


title: Improving Automatic English Writing Assessment Using Regression Trees and Error-Weighting

**SUMMARY** The proposed automated scoring system for English writing provides an assessment result including a score and diagnostic feedback to test-takers without human’s efforts. The system analyzes an input sentence and detects errors related to spelling, syntax and content similarity. The scoring model has adopted one of the statistical approaches, a regression tree. A scoring model in general calculates a score based on the count and the types of automatically detected errors. Accordingly, a system with higher accuracy in detecting errors raises the accuracy in scoring a test. The accuracy of the system, however, cannot be fully guaranteed for several reasons, such as parsing failure, incompleteness of knowledge bases, and ambiguous nature of natural language. In this paper, we introduce an error-weighting technique, which is similar to term-weighting widely used in information retrieval. The error-weighting technique is applied to judge reliability of the errors detected by the system. The score calculated with the technique is proven to be more accurate than the score without it.

key words: automated scoring system, error-weighting, regression tree, error frequency

1. Introduction

One of the best ways to improve one’s writing skills is to practice writing as many times as possible. While repeating the practice, identifying errors from the writings is one of the key factors for improvement. The errors are identified and their corresponding answers are suggested by a human instructor. However, this process requires enormous time and effort from human instructors. The large volume of writings makes the process even more difficult. Furthermore, consistency is not easy to maintain during the process of scoring. Considering these, an automated scoring system would ease the workload and the efforts of human instructors while contributing to enhancement of writing skills.

The automated English scoring system that this paper presents analyzes a single sentence using extended grammar tailored for detecting errors and offers a score for the sentence. Figure 1 shows the overall process of automatically scoring English sentences. As the first phase of the process, teachers provide the system with a set of English writing questions and their corresponding answers. A writing question is a Korean sentence which test-takers are supposed to translate into an English sentence representing the same meaning. Since more than one correct answer can be mapped to a question in most cases, all the possible answers should be provided to the system. The answers are parsed and the results are stored in a database. When the first phase is completed, the system is ready for the scoring process. A test-taker composes a sentence according to a given question. The system analyzes the test-taker’s input sentence and compares it with a set of correct answers. During the scoring process, the system detects spelling, mechanical and syntactic errors as well as the differences between the input sentence and the answer. With regard to the scoring result as shown in Fig. 1, (a) and (b) briefly describes error types, a syntactic error and a misspelling error, respectively. (c) and (d) are the errors detected by comparing the input sentence with the correct answer. Finally, the input sentence is awarded with the score 3, which is calculated based on the number and the types of the errors. When the scoring process is completed, the test-taker is given the test result which includes not only the overall score but also diagnostic feedback.

The usefulness of an automated scoring system depends on the accuracy of the score calculated by the system. In other words, a system should be able to calculate scores as close as possible to those which human raters provide. Human raters assign a score, considering both the number of errors and the types of errors identified from a test-taker’s sentence. The same factors have to be considered and implemented in the system to be able to produce the same result. Based on this assumption, we have built a score predictor model, using a regression tree. The regression tree grows using the number of errors and the types of errors collected...
Table 1 Examples of automated scoring results (I).

| Test-taker’s Input: ears is bigger than the moon. | Test-taker’s Input: As the prices become lower, people buy more. |
|------------------------------------------------|---------------------------------------------------------------|
| Answer Sentence: The Earth is bigger than the moon. | Answer Sentence: As the prices become lower, people buy more. |
| (Eₘ) FIRST_WORD_CASE_ERROR[1] | (Eₘ₁) SUBJ_VERB_AGR_ERROR[7]buy |
| (Eₘ₂) SUBJ_VERB_AGR_ERROR[1] | |
| (Eₘ₃) OBLIGATORY_NODE_MISSING_ERROR[1]-2 | (Eₘ₂) _MISSING_ERROR[8]more |
| (Eₘ₄) UNNECESSARY_NODE_ERROR[1](ear) | |

from various processing levels. As a score is directly related to the errors, the accuracy in detecting errors is essential for determining the performance of the system. However, it seems that the accuracy of errors produced by the automated scoring system is not always reliable, which could result in a disappointing performance of a scoring model built by a regression tree. Table 1 reports a disappointing case.

The example (EX1) shows one of the test-taker’s input sentences composed according to the question 76†. Among the answer set, “The earth is bigger than the moon.” is selected to be mapped to the input sentence based on the similarity of the two sentences. The following four lines labeled with (Eₘ) to (Eₘ₄) are the error reports†† on the input sentence produced by the automated scoring system. The example (EX2) is the analysis result of the question 29. ‘SUBJ_VERB_AGR_ERROR’, a subject-verb agreement error, has been reported for both sentences; correctly detected for (EX2), but not so for (EX1). The system detects the error ‘SUBJ_VERB_AGR_ERROR’ from the input string “ears is” in the case of (EX1). On the other hand, human raters detect an error at the same position of the sentence, but classify it as a minor spelling error; “earth” has been confused with “ears” due to similarities in their pronunciation and spelling. A system detects errors differently from the way a human does, and it sometimes fails to analyze a sentence correctly. Because of these reasons, automatically detected errors are not reliable enough to be converted directly to a score. The incorrectly detected error from the test-taker’s input in (EX1) misleads the system into calculating an incorrect score. In other words, the accuracy in detecting errors influences the process of calculating the final score.

The substring “ears is” hardly occurs among the test-takers’ sentences for the question 76††† because it is very rare for a test-taker to confuse ‘ears’ with ‘earth’. In comparison, the same type of an error often occurs as in a substring “people buys” for the question 29 shown in (Eₘ₃) because the confused usage of ‘people’ in terms of number is rather common. When a particular error occurs very frequently for a specific question throughout the test-takers’ answers, the error can be regarded as reliable. In other words, it is very likely that an infrequent error for a specific question is detected incorrectly.

In this paper we introduce a new error-weighting method in order to estimate reliability of an error. The basic information used in error-weighting is how frequently an error is detected for a specific question and how extensively an error is detected across the questions. The idea is very similar to the term-weighting technique commonly used in the study of information retrieval. Term-weighting generally consists of a term frequency and an inverse document frequency. An error with higher weight is regarded as more reliable than an error with lower weight; the error with lower weight is likely to have been detected incorrectly. Therefore, the errors with different weight are handled differently in calculating the final score. Implementing the new method has increased the accuracy of automated scoring on the whole.

This paper is organized as follows. Section 2 outlines related studies. Section 3 provides an overview of the automated English writing scoring system. Section 4 describes the error-weighting technique for automated scoring. Section 5 discusses the experimental result. Finally, Sect. 6 concludes the paper.

2. Related Studies

The research presented by Burstein, Chodorow and Leacock [2], Burstein and Higgins [3], Burstein, Marcu and Knight [4] introduces Criterion, an online English writing evaluation system developed by Educational Testing Service (ETS). Criterion consists of two parts: e-rater, a system dealing with score calculation, and Critique, a system providing feedback on the test-taker’s composition errors. Critique detects numerous errors on the areas of grammar, usage, and mechanics. It also identifies undesirable writing styles for an essay; too long or too short sentences, passive sentences, and excessive repetition of a word. Critique uses n-gram information extracted from large corpus or mutual information in order to check intra-sentential errors on number/person agreement, verb formation, word usage, punctuation marks, and spelling. E-rater calculates scores, taking into account the syntactic variety of writing styles, essay organization, appropriateness of topic, and lexical complexity. E-rater V.2 uses 12 features, half of which are derived from Critique. Those 12 features are extracted from test-takers’ English essays, and converted into the scores ranging from 1 to 6 by means of a regression model.

IntelliMetric™ developed by Vantage Learning is an intelligent English essay scoring system that emulates the process carried out by human raters. It is theoretically grounded on a blend of artificial intelligence, cognitive pro-

†You can see the questions and their identification numbers at http://marble.cnu.ac.kr/~kice/StarSun/scoring.php
††The format of an error report is described in detail in Sect. 5.2.
†††For the question 76, more frequently occurred errors related to agreement are the ones detected in the string “earth are” or “earths is”. In these cases the errors form “SUBJ_VERB_AGR_ERROR[1-6]is” and “SUBJ_VERB_AGR_ERROR[1-6]are”, respectively.
cessing, natural language processing and statistical technologies [11], [13]. An input text composed by a test-taker is analyzed to tag the discourse and grammatical structure of an essay. Several technologies are applied to examine the text in order to extract useful features associated with the final score. More than 500 linguistic and grammatical features are introduced into the system. These features fall into four major categories: (1) discourse/rhetorical features, (2) content/concept features, (3) syntactic/structural features, (4) mechanics features. These four kinds of features influence the following five components of scoring: (1) focus and unity—the features pointing toward cohesiveness and consistency in purpose and main idea, (2) development and elaboration—the features of text looking at the breadth of content and the support for concepts advanced, (3) organization and structure—the features targeted at the logic of discourse including transitional fluidity and relationships among parts of the response, (4) sentence structure—the features targeted at sentence complexity and variety, (5) mechanics and conventions—the features examining conformance to conventions of edited American English. This set of the features is coded to support computation of multiple mathematical models. The models differ in their mathematical form and the included features. In addition, the IntelliMetric™ system adopts optimization technique by which the information from different models is integrated into the final score. Using multiple mathematical models can be regarded as analogous to using multiple judges in the system.

The fundamental distinction of our system from Criterion or IntelliMetric™ can be the unit of the input for evaluation; the system works on a single sentence rather than an essay. The single sentence is evaluated against a set of correct answers which is provided by teachers and analyzed beforehand.

Neither a rhetorical structure nor a discourse structure can be extracted from a single sentence. The features extracted by analyzing a single sentence include mechanical, spelling, and syntactic errors as well as misused words identified by comparing the input to a correct answer sentence. Since these are the only errors available to associate with a score, the accuracy of the final score depends on the accuracy in detecting errors.

When we focus on an evaluation issue only, our work can be compared with the automatic evaluation of machine translation systems [10], [14] in that both try to evaluate sentences according to reference answers. Their works are generally n-gram based, which do not provide enough information to identify which kind of error is detected in a sentence. [14] has suggested a diagnostic evaluation platform for MT systems that can score a sentence according to predefined linguistic categories. It treats a sentence as a collection of linguistic patterns which a score is assigned to. While their evaluation results are meaningful for MT developers because it provides total scores to each linguistic pattern, the same approach is not appropriate for our system which is designed to improve test-takers’ writing skills. To provide more effective help to test-takers, feedback should contain not only the total score but also detailed error information. The proposed system keeps track of where an error occurs in a sentence and which kind of error is detected. According to the traced information, the system suggests how to correct the error. Each detected error is considered as meaningful information for test-takers. Therefore, the accuracy of detecting errors is an essential factor to improve the accuracy of the system. In this paper, we introduce a new technique that judges the reliability in detecting errors.

### 3. Automated English Writing Scoring System

Figure 2 presents an overall architecture of the automated English writing scoring system developed through this research [9]. The system consists of two modules. The first module is “intra-sentential error detection module” that recognizes errors occurring within a sentence. As the first step, a morpho-syntactic analysis is performed on the set of correct answers provided by teachers. The teachers provide not only a set of correct answers but also appropriate synonyms for each word appearing in correct answers, if available, which makes a word-for-word based paraphrase easier. The teacher-provided sentences are then converted to simplified dependency structures based on the result of the morpho-syntactic analysis. A dependency structure is composed of a set of nodes and relations between the nodes. A content word in a sentence is basically mapped into a node, which denotes its root form and grammatical features such as tense, number, person, and aspect. A relation represents a syntactic relation between the nodes, such as subject, object and so on.

When a test-taker completes entering a sentence, a morpho-syntactic analysis is performed on the input sentence. The syntactic analyzer presented in this research has 316 grammar rules; 235 for analyzing grammatical sentences, 26 for handling ungrammatical sentences, and 55 for covering both grammatical and ungrammatical sentences.
When analyzing correct answers, the syntactic analyzer uses only 290 rules. On the other hand, it activates all of the 316 rules, expecting errors to be encountered when analyzing test-takers' sentences since it is very likely that test-takers' sentences have errors. By doing so, the system detects the various types of errors including spelling, mechanical, morphological and syntactic errors. As illustrated in Fig. 2, the morpho-syntactic analysis can identify word and syntactic errors if the test-taker's answer contains errors. The information on the errors detected in this module is saved for an error report to be prepared at a later stage. The test-taker's sentence is also converted into simplified dependency structures, using the syntactic analysis information.

The second module is “inter-sentential error detection module” which recognizes errors through comparing a test-taker’s sentence with its correct answers. Even when there is no error in the test-taker’s sentence, the sentence may not coincide with any of the teacher-provided answers for various reasons; 1) not using appropriate tense, number, or person, 2) failing to convey what is meant by the question, 3) including unnecessary words or expressions in the sentence, and so on. Such sentences are evaluated as grammatical by the morpho-syntactic analyzer, but cannot be a correct answer for the question. In this case, the errors can only be recognized by comparing the test-taker’s sentence with the teacher-provided answer set. Semantic similarity between the test-taker’s sentence and the correct answer is evaluated by identifying the differences between the two dependency structures, each of which is the input to the inter-sentential error detection module. The dependency structure with the highest similarity among the set of correct answers is finally chosen to be compared with the test-taker’s sentence. Each node in the dependency structure is mapped to the most similar node of the corresponding dependency structure among the answers. While comparing the two dependency structures, the system detects mapping errors including mismatching feature values between the nodes, missing nodes, and unnecessary nodes. Given the error information obtained through the intra-sentential module and the inter-sentential module, the system reports the scoring result, and offers corresponding diagnostic feedback to the test-taker.

### 3.1 Error Coverage

Table 2 shows the error types that can be identified by the current version of the system. The system recognizes 82 different types of errors; 21 of word errors, 46 of syntactic errors, and 15 of mapping errors.

A word error ‘non-alphanumeric char error’ is activated when the system detects an incomplete sentence containing Korean characters. Test-takers often repeat the Korean words composing the question when not knowing the corresponding English expression. A syntactic error, ‘incomplete sentence error’ is implemented to detect a sentence

| word error                                                                 | syntactic error                                                                 | mapping error                                                                 |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| (1) sentence initial word case error                                      | (1) obligatory node missing error                                               | (1) unnecessary node error                                                     |
| (2) non initial word upper case error                                      | (2) optional node missing error                                                 | (2) modal matching error                                                       |
| (3) proper noun case error                                                 | (3) tense missing error                                                         | (3) noun word number agreement error                                           |
| (4) sentence final punctuation error                                       | (4) lexical mismatching error                                                   | (4) ungrammatical preposition error                                            |
| (5) general punctuation error                                               | (5) mood mismatching error                                                     | (5) tense mismatching error                                                    |
| (6) hyphen usage error                                                     | (6) voice mismatching error                                                     | (6) aspect mismatching error                                                    |
| (7) apostrophe usage error                                                  | (7) agreement mismatching error                                                 | (7) verb to-infinitive form error                                              |
| (8) non-alphanumeric char error                                            | (8) agreement mismatching feature                                               | (8) verb perfective form error                                                 |
| (9) spacing error                                                          | (9) agreement matching error                                                    | (9) verb voice form error                                                      |
| (10) compound-word spacing error                                           | (10) agreement matching feature                                                 | (10) verb bare infinitive form error                                           |
| (11) pluralization error I                                                 | (11) agreement matching feature                                                 | (11) verb past tense error                                                     |
| (12) pluralization error II                                                | (12) agreement matching feature                                                 | (12) verb to-infinitive form error                                             |
| (13) verb inflection error                                                 | (13) agreement matching feature                                                 | (13) verb past participle error                                                |
| (14) adverb/adjective -er -est form error                                  | (14) agreement matching feature                                                 | (14) verb gerund form error                                                    |
| (15) spelling error                                                        | (15) agreement matching feature                                                 | (15) verb gerund form error                                                    |
| (16) spelling spacing error                                                | (16) agreement matching feature                                                 | (16) verb past participle error                                                |
| (17) confusabale word error                                                | (17) agreement matching feature                                                 | (17) verb past participle error                                                |
| (18) idiom usage error                                                     | (18) agreement matching feature                                                 | (18) verb past participle error                                                |
| (19) frequently misused word error                                         | (19) agreement matching feature                                                 | (19) verb past participle error                                                |
| (20) word formality error                                                  | (20) agreement matching feature                                                 | (20) verb past participle error                                                |
| (21) determiner unmatched error                                            | (21) agreement matching feature                                                 | (21) verb past participle error                                                |
| (22) determiner & 1st letter of noun agreement error                       | (22) agreement matching feature                                                 | (22) verb past participle error                                                |
| (23) agreement matching feature                                             | (23) agreement matching feature                                                 | (23) verb past participle error                                                |
| (24) agreement matching feature                                             | (24) agreement matching feature                                                 | (24) verb past participle error                                                |
| (25) agreement matching feature                                             | (25) agreement matching feature                                                 | (25) verb past participle error                                                |
| (26) agreement matching feature                                             | (26) agreement matching feature                                                 | (26) verb past participle error                                                |
| (27) agreement matching feature                                             | (27) agreement matching feature                                                 | (27) verb past participle error                                                |
| (28) agreement matching feature                                             | (28) agreement matching feature                                                 | (28) verb past participle error                                                |
| (29) agreement matching feature                                             | (29) agreement matching feature                                                 | (29) verb past participle error                                                |
| (30) agreement matching feature                                             | (30) agreement matching feature                                                 | (30) verb past participle error                                                |
| (31) agreement matching feature                                             | (31) agreement matching feature                                                 | (31) verb past participle error                                                |
| (32) agreement matching feature                                             | (32) agreement matching feature                                                 | (32) verb past participle error                                                |
| (33) agreement matching feature                                             | (33) agreement matching feature                                                 | (33) verb past participle error                                                |
| (34) agreement matching feature                                             | (34) agreement matching feature                                                 | (34) verb past participle error                                                |
| (35) agreement matching feature                                             | (35) agreement matching feature                                                 | (35) verb past participle error                                                |
| (36) agreement matching feature                                             | (36) agreement matching feature                                                 | (36) verb past participle error                                                |
| (37) agreement matching feature                                             | (37) agreement matching feature                                                 | (37) verb past participle error                                                |
| (38) agreement matching feature                                             | (38) agreement matching feature                                                 | (38) verb past participle error                                                |
| (39) agreement matching feature                                             | (39) agreement matching feature                                                 | (39) verb past participle error                                                |
| (40) agreement matching feature                                             | (40) agreement matching feature                                                 | (40) verb past participle error                                                |
| (41) agreement matching feature                                             | (41) agreement matching feature                                                 | (41) verb past participle error                                                |
| (42) agreement matching feature                                             | (42) agreement matching feature                                                 | (42) verb past participle error                                                |
| (43) agreement matching feature                                             | (43) agreement matching feature                                                 | (43) verb past participle error                                                |
| (44) agreement matching feature                                             | (44) agreement matching feature                                                 | (44) verb past participle error                                                |
| (45) agreement matching feature                                             | (45) agreement matching feature                                                 | (45) verb past participle error                                                |
| (46) agreement matching feature                                             | (46) agreement matching feature                                                 | (46) verb past participle error                                                |

### Table 2 Error types covered by the system.
which is too short or too messy to be analyzed successfully. As a mapping error is detected by comparing a test-taker’s input sentence with an answer sentence, the majority of errors in this phase are classified under ‘mismatching’ errors.

To expand the coverage of the system, the answer set includes two different subsets of sentences. One of the subsets is the collection of sentences which syntactically and semantically coincides with what each question intends. The other subset includes syntactically correct sentences which are semantically acceptable, but slightly different from what the question implies. When the input sentence is determined to be the closest to one of the second subset, the sentence is marked with ‘meaning difference error’.

After completing the error detecting process, another processing module is activated to compute an edit distance between the test-taker’s sentence and the answer sentence. When an edit distance between the sentences is a single word, the input sentence is marked with ‘one-word edit distance error’. This particular error is interpreted as the sentences are identical except for a word.

### 3.2 Rudimentary Score Calculation Model

The scores are computed at the following three phases: word, syntax, and mapping. The final score is calculated by summing up the scores of each phase. The maximum score is 6, given each phase with the score ranged from 0 to 2. We asked junior high school teachers to deliver basic criteria to score a sentence. We have rephrased teachers’ criteria for scoring heuristics as depicted in Fig. 3 and Table 3. The criteria are common to every question regardless of the length of a sentence and the degree of question difficulty.

Teachers suggested scoring 0 on the sentences containing non-alphanumeric characters or missing an obligatory constituent such as subject or a main verb. According to the criteria, a sentence with 3 or more syntactic errors is to be scored 0.

This scoring heuristics is implemented based on teachers’ holistic evaluation rubric of writings. In Sect. 5.1, we will show how well human’s heuristics work by comparing scoring results produced by human and the system respectively.

### 4. Improving Automated Scoring Model Using Error-Weighting

We have implemented a scoring model based on a regression tree. A regression tree is a tool which can predict a target continuous value, which is a score in this case, based on the input features—the number of errors and their types.

#### 4.1 Training Data and Feature Set for Growing Regression Trees

The training data for growing regression trees consist of 79 questions and their corresponding test-takers’ answers collected through writing tests. Each question has been developed based on general English grammar covered by the textbooks for third grade students of junior high school, which are adopted by the Ministry of Education in Korea. The test was performed at two junior high schools in Seoul and taken by approximately 360 students. We have obtained 345 training sentences for each question in average after discarding empty sentences. The training data were completed after all the test-takers’ inputs were scored by human raters. To build reliable training data, two trained human raters scored the test-takers’ sentences according to the scoring measurement presented in Fig. 3 and Table 3. When there was a discrepancy between those scores, the third rater determined the final result.

Each error count has been adopted as a set of fundamental features for growing a regression tree. For this research, we have used 82 features which are mapped to the error count for 82 individual errors. Each error was then classified into one of three categories—word error, syntactic error or mapping error, as described in Table 2. Due to insufficient training data, however, the scoring model showed a tendency to be highly skewed by the count of some specific errors when 82 features were used simultaneously. In order to stabilize the scoring model by removing the tendency, we have adopted the error count of each category as a set of features instead of the actual count of an individual

---

**Table 3** Basic scoring measurement.

| Score | number of word errors | number of syntactic errors | number of mapping errors |
|-------|-----------------------|---------------------------|------------------------|
| 2     | 0                     | 0                         | 0                      |
| 1     | 1 – 2                 | 1                         | 1 – 2                  |
| 0     | 3 or more             | 2 or more                 | 3 or more              |

---

1. In the current version, a sentence is regarded as incomplete and assigned with ‘incomplete sentence error’ in case the length of the sentence is half shorter than that of the shortest correct answer, or more than half of the words in the sentence are identified as having a spelling error.

2. They are Korean English teachers at junior high school, who have a teaching experience of 10 years or more.
error. With considering the teacher-provided scoring criteria as described in Sect. 3.2, we used 12 features shown in Table 4 to build a regression tree.

The error count was normalized by the number of words in a sentence. The features from (1) to (6) have a numeric value while those from (7) to (12) have a Boolean value, 0 or 1. The features from (7) to (12) were considered when the errors were counted using the features (2) ~ (6). Nevertheless, they have been selected as independent features since they influenced the final score to a fair extent.

4.2 Error-Weighting

As described in Sect. 1, the accuracy of the errors detected in the processing phases is low due to several reasons. Table 5 shows the accuracy of the errors detected by the system according to each processing phase. The accuracy of word errors is acceptable as the data in Table 5 suggest whereas the accuracies of both syntactic and mapping errors are so low. Approximately 38% of the errors did not coincide with the errors identified by human raters, which resulted in the final score with a low accuracy. A linear regression of error count has caused low accuracy of the final score.

In order to deal with this problem we measured error weights by adopting the idea of term weights which has been widely used in information retrieval. For a document, terms with higher weight are more important than those with lower weight to represent the document. Generally speaking, a term with higher frequency is awarded with a higher weight for the document because there is a correlation between the statistics of occurrence of the term and its importance for the document. In the same vein, we assign weight to each error found in input sentences for a specific question, depending on the number of occurrences of the error. So, when an error is frequently detected from the sentences for a specific question, higher weight is assigned to the error. Accordingly, the error can be considered to be the most typical error for the question, and it is assumed to be detected correctly. On the other hand, an error with lower weight can be regarded as incorrectly detected one. Therefore, we can determine that error weight would be a measure of how accurate the detected error is. By using error weight, we can obtain more reliable error count to build regression trees. We have reenumerated error count in consideration of error weight.

A collection of test-takers’ sentences for a specific question can be viewed as a document, and a specific error detected in the collection replaces a term. The basic information used in error-weighting is ‘error frequency’ and ‘question frequency’ borrowed from term frequency and document frequency, respectively. Error frequency is represented by $ef_{e,q}$, with the subscripts denoting the type of error and the question in order. From this processing, error frequency has been counted according to both a question and an error type as shown in Fig. 4. The error frequency $ef_{e,q}$ indicates the number of occurrences of error $e$ detected from the test-takers’ sentences for the question $q$.

When an error is detected from the sentences for almost every question, the error is considered too general to be meaningful for a specific question. In order to attenuate the effect of the errors that occur across too many questions, we have adopted the concept of inverse document frequency. We defined the inverse question frequency ($iqf$) of an error as follows:

$$iqf_e = \log \frac{N}{qf_e}$$

(1)

$N$ is the total number of questions, and $qf_e$ is the question frequency—the number of questions in which error $e$ occurs.

We then multiplied the value of error frequency by the value of inverse question frequency to calculate the weight of every single error for each question. Equation (2) calculates error weight for an error $e$ detected in question $q$.

$$ef \cdot iqf_{e,q} = ef_{e,q} \times iqf_e$$

(2)

In order to reenumerate the reliable error count in con-
sideration of the error weight, we have introduced a weighting function \( \alpha(\cdot) \), which maps an \( ef - iqf \) continuous value to a quantized weighting factor ranging from \( f_L \) to \( f_U \) by incrementing \( f_\Delta \). This function adjusts the error count to being more reliable. For example, when \( f_L \) is set to 0.5 and \( f_U \) to 1.5, the error count with the highest weight is multiplied by 1.5, while the count with the lowest weight is multiplied by 0.5. In other words, an error with the highest weight is counted as 1.5 while an error with the lowest weight is regarded less than a single count. We can interpret this as follows: an error with higher frequency in input sentences for a question is counted more because it is hardly incorrect, while an error that occurs infrequently in input sentences for a question is counted less because it can be incorrectly detected.

When a set of errors detected from a test-taker’s sentence for question \( q_s \) is \( E_t \), the reliability of the error count can be increased by applying a weighting function \( \alpha(\cdot) \) to the error count \( C(e) \) for error \( e \). The reliable error count in \( E_t \) is calculated as in Eq. (3), while the original error count before applying the function is a simple sum of \( C(e) \).

\[
\sum_{e \in E_t} \alpha(ef - iqf_{f_{\Delta}}) \times C(e) 
\]

We replaced the original error count with the reliable count calculated by Eq. (3) in order to form a set of features shown in Table 4, which was used to grow regression trees. In doing so, more accurate scores were expected to be generated.

5. Experimental Results

In this section, we evaluated three different automated scoring models—a scoring model based on teacher’s heuristics, a regression tree model, and a regression tree model with error-weighting.

5.1 Preliminary Evaluation

Agreement values between human scoring and automated scoring were introduced as the criteria on evaluating the system. As we mentioned in Sect. 4.1, all the sentences have already been scored by two human raters. Table 6 presents a summary of the experimental results. ‘Exact agreement’ [1] in the table counts only the perfect agreement between the scores of the two raters while ‘within one point agreement’ includes the cases in which the difference between the scores of the two raters is within one point or none.

The first row in Table 6 is the agreement result obtained by comparing the two human raters’ scoring outputs. The result is disappointing although the human raters have been trained to score test-takers’ sentences according to the same criteria. This becomes an obstacle to building a decent automatic scoring system. The second row in Table 6 shows the agreement result between the scoring model based on the heuristics described in Sect. 3.3 and human raters. Even though the scoring model has been implemented based on the same heuristics as what human raters used for scoring, there are some differences between the scoring results. One of the major reasons causing the scoring gap is that the teacher-provided criteria are different from what they have actually used for scoring. The criteria were not specific enough to be implemented in the system. Another reason is the low accuracy of detecting errors performed by the system because of the system’s incompleteness.

Another model that we have evaluated uses a regression tree. We used CART® to grow a regression tree. The features and the training data used in growing the tree are described in Sect. 4.1. In order to evaluate the scoring model using the regression trees we performed 10-fold cross validation. 90% of test-takers’ sentences were used to build training data, and the rest 10% of the sentences were tested for automatically producing scores using the regression trees. This validation process is repeated 10 times. Since a score value that CART® predicts is a continuous one, we have rounded-off this value to become an integer ranged from 0 to 6 when calculating an agreement value between a score generated by the regression model and a human rater’s score. The results are shown in the 3rd and 4th row in Table 6. The 3rd row is the result of the scoring model independent from the questions, which can grade any sentences regardless of the questions. As a result, one regression tree is created for this particular model. The 4th row is the result produced by the question-specific scoring model. A regression tree has been built for each question separately, which adds up to 79 regression trees in total.

The experimental results lead us to conclude that the regression tree model is far better than the heuristics model, and the question specific regression model is slightly better than the question independent one.

5.2 Evaluation Scoring Model Based on Regression Tree with Error-Weighting

The question-specific regression tree has been implemented as the base system in our evaluation. Error-weighting technique was then plugged into the model for performance improvement. In order to calculate error weights, error information was first defined. The error information reported by the system is represented with the three fields delimited by

| Table 6 | Comparison of heuristics-based model and regression model. |
|---------|----------------------------------------------------------|
| Scoring model | Exact agreement | Within one point agreement |
| Human rater1 vs. Human rater2 | 62.83 | 82.95 |
| Heuristics model | 57.48 | 83.15 |
| Regression tree (question-independent) | 65.47 | 90.03 |
| Regression tree (question-specific) | 66.12 | 91.55 |

†Throughout the experiments described in this paper, all CART® parameters are set to default.
Table 7 Examples of lexicalized and generalized errors.

| Original Error Information | Lexicalized Error Information | Generalized Error Information |
|----------------------------|-------------------------------|------------------------------|
| SUBJ_VERB_AGR_ERROR|buy|SUBJ_VERB_AGR_ERROR|buy|SUBJ_VERB_AGR_ERROR|
| OPTIONAL_NODE_MISSING_ERROR|more|OPTIONAL_NODE_MISSING_ERROR|more|OPTIONAL_NODE_MISSING_ERROR|
| FIRST_WORD_CASE_ERROR|buy|FIRST_WORD_CASE_ERROR|buy|FIRST_WORD_CASE_ERROR|

Table 8 Performance comparison in error-weighting.

| Scoring model | Exact agreement | Within one point agreement |
|----------------|-----------------|---------------------------|
| Base model (regression tree: question-specific) | 66.12 | 91.55 |
| Base model with error-weighting | 69.58 | 94.78 |
| Base model with error-weighting & smoothing | 67.87 | 93.04 |

The following is an example presented in Sect. 1.

SUBJ_VERB_AGR_ERROR | 7 | buy

The first field presents an error identifier, and the second indicates the error position occurred in a test-taker’s sentence. The third one is reserved for additional information such as a string to replace. The string in the 3rd field is one of the substrings in the answer sentence. Because an error position varies depending on a test-taker’s input, the second field is excluded from the error information. Finally, the error information is completed by appending the string in the 3rd field to an error identifier. The error identifier is thought to be lexicalized by a string to fix the error. Table 7 shows the examples of lexicalized errors.

All the sentences in the training data were analyzed by the automated scoring system, and then 3,741 lexicalized errors were obtained from the training data. The total of 9,073 $ef- iqf_{e,q}$ values has been derived from the training data when computed for each question. We recalculated the features from (2) to (6) among the feature set shown in Sect. 4.1 by using error weights along with Eq. (3), and then rebuilt regression trees for the training data. In the experiment, the values of $f_e$ and $f_{V_e}$ were set to 0.5 and 1.5, respectively, and $f_3$ to 0.2, as described in Sect. 4.2.

The final experimental results are presented in Table 8. ‘Base Model’ in this table refers to the scoring model using question-specific regression trees. The model with error-weighting has been improved by approximately 3%, compared to the base model. This proves that implementing the error-weighting technique increases the reliability of the error count, which, in turn, improves the accuracy of system-generated scores.

When the system encounters a new sentence which is not included in the training data, it is likely that $ef- iqf_{e,q}$ value of a newly detected error is not to be included in the training data. In other words, there are cases that the $ef- iqf_{e,q}$ value cannot be calculated for an error if the error is new to the training data. Smoothing $ef- iqf_{e,q}$ values makes it possible to estimate the weight for an unknown error. We implemented the back-off method to get the smoothed value of error weight as Eq. (4).

$$ef- iqf_{e,q} = \begin{cases} e f_{e,q} \times iqf_{e} & \text{if } C(e) \geq K \\ \omega \times e f_{e,q} \times iqf_{e} & \text{otherwise} \end{cases}$$  \tag{4}$$

In the equation (Eq. (4)), $\epsilon_e$ refers to the generalized information of error $e$, that is, error information without a lexicalized string. The examples of generalized error information are shown in Table 7. The value of $C(e)$ indicates the count of error $e$ occurred in test-taker’s sentences, and $K$ is set to 1. The value of $\omega$ denotes a normalizing factor and is set to 0.1 in this experiment.

The final row of Table 8 shows the result, in which the ‘exact agreement’ level degrades a little, but ‘within one point agreement’ still maintains over 93% of accuracy. In other words, when the automated scoring system produces a score, only 7 out of 100 sentences have more than 2 point differences from human rated scores. This leads us to conclude that the automated scoring system based on regression trees with error-weighting can be utilized as an aid tool to confirm human rater’s scoring.

5.3 Discussion

Although error-weighting can improve the accuracy of the system as a whole, it also has a side-effect. In example (EX1) in Table 9, the teacher scored 3 on the test-taker’s input while the regression tree without error-weighting generated 2 as the final score. However, the regression tree with error-weighting generated 3 points for the same sentence since the weight of the error ($E_{a2}$) was low, which hardly affected the error count. In example (EX3), the teacher’s score is 0 because its meaning is completely different from the intended meaning of the correct answer. On the other hand, the heuristics model generated 3 points for the same sentence. The scoring model using regression trees without error-weighting generated 4 points whereas the model with error-weighting scored 5.

The system was not able to analyze the string “the learned” correctly, which resulted in producing incongruent errors, ‘DET_UNMATCHED_ERROR(have learned)’ and ‘POS_UNMATCHED_ERROR(have learned)’.

These errors indicate that the string “the learned” should be changed into “have learned”, which is the substring of the answer sentence. Since these errors have hardly been committed for this question, their weights are relatively low compared to other errors. The scoring model ignores these two errors in enumerating reliable counts of errors, which results in the final score for this sentence to be higher than the score generated by the model without error-weighting. Both results, however, do not show a satisfactory performance, regardless of using error-weighting.
Furthermore, in this particular case, the scoring model without error-weighting has generated a score closer to the score granted by the teachers than the model with error-weighting as shown in (EX3).

The overall procedure of the automated scoring can be generalized as follows. Some of the inputs are initially scored by two human raters according to detailed scoring rubrics. If the scores differ by more than 1 point on a 0 to 6 point scale, a third rater adjudicates them. Once a sufficient number of inputs for each question are scored manually, an automated scoring model is built using those training inputs. When the model is formulated, the system replaces one of the two human raters. In some other cases, the system can replace all the human raters depending on the purpose of its applications.

The system, without human raters, checks if a new error is similar to any of the training data so that the trained scoring model can rate the error. If a new input sentence contains a heterogeneous error pattern, the system could fail in producing an accurate score because the scoring model has never been trained for the pattern. In such cases, the system could leave the new input to a human rater.

Error weight can be used as a measure for estimating the reliability of errors that the automated scoring system detects from a new input. If the weight of the errors is too low, the score of the input sentence has to be reviewed manually. Error weight in fact can be calculated without human-rated scores, which becomes useful when building applications of an automated scoring system. For example, when developing an interactive educational system to guide test-takers how to correct their sentences, the system would look more reliable if the error messages with higher weight could be suggested first, and those with lowest weight could be hidden.

6. Conclusion

This paper presented an error-weighting technique for building an automated scoring system that grades single English sentences. We built four scoring models and compared their performances. The scoring model based on the teacher-provided heuristics has presented the lowest accuracy among them. The main reason is due to the lack of mutual understanding between the teachers and the system designers. The scoring model based on regression trees has shown a better performance than the heuristics based model.

The accuracy of a score generated by the system tends to be influenced heavily by the accuracy of detecting errors. Correctly detecting errors is crucial, but there are several huddles to cross in order to reach close to 100% of accuracy in detecting errors. As a solution, error-weighting technique has been introduced and used to measure the reliability of the errors. We have proved the importance of error-weighting by improving the accuracy of the scores. Error-weighting technique is expected to be used in resolving other scoring problems in addition to scoring sentences, only if the system can collect error frequencies and question frequencies for each error and question.

References

[1] Y. Attali and J. Burstein, “Automated essay scoring with e-rater® V.2,” J. Technology, Learning, and Assessment, vol.4, no.3, pp.3–30, Feb. 2006.
[2] J. Burstein, M. Chodorow, and C. Leacock, “Automated essay evaluation: The criterion online writing service,” AI Magazine, vol.25, no.3, pp.27–36, 2004.
[3] J. Burstein and D. Higgins, “Advanced capabilities for evaluation student writing: Detecting off-topic essays without topic-specific training,” Proc. International Conference on Artificial Intelligence in Education, July 2005.
[4] J. Burstein, D. Marcu, and K. Knight, “Finding the WRITE stuff: Automatic identification of discourse structure in student essays,” IEEE Intelligent Systems, vol.18, no.1, pp.32–39, 2003.
[5] M. Chodorow and C. Leacock, “An unsupervised method for detecting grammatical errors,” Proc. First Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL -2000), pp.140–147, 2000.
[6] C. Fellbaum, WordNet An Electronic Lexical Database, The MIT Press, 1998.
[7] K. Gojenola and M. Oronoz, “Corpus-based syntactic error detection using syntactic patterns,” Proc. Workshop on Student Research, pp.24–29, 2000.
[8] D. Higgins, J. Burstein, D. Marcu, and C. Gentile, “Evaluating multiple aspects of coherence in student essays (PDF),” Proc. Annual Meeting of HLT/NAACL., 2004.
[9] K.J. Lee, Y. Choi, and J.E. Kim, “Building an automated English sentence evaluation system for student English as a second language,” Computer Speech and Language, 2010. (accepted)
[10] K. Papini, S. Roukos, T. Ward, and W-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” Proc. ACL 2002, 2002.
[11] L. Rudner, V. Garcia, and C. Welch, “An evaluation of the IntelliMetric™ essay scoring system,” J. Technology, Learning, and Assessment, vol.4, no.4, pp.3–21, 2006.
[12] D.A. Schneider and K.F. McCoy, “Recognizing syntactic errors in the writing of second language learners,” Proc. 17th International Conference on Computational Linguistics (COLING-AACL ’98), pp.1198–1204, 1998.
[13] M. Shermis and J. Burstein, Automated essay scoring: A cross-disciplinary perspective, Lawrence Erlbaum Associates, Hillsdale, N.J., 2003.
[14] M. Zhou, B. Wang, S. Liu, D. Zhang, and T. Zhao, “Diagnostic
evaluation of machine translation systems using automatically con-
structed linguistic check-points,” Proc. 22nd International Conference on Computational Linguistics, pp.1121–1128, 2008.

**Kong-Joo Lee** received the B.S. degree in Computer Science from the Sogang University, Korea, in 1992, the M.S. degree in 1994 and the Ph.D. degree in 1998 in Computer Science from KAIST. During 1998 through 2002 she was working for Microsoft, Korea. In 2003, she was a faculty member of the Department of Computer Science in Ewha Women’s University. In 2004, she was a faculty member of the School of Computer Information Technology in Kyungin Women’s College. Since 2005, she has been a faculty member of the Department of Information Communications Engineering in Chungnam National University. Her research interests include information retrieval, natural language parsing and machine translation.

**Jee-Eun Kim** received the B.S. degree in English Language and Linguistics from Hankuk University of Foreign Studies, Korea, in 1985, the M.S. degree in 1989 and the Ph.D. degree in 1993 at Linguistics, School of Language and Linguistics, Georgetown University, U.S.A. During 1995 through 2003 she was working for Microsoft. Since 2008, she has been a faculty member of the Dept. of English Linguistics, College of English, Hankuk University of Foreign Studies, Korea. Her research interests include information retrieval, natural language parsing and machine translation.