A reliable approach to automatic assessment of short answer free responses

Lyle F Bachman, Nathan Carr, Greg Kamei, Mikyung Kim, Michael J Pan, Chris Salvador, Yasuyo Sawaki
WebLAS, Applied Linguistics & TESL, UCLA
Los Angeles, CA 90095
{bachman, carr, kamei, kimmi, mjpan, chriss, ysawaki}@WebLAS.ucla.edu

Abstract

This paper discusses an innovative approach to the computer assisted scoring of student responses in WebLAS (web-based language assessment system)- a language assessment system delivered entirely over the web. Expected student responses are limited production free response questions.

The portions of WebLAS with which we are concerned are the task creation and scoring modules. Within the task creation module, instructors and language experts do not only provide the task input and prompt. More importantly, they interactively inform the system how and how much to score student responses. This interaction consists of WebLAS' natural language processing (NLP) modules searching for alternatives of the provided "gold standard" (Hirschman et al, 2000) answer and asking for confirmation of score assignment. WebLAS processes and stores all this information within its database, to be used in the task delivery and scoring phases.

1 Introduction

Most assessment for placement, diagnosis, progress and achievement in our language programs are presently administered in paper and pencil (P&P) format. This format carries a number of administrative costs and inefficiencies. It requires new hard copy forms of assessments for each course and class, incurring costs associated with copying, handling, distributing, and collecting test booklets and answer sheets to test takers. Although some of the assessments can be scored by machine, teachers score those with free responses, such as open-ended questions and cloze (gap filling) tests.

WebLAS addresses the problems of a P&P format. It provides an integrated approach to assessing language ability for the purpose of making decisions about placement, diagnosis, progress and achievement in the East Asian language programs, as the content specifications of the assessment system for these languages are based directly on the course content, as specified in scope and sequence charts, and utilize tasks that are similar to those used in classroom instruction. WebLAS is thus being designed with the following expected advantages as objectives:

1. Greater administrative efficiency
2. More authentic, interactive and valid assessments of language ability such as integration of course and assessment content and incorporation of cutting edge and multimedia technology for assessment

Nested within these objectives is the ability to automatically assess limited production free responses. Existing systems such as e-Rater (Burstein et al) focus on holistic essay scoring. Even so, systems such as PEG (Page 1966) disregard content and simply perform surface feature analysis, such as a tabulation of syntactical usage. Others like LSA (Foltz et al 1998) require a large corpora as basis for comparison. Lately, there has been more interested in approaching the short answer scoring problem. These few such as MITRE (Hirschman et al, 2000) and ATM (Callear et al, 2001) are extraordinarily programming intensive however, and incomprehensible to educators. Additionally, they do not permit a partial credit scoring system, thereby introducing subjectivity into the scoring (Bachman 1990). None are truly suited for short answer scoring in an educational context, since the scores produced are neither easily explanable nor justifiable to testtakers.

WebLAS is developed in response to the needs of the language assessors. Current methods for scoring P&P tests require the test creators to construct a scoring rubrid, by which human scorers reference as they score student responses. Weblas imitates this process by prompting the test creator for the scoring rubrid. It tags and parses the model answer, extracts relevant elements from within the model answer and proposes possible alternatives interactively with the educator. It also tags, parses, and extracts the same from the student responses. Elements are then pattern matched and scored.

2 Using WebLAS

Just as a scoring rubric for short answer scoring cannot be created in a vacuum, it would be difficult for us to discuss the scoring process without describing the task creation process.

Task development consists of all the efforts that lead to the test administration. The task development portion of WebLAS consists of three modules- task creation, task modification, and lexicon modification. These are explained below.

2.1 Using WebLAS

WebLAS is written mostly in Perl. Its capacity for regular expressions (regex) make it well suited for natural language processing (NLP) tasks, and its scripting abilities enable dynamic and interactive content deliverable over the web. There is also a complete repository of open source Perl modules available, eliminating the necessity to reinvent the wheel.

One of the tools WebLAS incorporates is Wordnet, an English lexicon under development at Princeton with foundations in cognitive psychology (Fellbaum 1998). A second tool WebLAS uses is the Link Grammar Parser, a research prototype available from Carnegie Mellon University (Grinberg et al 1995). Both Wordnet and Link Grammar are written in C/C++. To interface with the systems, we make use of 2 Perl modules developed by Dan Brian[1]. Linguana::Wordnet converts to Berkeley DB format[2] for fast access, and allows for modifications to the lexicon. Linguana::LinkGrammar interfaces with the Link Grammar for parts of speech (POS) tagging and syntactic parsing. For our web server we use the Apache Advanced Extranet web server. To run perl scripts via the web, we use mod_perl, which enables us to run unmodified scripts. Our database is MySQL server[3].

2.2 Task Development

WebLAS is organized into four major components relative to the test event itself. These are test development, test delivery, response scoring, and test analysis. Two of these are relevant to NLP- task development and test scoring.

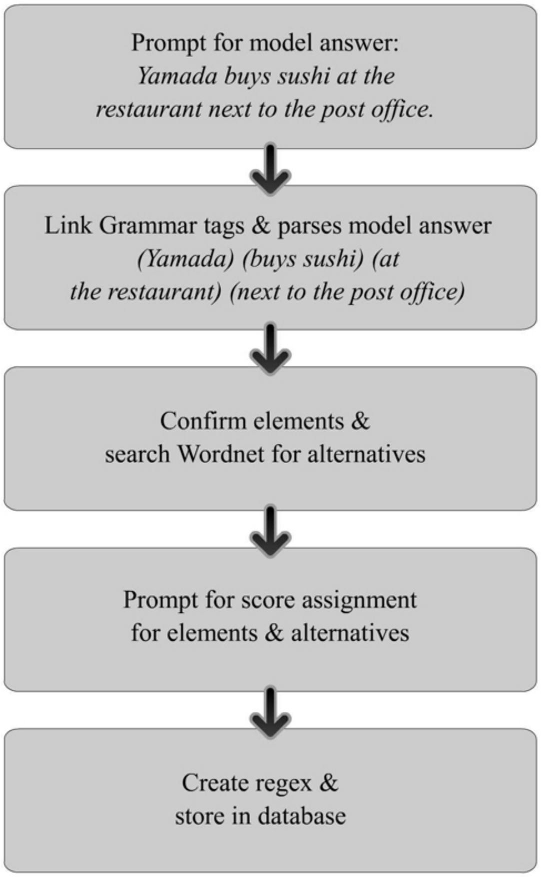
Prompt for model answer:
Yamada buys sushi at the restaurant next to the post office.
Link Grammar tags & parses model answer
(Yamada) (buys sushi) (at the restaurant) (next to the post office)
Confirm elements & search Wordnet for alternatives
Prompt for score assignment for elements & alternatives
Create regex & store in database

Figure 1. Task Creation Flowchart

[1] http://www.brians.org

[2] http://www.sleepycat.com

[3] http://www.mysql.org

2.2.1 Task Creation

The task creation module is somewhat of a misnomer. At the time of using this module, the task has already been specified according to language assessment requirements. The module actually facilitates the instructor with the process of storing into the database and preprocessing the task for automatic scoring, rather than creating the task itself. This process is shown in the flowchart in Figure 1.

a. The module requests from the instructor the task name, task type, input prompt, response prompt, and model answer for the task. This information is stored within the database for retrieval.
b. WebLAS sends Link Grammar the model answer, which returns the answer after tagging the POS and parsing it. WebLAS then finds important elements of the model answer which are necessary to receive full credit from the parsed answer and confirms each one with the instructor. Elements are generally phrases, such as "the sushi restaurant" or "next to the post office" but could be singletons as well, such as "Yamada-san" as well.
c. After each element is confirmed, WebLAS searches Wordnet for possible alternatives of the elements and their individual words. For example, it may deem "near the post office" as a possible alternate to "next to the post office." Once found, it asks for confirmation from the instructor again. Additionally, the educator is prompted for other possibilities that were not found.
d. The task creator dictates a ratings scale. Point values assigned to elements deriving directly from the model answer are assumed to be maximum values, i.e. full credit for the given element. Alternatives to the model answer elements found can be assigned scoring less than or equal to the maximum value. Thus an answer with numerous elements can be scored with a partial credit schema.
e. WebLAS takes the input (model answer, elements, alternatives, score assignments) to create a scoring key. The scoring key employs regular expressions for pattern matching. For example, "(next|near)" indicate that either "next" or "near" are acceptable answers. Along with each regex is a point value, which is added to a test taker's final score if the regex is matched with the student response.

2.2.2 Task Modification

The task modification module allows for instructors to go back and modify tasks they have created, as well as tasks others created. The database tracks information relevant to the changes, including information on the modifier, date and time of the modification, evolving changes to the tasks, and any comments on the reasons for the change. The database supports data synchronization, so that two instructors cannot and do not change tasks simultaneously.

Should the model answer be changed, the scoring key creation of the task creation module is activated and the instructor is guided through the process again.

2.2.3 Lexicon Modification

The WebLAS lexicon is based on Wordnet. Wordnet is by no means complete, however, and it may be possible that instructors may find the need to add to its knowledge. The lexicon is automatically updated given the input given during scoring key creation.

One can also manually modify the lexicon through a guided process. The system prompts for the word and its parts of speech and returns all possible word senses. The user chooses a word sense, and is then given a choice of the relation type to modify (i.e. synonyms, antonyms, hyponyms, etc.). The final step is the modification and confirmation of the change to the relation type.

2.3 Test Scoring

Once the task creation module creates the regexes, task scoring becomes trivial. WebLAS simply needs to pattern match the regexes to score each element. Additionally, WebLAS can be quite flexible in its scoring. It is tolerant of a wide range of answers on the part of test takers, incorporating adapted soundex, edit distances, and word stemming algorithms, for phonetic, typographic, and morphological deviations from model answers.

3 Lexicon Modification

There are advantages to the WebLAS system. The first is a computational efficiency factor. The system is not a learning system (yet). The automatic scoring section, if it did not use preprocessed regexes, would perform the same search for each student response. This search becomes redundant and unnecessary. By preprocessing the search, we reduce the linear time complexity- O(n), to a constant- O(1), with respect to the number of student responses.

Second, partial scoring eliminates arbitrariness of scoring. Rather than a simple credit/no credit schema, each element individually contributes to the final score tabulation.

Reliability also increases. Since the scores produced are repeatable, and do not change with each scoring, WebLAS has perfect intra-rater reliability. Because the instructor confirms all scoring decisions beforehand, the scores are also explainable and justifiable, and can withstand criticism.

4 Conclusion

Our approach towards automatic computer scoring of open ended responses show promising potential for reasons of its reliability and robustness. Future plans include making use of additional NLP algorithms such as inference and pronoun resolution, as well as inclusion of additional task types such as summarization, outline, and gap-fill tasks. We should also like to bring the scoring online and provide the student with instantaneous feedback. Pilot testing within the campus is scheduled for Winter and Spring 2003 quarters, with full campus rollout in Fall 2003.

References

Bachman, Lyle F. (1990) *Fundamental Considerations in Language Testing*. Oxford University Press: Oxford.

Bachman, Lyle F.; Palmer, Adrian S. (1996) *Language Testing in Practice*. Oxford University Press: Oxford.

Brian, Daniel. (2001) Linguana: Perl as a language for conceptual representation in NLP systems. *Proceedings of the O'Reilly Perl Conference 2001*. 24-31.

Burstein, Jill; Leacock, Claudia; Swartz, Richard. (2001) Automated evaluation of essays and short answers. *Proceedings of the 5th International Computer Assisted Assessment Conference* (CAA 01).

Callear, David; Jerrams-Smith, Jenny; Soh, David. (2001) CAA of short non-MCQ answers. *Proceedings of the 5th International Computer Assisted Assessment Conference* (CAA 01).

Chung, Gregory K.W.K; O'Neil, Harold F., Jr. (1997) *Methodological approaches to online scoring of essays*. University of California Los Angeles, Center for Research on Evaluation, Standards, and Student Testing technical report 461.

Grinberg, Dennis; Lafferty, John; Sleator, Daniel. (1995) *A robust parsing algorithm for link grammars*, Carnegie Mellon University Computer Science technical report CMU-CS-95-125, and *Proceedings of the Fourth International Workshop on Parsing Technologies*, Prague.

Hirschman, Lynette; Breck, Eric; Light, Marc; Burger, John D.; Ferro, Lisa. (2000) Automated grading of short answer tests. *IEEE Intelligent Systems*. 15(5):31-35.

Fellbaum, Christiane (editor). (1998) *Wordnet: An electronic lexical database*. MIT Press, Cambridge, MA.

Foltz P; Kintsch W; Landauer T. (1998) "The measurement of textual coherence with latent semantic analysis." *Discourse Processes*. 25(23):285-307.

Page, E.B. (1966) "The imminence of grading essays by computer." *Phi Delta Kappan.* 47:238-243.