# Statistical Analysis of Text in Educational Measurement

Claudia Leacock

Educational Testing Service – Rosedale Road – Princeton, NJ 08541 – USA
cleacock@ets.org

## Abstract

This paper describes tools developed at Educational Testing Service which use statistical modeling of textual corpora to provide automated assessments of student responses. Details are given for three of these systems: *Critique* Writing Analysis Tools for providing feedback to students, *e*-rater for assigning a holistic score to student essays, and *c*-rater, which scores responses to content-based, short-answer test or chapter review questions.

**Keywords:** automated scoring, *e*-rater, *c*-rater.

## 1. Introduction

At Educational Testing Service (ETS), a team of linguists and computer scientists that specializes in natural language processing have developed a variety of tools for instruction and for educational measurement. The goal of our work is to assist students and teachers by providing feedback on the form, quality, organizational structure, and content of writing. It is intended to be an aid, *not a replacement*, for classroom instruction. By providing automated feedback and essay evaluation, our tools ease the instructor's load, thereby enabling the instructor to give students more practice writing essays and answering test questions.

This paper describes three systems developed and deployed by ETS: *Critique* Writing Analysis Tools, *e*-rater, and *c*-rater. The *Critique* Writing Analysis Tools detect numerous errors in grammar, usage, and mechanics. They also highlight undesirable stylistic elements and provide information about essay-based discourse structure. *E*-rater assigns an overall or holistic score to an essay based on the kinds of criteria that human readers are asked to use in evaluating writing on standardized tests, such as the Test of English as a Foreign Language (TOEFL). *C*-rater scores responses to content-based questions by comparing each one to a gold-standard model of the correct answer. Although these applications vary in their methods and goals, they have much in common. Each was built from a training corpus in which features were extracted from examples, many of which had previously been categorized by human judges. The features were then used to build a model through statistical learning techniques (for *Critique* and *e*-rater) or manually (for *c*-rater) in order to categorize new student responses.

## 2. Grammar, Usage and Mechanics

The *Critique* Writing Analysis Tools identify errors in grammar, usage and mechanics. Grammar errors that are identified include subject-verb agreement (**A popular Mexican food are tacos*), ill-formed verbs (**It is must be miserable*), and the improper use of pronouns (**Them are my two reasons*). Usage errors include determiner noun agreement errors (**those*

*problem*) and faulty comparatives (**most strangest*). Finally, mechanics errors include punctuation errors such as a missing apostrophe (*the teachers book*) or hyphen (*the better known name*), or repetition of, for example, determiners (**the another town*).

A corpus-based, statistical approach is used to detect these violations of English grammar. The system is trained on a large corpus of edited text, from which it extracts and counts *bigrams* that consist of sequences of adjacent word and part-of-speech pairs. The system then searches student essays for bigrams that occur *much less often* than is expected based on the corpus frequencies.

The expected frequencies come from a model of English that is based on 30-million words of newspaper text. Every word in the corpus is tagged using a part-of-speech tagger (Ratnaparkhi, 1996) that has been trained on student essays. This tag set is subsequently enriched to include information about case and definiteness. For example, the sequence "I dropped the pencil" would be tagged as: I_PPSS dropped_VBD the_ATI pencil_NN. The tag PPSS indicates that "I" is a singular subject pronoun, VBD the past form of a verb, ATI a singular definite determiner and NN a singular noun. Bigrams are created from adjacent sequences of part-of-speech tags and of function words. The bigrams generated from the sequence are: I_VBD, PPSS_VBD, VBD_the, VBD_ATI, the_NN, ATI_NN.

To detect violations of English grammar, the system compares observed and expected frequencies in the corpus. The statistical methods that the system uses are commonly used by researchers to detect combinations of words that occur with greater frequency than would be expected if the words were independent. These methods are typically used to find technical terms or collocations. We use the measures for the opposite purpose – to find combinations that occur *less often* than expected, and therefore might be evidence of a grammatical error (Chodorow and Leacock, 2000).

The system uses two complementary methods to measure association: pointwise mutual information and the log-likelihood ratio. Pointwise mutual information gives the direction of association (whether a bigram occurs more often or less often than expected based on the frequencies of its parts). However, mutual information is known to be unreliable when the data are sparse, while the log-likelihood ratio performs better with sparse data. The log-likelihood ratio gives the likelihood that the elements in a sequence are independent (we are looking for non-independent, dis-associated words), but it does not tell whether the sequence occurs more often or less often than expected. By using both measures, we get both the direction and the strength of association, and performance is better than it would otherwise be when data are limited.

Of course, a model based on adjacent pairs cannot capture English grammar. For this reason, we have developed rule-based filters to allow for low probability sequences that are, in fact, grammatical. For example, in the sequence *these pencil erasers*, where the singular noun *pencil* is part of a plural compound noun, the error message is suppressed.

## 3. Confusable Words

Some of the most common errors in writing are due to the confusion of homophones, words that sound alike but are spelled differently. Martin Chodorow implemented a system to detect errors among *their/there/they're, its/it's, write/right* and hundreds of other homophone sets. For the most common homophones, we extracted 10,000 training examples of correct usage from newspaper text and built a representation of the local context in which each confusable word occurs. The context we use is a five word window : the two words and part-of-speech

tags that precede the confusable word, and the two that follow it. For example, a context for *right* might be "*find the right person to*", consisting of a verb and determiner that precede the homophone, and a noun that follows it. For *write*, an example of a local context is "*they will write the script*", where *write* is preceded by a subject pronoun and modal verb, and followed by a determiner and noun.

Sometimes one or both words in a confusable word pair are so rare that a large training set cannot be assembled from the corpora available to us. One example is the verb *purl* in the pair *purl/pearl*. In this case, Chodorow has developed generic representations. The generic local context for nouns consists of all the part-of-speech tags found in the two positions preceding each noun and in the two positions that follow it in a large textual corpus. Generic local contexts are created for verbs, adjectives, adverbs, and so on. These serve the same role as the word-specific representations built for more common homophones. Thus, *purl* would be represented as a generic verb and *pearl* as a generic noun.

The frequencies found in training are used to estimate the probabilities that particular words and parts of speech will be found at each position in the local context. When a confusable word is encountered in an essay, *Critique* uses a Bayesian classifier (Golding 1995) to select the more probable member of its homophone set, given the local context in which it occurs. If this is not the word that appears in the essay, then the system highlights it as an error and suggests the more probable homophone. For example, when the system encounters *write* in "We have the *write* to express our opinions", it is highlighted and a pop-up window suggests that it be changed to *right.*

For reporting errors that are detected using bigrams and errors caused by the misuse of confusable words, we have chosen to err on the side of precision over recall. That is, we would rather miss an error than tell the student that a well-formed clause is ill-formed. A minimum threshold of 90% precision was set in order for a bigram error or confusable word set to be included in the writing analysis tools.

Since the threshold for precision is between 90-100%, the recall varies from bigram to bigram and confusable word set to confusable word set. To estimate recall, 5,000 sentences were annotated to identify specific types of grammatical errors. For example, the writing analysis tools correctly detected 40% of the subject-verb agreement errors that the annotators had identified and 70% of the possessive marker errors. Errors in the use of confusable word were detected 71% of the time.

## 4. Elements of Style

*Critique* also looks at larger domains to give some stylistic information about the essay as a whole, such as the number of words and number of sentences. It points out stylistic constructions and forms that are usually, but not always, considered to be undesirable, such as the use of passive sentences and of overly-long sentences. Of most interest to us, since it is a real problem in student essays, is the identification of highly repetitious use of words in an essay.

In order to create the training corpus, two writing experts identified which words in an essay are repeated or overused so much that the repetition interferes with a smooth reading of the essay. Since this is a very subjective judgment – what is irritating to one reader may seem fine to another – repetitious word use is not presented as being an error. Instead, *Criterion* highlights repetitiveness and lets the student judge whether the essay would be improved with revision.

The system uses a machine learning approach to finding excessive repetition. It was trained on a corpus of 300 essays in which two judges had labeled the occurrences of overly repetitious words. Seven features were found to reliably predict which word(s) should be labeled as being repetitious. These include the word's total number of occurrences in the essay, its relative frequency in the essay and in a paragraph, its length in characters, and the average distance between its successive occurrences. Using these features, a decision-based machine learning algorithm, C5.0, was used to model repetitious word use, based on the human experts' annotations. See Burstein and Wolska (2003) for a detailed description and system results.

Not surprisingly, the two human judges showed considerable disagreement with each other in this task, but each judge was internally consistent. When the repetitious word detection system was trained on data of a single judge, it could accurately model that individual's performance with 95% precision and 90% recall.

## 5. Discourse Structure

Scoring rubrics for essays typically define an excellent essay as one that "*develops its arguments with specific, well-elaborated support*" and a poor essay as one that "*lacks organization and is confused and difficult to follow*." In order for an essay to be well-structured, it should contain introductory material, a thesis statement, several main and supporting ideas, and a conclusion.

The organization and development module identifies these discourse components in each essay. In order to do this, a training corpus was created by two writing experts who annotated a large sample of student essays identifying and labeling each discourse unit in the essays. The discourse analysis component uses a decision-based voting algorithm that takes into account the discourse labeling decisions of three independent discourse analysis systems. Two of the three systems use probabilistic methods, and the third uses a decision-based approach to classify a sentence in an essay as a particular discourse element. For a description of the three classifiers and detailed results, see Burstein *et al.* (2003).

When too few discourse elements are identified, *Critique* offers suggestions to the student about how to improve the essay's discourse structure. For example, if only a single main idea is identified, then the *Critique* will advise the student to incorporate two more main ideas into the essay.

To evaluate the system's performance, it was compared against a baseline algorithm. The baseline algorithm assigns a discourse label to each sentence in an essay based solely on its position within the essay. For example, a baseline algorithm would label the first sentence of every paragraph in the essay as a main point. For the baseline algorithm, the overall precision is 71% and the overall is 70% while the precision and the recall for the discourse analysis system are both 85%.

## 6. *E*-rater: Essay Scoring

In large assessment programs in the United States, such as TOEFL or the Graduate Management Admissions Test, the student is asked to write an essay-length response to a writing prompt within a limited amount of time. A holistic score is assigned to the essay based not so much on the content of the student's response (there is no correct or incorrect answer), but on how well a student frames that response in the context of an essay. These

holistic scores usually range from 1 for a poorly written essay to 6 for an excellent, well-crafted essay. The holistic scoring rubric bases the essay score on such features as the essay's clarity and persuasiveness, the organization and development of its ideas, the use of sentence variety, the choice of language, and the essay's overall grammatical correctness.

*E*-rater is the automated essay scoring engine that has been developed and deployed at ETS. *E*-rater version 2.0, which is described here, was invented by Jill Burstein, Yigal Attali and Slava Andreyev and will be deployed in Spring, 2004. The *e*-rater 2.0 feature set is closely aligned to the scoring rubric. For example, two features that are derived from the Writing Analysis Tools' organization and development component directly address the essay's organization and the development of its ideas. Three features are derived from the grammar, usage and mechanics components to assess the grammatical correctness and use of punctuation in the essay. Features that are based on content vector analysis, type/token ratios, and an index of word difficulty evaluate the choice of language. For a full description of *e*-rater 2.0, see Burstein, et al (forthcoming).

To model human scores, *e*-rater needs to combine this homogeneous set of features that correspond to the elements in the scoring rubric. It builds an individual model for each essay question by training on a sample of about 250 essays that two human readers have scored and that represent the range of scores from 1 to 6. *E*-rater uses a multiple regression approach to generate weights for the feature set. The result of training is a regression equation that can be applied to the features of a novel essay to produce a predicted score value. The last step in assigning an *e-rater* score is to convert the continuous regression value to a whole number along the six-point scoring scale.

The performance of *e*-rater 2.0 is evaluated by comparing its scores to those of human judges. This is carried out in the same manner that the scores of two judges are measured during reader scoring sessions for standardized tests such as TOEFL. If two judges' scores match exactly, or if they are within one point of each other on the 6-point scale, they are considered to be in *agreement*. Typical agreement between *e-rater* 2.0 and the human score is approximately 97%, which is comparable to agreement between two human readers.

## 7. *C*-rater: Scoring for Content

*C*- rater is an automated scoring engine under development at ETS that is designed to measure a student's understanding of specific content material. Unlike the Writing Analysis Tools and *e*-rater, it measures the student's understanding with little regard for writing skills.

Below is an example of the type of question that c-rater is designed to score. This example is an approximation of a 4th grade math question used in the National Assessment for Educational Progress (NAEP):

> A radio station wanted to determine the most popular type of music among those in the listening range of the station. Would sampling opinions at a country music concert held in the listening area of the station be a good way to do this?
>
> ▢ YES ▢ NO
>
> Explain your answer.

To answer the first part of the question, the student indicates "yes" or "no" by filling in a bubble. *C*-rater can be used to score the second part of the question – the student's explanation.

To create a *c*-rater scoring model, a content expert, such as a test developer or a teacher, needs to develop a scoring guide that breaks down how many score points are assigned to each of the component concepts that make up a correct answer. In the example above, only one concept needs to be identified in order for the student to receive credit – the concept that the sample would be biased. The content expert, working with a set of scored student responses, must identify the variety of ways that the concept can be expressed lexically and syntactically. We call these *gold standard* responses. In this case, most of the students express the concept of bias by saying something like: "They would say that they like country music".

The *c*-rater scoring engine tries to recognize when a response is equivalent to a correct answer. *C*-rater is, in essence, a paraphrase recognizer. In order to recognize paraphrases, it breaks down each student response into its underlying predicate argument structure, resolves the referent of any pronouns in the response, regularizes over morphological variation, matches on synonyms or similar words, and tries to resolve the spelling of unrecognized words. The resulting canonical representations are mapped onto canonical representations of the gold standard responses.

In the current version of *c*-rater, the module that maps between the canonical representation of a gold standard answers and that of a student response is rule driven – *not* statistical. The rule-driven approach allows for good accuracy. In a pilot study with the state of Indiana, *c*-rater was used to score seven 11th grade reading comprehension items. On average, *c*-rater had exact agreement with human scorers 84% of the time. (See Leacock and Chodorow, 2003b for a full description of *c*-rater and the pilot study.) However, the scoring engine is unable to give an indication of its confidence about the score. If *c*-rater could generate a reliable measure of confidence, then it could be used to grade those responses that it is confident about and an instructor would only need to grade the responses on which system confidence is low.

Thomas Morton is currently developing a statistical version of *c*-rater that gives a score as well as a probability indicating the system's confidence that the score is correct. This version trains a maximum entropy model on a corpus of student responses, where the mapping to one or more corresponding gold standard sentences whose meaning they capture was manually annotated. The features used to do this are based on lexical similarity as well as structural similarity. The lexically-based features include exact string match, head word match, and synonymous words. The structural features are based on the labels of the constituents where matching arguments are found. For example, the model can learn that a subject can typically be matched to an object in a passive sentence. Thus the model learns which syntactic variations maintain meaning and which ones do not. Since this model is based on syntactic variations and lexical similarity, and not the actual lexical items found in the corpus, it can be used for any question that exhibits the same types of syntactic variation that are found in the training data.

Both versions of *c*-rater generate a score based on the similarity of the response to the gold standard answer. This is done by comparing each gold standard sentence to each sentence in the response, and a score is assigned based on which concepts are present in the response. To generate a measure of confidence in the score, the machine learning version keeps a ranked set of the most likely mappings between a response and the gold sentences. Each of these sets of mappings is scored and the weight of the set is used to produce a distribution of scores.

## 8. Conclusion and Future Directions

In developing tools for instruction and for educational measurement we have systematically exploited the information that can be found in annotated and unannotated textual corpora.

With *e*-rater, the annotation consists of a holistic score. In the Writing Analysis Tools, the annotations either identify the elements of discourse in an essay or identify overly-repetitious word use. Statistical classifiers train on the annotated corpora to learn how to annotate new test responses. In *c*-rater, a corpus of aligned sentences is used to generate a model of meaning-preserving variations. We have also exploited large unannotated corpora of published, well-formed text to build a model of well-formed English against which individual student responses are compared in order to identify errors in grammar, usage, and mechanics.

We are currently expanding the functionality of the Writing Analysis Tools, *e*-rater and *c*-rater. For example, we are implementing the detection of grammatical errors that are important to specific native language groups (Han *et al.*), such as identifying when a determiner is missing (a common error among native speakers of East Asian languages and of Russian) or when the wrong preposition is used. The current organization and development module of the Writing Analysis Tools identifies discourse elements but does not evaluate their quality. We are extending the analysis of discourse so that the expressive quality of each discourse element can also be assessed. As previously mentioned, we also plan to replace the c-rater module that assigns a score with a statistically based module that assigns a score along with a confidence measure.

**Acknowledgements:** I am grateful to Martin Chodorow, Ray C. Dougherty and Thomas Morton for helpful comments and suggestions. Any opinions expressed here are those of the authors and not necessarily of the Educational Testing Service.

## References

Burstein J., Chodorow M. and Leacock C. (forthcoming). Automated essay evaluation: The *Criterion*$^{SM}$ Online Service. *AI Magazine.*

Burstein J., Marcu D. and Knight K. (2003). Finding the WRITE stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems: Special Issue on Natural Language Processing*, vol. (18/1): 32-39.

Burstein J. and Wolska M. (2003). Toward evaluation of writing style: Overly repetitious word use in student writing. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary.

Chodorow M. and Leacock C. (2000). An unsupervised method for detecting grammatical errors. *Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics*: 140-147.

Golding A. (1995). A Bayesian hybrid for context-sensitive spelling correction. In *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge, MA: 39-53.

Han N-R., Chodorow M. and Leacock C. (2004). Detecting errors in English article usage with a maximum entropy classifier trained on a large, diverse corpus. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. Lisbon, Portugal.

Leacock C. and Chodorow M. (2003a). Automated grammatical error detection. In Shermis M.D. and Burstein J. (Eds), *Automated Essay Scoring: A Cross-Disciplinary Perspective*. Lawrence Erlbaum.

Leacock C. and Chodorow M. (2003b). C-rater: Automated scoring of short answer questions. *Computers and the Humanities*, vol. (37/4).

Ratnaparkhi A. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*. University of Pennsylvania.