

Vanderwende, L., and Dolan, W. B. 2006. What syntax can contribute in the entailment task. In *MLCM 2005, LNAI 3944, pp. 205-216. J. Quinonero-Candela et al. (eds.). Springer-Verlag.*