Listening. Learning. Leading.®

Research Report
ETS RR-12-23

A Preliminary Analysis of Keystroke Log Data From a Timed Writing Task

Russell Almond

Paul Deane

Thomas Quinlan

Michael Wagner

Tetyana Sydorenko

November 2012

ETS Research Report Series

Since its 1947 founding, ETS has conducted and disseminated scientific research to support its products and services, and to advance the measurement and education fields. In keeping with these goals, ETS is committed to making its research freely available to the professional community and to the general public. Published accounts of ETS research, including papers in the ETS Research Report series, undergo a formal peer-review process by ETS staff to ensure that they meet established scientific and professional standards. All such ETS-conducted peer reviews are in addition to any reviews that outside organizations may provide as part of their own publication processes.

The Daniel Eignor Editorship is named in honor of Dr. Daniel R. Eignor, who from 2001 until 2011 served the Research and Development division as Editor for the ETS Research Report series. The Eignor Editorship has been created to recognize the pivotal leadership role that Dr. Eignor played in the research publication process at ETS.

A Preliminary Analysis of Keystroke Log Data From a Timed Writing Task

Russell Almond, [1] Paul Deane, Thomas Quinlan,[2] and Michael Wagner
ETS, Princeton, New Jersey

Tetyana Sydorenko
Michigan State University, East Lansing

November 2012

As part of its nonprofit mission, ETS conducts and disseminates the results of research to advance quality and equity in education and assessment for the benefit of ETS's constituents and the field.

To obtain a PDF or a print copy of a report, please visit:

http://www.ets.org/research/contact.html

Associate Editor: Joel Tetreault

Technical Reviewers: Tenaha O'Reilly and Michael Flor

.

Abstract

The Fall 2007 and Spring 2008 pilot tests for the *CBAL*™ Writing assessment included experimental keystroke logging capabilities. This report documents the approaches used to capture the keystroke logs and the algorithms used to process the outputs. It also includes some preliminary findings based on the pilot data. In particular, it notes that the distribution of most of the pause length is consistent with data generated from a mixture of lognormal distributions. This corresponds to a cognitive model in which some pauses are merely part of the transcription (i.e., typing) process and some are part of more involved cognitive process (e.g., attention to writing conventions, word choice, and planning). In the pilot data, many of the features extracted from the keystroke logs were correlated with human scores. Due to the small sample sizes of the pilot studies, these findings are suggestive, not conclusive; however, they suggest a line of analysis for a large sample containing keystroke logging gathered in the fall of 2009.

Key words: writing assessment, writing fluency, automated scoring, timing, mixture models

Acknowledgments

The authors would like to thank Yigal Attali for suggestions about how to easily generate key logs and Joel Tetreault for assistance with identifying the amount of pasted text. The authors would also like to thank Michael Flor, Tenaha O'Reilly, and Joel Tetreault for editorial suggestions that improved the quality of the presentation.

Table of Contents

List of Tables

Page

List of Figures

Page

1. Overview

Among ideas for improving assessment, there is considerable consensus around the importance of focusing on critical thinking, as opposed to basic skills (e.g., Calfee & Miller, 2007; Shepard, 2006). Yet, developing such a test poses some major challenges. The application of literacy skills to critical thinking encompasses a large, complex construct. It is not surprising that many achievement tests have focused on basic skills, which tend to be easier to measure. However, in order to have a positive influence on instruction, future achievement tests should model the kinds of classroom lessons and activities that teachers value and that have shown to be effective. A major ETS initiative, Cognitively Based Assessment of, for, and as Learning aims to realize this ambitious goal (Bennett & Gitomer, 2009).

The *CBAL*™ writing (Deane, 2011, 2012; Deane, Fowles, Baldwin, & Persky, 2011; Deane, Quinlan, & Kostin, 2011) assessment features a variety of writing and literacy tasks, culminating in a short essay. The short essay is thought to be representative of other more complex writing tasks that students will face later in academic or professional life. However, a principle challenge with such constructed response tasks is scoring the student's performance.

Scoring, both human and automatic, has focused on the end product of the writing task, the completed (or partially completed) essay. Human raters are able to identify issues with both writing mechanics and critical thinking as expressed through the essay. Automated essay scoring algorithms are generally good at identifying mechanical issues and estimating writing fluency; they can then predict critical thinking scores through the correlation between fluency and critical thinking.

While scoring has focused on the final product, writing instruction usually emphasizes the writing process. Students are taught to organize their work through outlines and other tools, proofread and revise their drafts, and otherwise think about how they write as much as what they write. Simply looking at the final product does not reveal much about the process that was used to create it.

The use of computers for writing assessments introduces another possibility for capturing some of this process information: the series of computer events (key presses and mouse clicks) that the student uses to create the essay. This can be captured on standard computing equipment with only minor modifications to existing editing software. This key log could potentially reveal information about the student's writing process that is not readily apparent in the final essay.

That information could then be used either as formative feedback or to help predict the final grade assigned later by human raters.

This report looks at pilot keystroke logging data obtained as part of the pilot testing program for ETS's CBAL Writing assessment development effort, in which 79 eighth-grade students from three public schools in a mixed urban/suburban school district in Maine participated. Each particular CBAL writing assessment consists of some quantity of source material that the students must read, followed by some warm-up questions to get the students thinking about the material and to provide the assessors with information about how well they absorbed it. These are followed by an essay task. The essays were graded using a combination of human and computer scoring. Additionally, a keystroke logger was used for the essay tasks, providing data about which keys were pressed and the time spent between each keystroke.

To date, in the development of the CBAL writing assessment, two pilot tests have employed keystroke logging, one in the fall of 2007 (two forms) and the other in the spring of 2008 (one form).[3] All of these pilot studies were designed as small-scale studies with less than 100 students participating. During each of the pilots, technical issues arose with the keystroke logging, and the logging was turned off midway through the data collection. This still leaves us with sample sizes ranging between 20–80 students per form.

Although the number of students is not large, a large amount of data is collected for each student. A major challenge in making use of the keystroke logs is defining meaningful summaries that can then be input into scoring algorithms or factor analyses. Although conducted with relatively small groups of students, these data sets allow us to define potential summary variables and algorithms for extracting them from keystroke logs and to test the robustness of those algorithms. We can also see which of the variables have meaningful amounts of variation and, hence, might be candidates for use in large-scale studies.

Section 2 of this report supplies some background on the capture of the writing process. Section 3 talks about the procedures for collecting the keystroke logs, the format of the logs, and the processing and annotation of the logs. Section 4 shows the results of some exploratory data analyses using the pilot data. Finally, Section 5 makes some recommendations for which measures are promising for future study.

2. Background

One challenge in assessing writing lies in the inherent complexity of the construct: Any demonstration of critical thinking depends upon basic skills. For example, suppose a student has extensive knowledge about a topic (e.g., baseball) and can reason about it in complex ways; however, he may or may not be able to compose a well-formed essay on that topic. In this case, we would want to determine whether the student has adequate basic writing skills. In writing, the writer constructs a document, inscribing letters to make syllables, syllables to make words, words to make clauses and phrases, and so on. From a practical standpoint, this construction requires a certain level of fluency in basic writing skills. If a student is capable of only producing a few sentences per hour, she will likely have difficulties composing a whole document.

By *basic skills* in writing, we loosely mean the ability to put ideas into words and get them on the page. In terms of Hayes and Flower's (1980) model of writing competency, basic writing skills are encompassed by the translating process. Berninger and Swanson (1994) have posited two distinct subprocesses within translating: text generation, the process responsible for converting ideas into language representations, and transcription, which transforms those representations into written words. This distinction is helpful for understanding basic writing skills. Text generation largely refers to the mental manipulation of linguistic information (i.e., lexical and grammatical). In contrast, transcription largely refers to orthographic and graphomotor processes responsible for interacting with a writing tool. For example, pen and paper requires orthographic processes (retrieving word spellings and parsing them into letters) and graphomotor processes (for guiding the pen to draw letters). Notably, transcription is tool-specific, and to some extent differs from one tool (i.e., pen) to another (i.e., keyboard). In the latest version of the CBAL writing competency (Deane, 2012) model, text generation and transcription map to the Phrase and Inscribe processes, respectively.

Deane (2012) produced a more elaborate literacy competency model (Figure 1). This model recognizes that reading (Interpretation) and writing (Expression) are related and integrated processes, and that Reflection/Deliberation (which is part of the process of constructing an essay) is a combination of the two. For each of these three directions, Deane posed a series of layers of complexity ranging from simple (orthographic) to complex (social). During a pause we observe in the course of a student writing an essay, the student could be engaging in any of these

processes; however, higher level processes should, in general, generate longer pauses (Matsuhashi, 1981; Schilperoord, 2002).

A fundamental assumption of our analysis of the pause logs is that pauses at different places in the document should be generated by a different mixture of the literacy processes. A fluent middle-school writer should not be engaging in the higher level cognitive activities in the middle of typing a word. Thus, within-word pauses should be mostly generated by the inscribe (or transcribe) process and be relatively short. A student with many long pauses within words may be having difficulty with writing due to dysfluency in basic text production. In contrast, the pauses between larger units of text (i.e., sentences and paragraphs) should be generated by the higher level processes (e.g., structure and inquire) and, hence, they should be longer and more varied in length.

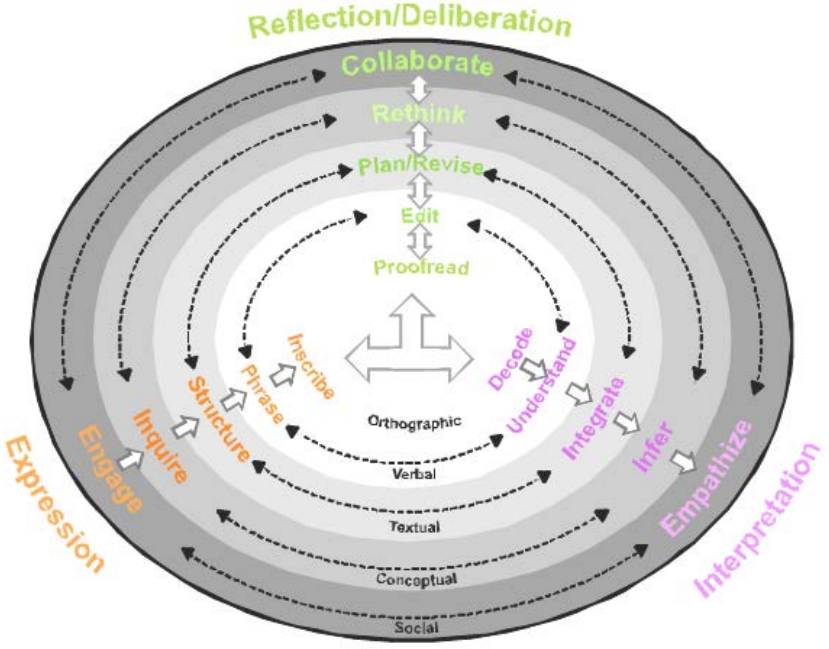

Figure 1. **Literary processes. From "Rethinking K-12 Writing Assessment," by P. Deane, 2012, in *Writing Assessment in the 21st Century: Essays in Honor of Edward M. White* by N. Elliot and L. Perelman (Eds.), pp. 87–100, New York, NY: Hampton Press. Copyright 2012 by Hampton Press. Reprinted with permission.**

It is important that children develop fluent basic writing skills. Educators may have a practical concern that children complete their schoolwork in a timely fashion. However, a greater concern is that dysfluent basic writing skills can hinder the development of critical literacy. There is strong evidence that problems either in text generation or transcription can interfere with other processes. For example, Bourdin and Fayol (1994; 2000) found that dysfluent handwriting in adults (artificially imposed) interfered with lexical retrieval. This result suggests that students with dysfluent handwriting (or typing) may have difficulty finding the right words. McCutchen and her colleagues (1994) found that more skilled writers, relative to less skilled writers, tend to be more fluent in (a) generating sentences (i.e., text generation) and (b) lexical retrieval. In writing research, there is considerable evidence suggesting that writing competency depends on developing a certain level of fluency in basic writing skills.

Accordingly, in order to assess children's skills in critical literacy, we must know something about their basic skills. This presents the challenge of assessing both critical thinking and basic skills. Ordinarily, a writing assessment might include a battery for assessing basic writing skills. For example, the Woodcock-Johnson Tests of Achievement (Woodcock & Johnson, 2001) include six different tests for measuring aspects of basic writing skill, such as sentence writing, spelling, editing, and punctuation. However, in developing a new writing assessment, this approach (adding separate measures of basic skills) is inadvisable. First, adding such measures would increase time spent on assessment, thus taking time away from other instruction. While we envision the development of richer assessment tasks, by which assessments can become learning experiences in their own right, these cannot take the place of good instruction. Second, adding measures of basic skills would tend to undermine the goal of keeping assessment squarely focused on eliciting critical thinking. Seemingly, explicitly measuring basic skills is not a desirable option.

As an alternative, we are investigating methods for passively measuring basic writing skills. CBAL assessments are computer-based, which has two important implications. First, since students will take the writing assessment on a computer, transcription will involve keyboarding. While keyboarding clearly involves different skills than handwriting, the research evidence suggests that keyboarding may afford some fluency benefits to struggling writers (cf. Cochran Smith, 1991). Second, computer-based testing presents the opportunity to capture a stream of information about student performance in a nonintrusive way. From information about students'

keyboarding, we might draw inferences about their relative fluency in producing words, sentences, and paragraphs.

There is strong rationale for analyzing keystroke data to understand student writing. Along with other cognitive researchers, writing researchers have used latency information as a measure of problem-solving. For example, experienced writers have shown to pause longer at major text junctures (i.e., paragraph and sentence boundaries), relative to minor text junctures (Matsuhashi, 1981; Schilperoord, 2002). With the move to computers, writing researchers have developed tools for capturing and analyzing keystroke data (Ahlsén & Strömqvist, 1999; Van Waes & Leijten, 2006).

We might expect to observe a different pattern of pausing for less skilled writers. In contrast to their more skilled peers, students with weaker basic writing skills may struggle more at the word level. If so, the location of the pause may reveal something about the nature of the struggle. Hypothetically, pauses between words may reflect the process of finding the right word (i.e., lexical retrieval and/or rehearsal); pauses within words may reflect typing speed (i.e., transcription), the process of spelling (i.e., orthographic processing), or perhaps even higher level processes (e.g., reconsidering word choice, planning, or the need to edit).

Finally, good writers do not produce text in a strictly linear fashion. They often will return and revisit previously written text to correct mistakes and to adjust the order in which information or arguments are presented. In a computer word processing environment, such revisions are usually made by moving the cursor to a new position, using either the mouse or keyboard navigation. Thus, the ability of the keystroke logs to track such jumps in editing location provides insight into the degree to which students are revising their documents versus trying to type them in a linear fashion.

3. Data Collection and Processing

CBAL is a computer-administered assessment. This means that it is relatively simple to capture additional information about timing of data entry as part of the assessment. However, the raw stream of information is very large, and it needs to be processed in various ways. In particular, we want to be able to classify the various events in the log according to what kind of processing the student was doing at the time. To this end, we want to classify each entry in the timing log into one of six possible states:

- *InWord*—Writer is pausing between letters (or other symbols) within a word.
- *(Between) Word*—Writer is pausing between words.
- *(Between) Sentence*—Writer is pausing between sentences.
- *(Between) Paragraph*—Writer is pausing between paragraphs.
- *BackSpace*—Writer is pausing before beginning an editing operation, either a single backspace or multiple backspaces (i.e., several backspaces in a row) events.
- *Edit*—Writer is pausing before cut, paste, or replace operations or before using a mouse to navigate to a different part of the essay.

This section lays out the basics of the data capture and classification process. Section 3.1 describes the CBAL tasks and the environment in which the essay task is performed. Section 3.2 describes the keystroke logger and the information captured in this analysis. Section 3.3 describes the first phase of processing: a character classifier that classifies individual keystrokes according to their linguistic purpose. Section 3.4 describes the second phase of processing: a classifier that attaches one of the categories above to each event in the event log.

3.1 CBAL Writing Tasks

The goal of the CBAL writing assessment is not to just have students write isolated essays, but to create writing tasks that more closely resemble authentic writing. In particular, all of the CBAL writing tasks contain source material, which the student is expected to read and write about. The genre of the essay and the purpose of the writing vary slightly from form to form. Each form contains a main essay task, for which the student is given 45 minutes to complete. The form also contains warm-up tasks before the main essay (and some forms also contain a post-essay task). These tasks are related to the genre, purpose, and source material of the main essay and generally collect both selected and short constructed responses. The key logging facility was only included in the central long essay task. The assessment was delivered on school-issued laptops that the students used throughout the school year, so it is reasonable to suppose that the students were familiar with the keyboard feel and layout.

The Spring 2008 assessment, on which the bulk of the analyses below are based, had an intended genre of expository writing. Students were given several source documents providing information about issues related to starting the school day later for high school students. The

intent was that students would produce a written synthesis of the material. Many of the students instead took and argued a position. Also, the students had the ability to cut and paste excerpts from the source documents. Consequently, a large fraction of many documents was actually pasted from the source material, sometimes with attribution, sometimes without (this resulted in logs that were short in comparison with the length of the text). One essay consisted entirely of quoted material.

The Spring 2008 essays were scored by the teachers, who received minimal training. Consequently, the interrater reliability of the scoring was moderately low (weighted kappa values of .63, .78, and .60 for the three strands—sentence level mechanics, organization and development, and critical thinking—respectively). Although plans were made to rescore the essays, the rescoring was dropped in favor of work on new forms and planning for the large-scale pilot test in Fall 2009.

The Fall 2007 administration was also an early pilot, although this time the students were randomly assigned to one of two forms. The structure of the forms was similar (one large essay task and several smaller tasks using the same material). As with the Spring 2008 task, these were the first pilot tests of the forms, so students' lack of understanding of the task goals contributed an additional source of variability to the observed outcomes.

3.2 Log Capture

Although the CBAL assessments contain a variety of tasks of varying lengths (both selected and constructed responses), we captured keystroke logging data only for the long essay. The initial CBAL pilot test was conducted in the fall of 2007, and two different writing forms were used during that pilot. Based on the experiences with that pilot, an additional pilot administration was given in the spring of 2008. In both pilot tests, the amount of data returned by the keystroke logging system challenged the capacity of the computer networks in the schools participating in the pilot. Consequently, the keystroke logger was disabled partway through each pilot, and keystroke logs are only available for a subset of the students. However, there is no reason to believe that this subset is not representative of the larger population, or that the logger functioned improperly for the subjects who had it turned on.

In a modern computer system, every time a key is pressed or a mouse is clicked, the computer generates an event. As this event is associated with the low-level physical hardware of the computer, it is known as a physical event. The computer passes this event to the application

software that currently has focus. The application software translates the physical event into a logical event that makes sense in the context of the application. For example, a word processor might translate a key press physical event into a character insertion logical event, while a game might translate the same physical event into a different logical event, such as the movement of the player's avatar on the screen. Although the physical events contain a complete record of the interaction, they are often difficult to interpret. Since the goal of analyzing the keystroke logs is to gain insight into the writer's cognitive process, the logical events are a better reflection of the user's thinking.

Despite the fact that logical events are better for our purposes, physical events are easier to capture. The first attempt at the logger, based on the Fall 2007 data, was based on physical events. For each event, it logged which key was pressed and at what time. This format proved problematic for several reasons. An important hole in the data capture was that mouse events were not captured. As writers normally change the position of the cursor in the document using the mouse, this means that it is impossible to tell whether a given burst of typing (i.e., typing not interrupted by pause events) is appending text to the end of the essay or revising/inserting text in the middle of the essay. The lack of mouse events means that cut and paste events were not captured. Even so, the initial key log data reveal some interesting patterns that are worth investigating in follow-up studies.

For the Spring 2008 administration, the keystroke logger was revised along the lines suggested by Yigal Attali (personal communication, November 1, 2007). For technical reasons, it turned out to be difficult to capture logical events. Instead, the logger worked by comparing the current and revised version of the document every 5 milliseconds. If differences were found, then an event would be logged. The logged information consisted of the time of the event, the position (in characters from the beginning of the document) of the change, and the added (new) and removed (old) text. For an insertion event, the old text would be empty; for a deletion event, the new text would be empty; and for a replacement event, both the old and new text would have value.

Using differences between the document versions at short time intervals to generate key logs presents some technical challenges. One problem is that it is possible for a fast typist to type two or even three characters within the 5 milliseconds window of time. For example, entries such as *th* or even ␣*of* can be found (where ␣ represents a space typed on the keyboard). However, by

setting a reasonable threshold for length of the inserted string, we can distinguish between events that are likely typing activity and those that are likely cut-and-paste operations.

3.3 Character Classification

If the goal is to classify the typing events as to whether or not the student is within a word, we must first identify which characters are parts of a word. Obviously, we would like letters to be parts of words. It also makes sense for numbers to constitute words. Furthermore, certain punctuation marks (such as hyphens and apostrophes) appear within compound words and contractions and should be treated as word parts for the purposes of the classifier.

A preliminary step to classification then is to classify the characters. The Portable Operating System Interface (POSIX) standard for extended regular expressions (ISO/IEC 9945-2:1993; International Organization for Standardization, 1993) provides a starting point for this work. The POSIX standard defines character classes [:upper:], [:lower:], [:digit:], [:punct:], and [:space:], which provide a starting point for our own classification system. Working with the POSIX classes has two advantages: (a) they are supported by commonly available programming languages such as Perl, Java, and R (R Development Core Team, 2009), and (b) the Institute of Electrical and Electronics Engineers (IEEE) and International Organization for Standardization (ISO) committees that worked on them have already put some thought into internationalization.[4]

Although the [:upper:], [:lower:], and [:digit:] classifications work well out of the box, the POSIX [:punct:] class contains characters that can be used for multiple purposes within a document. Three punctuation marks (period, exclamation point, and question mark) are used to terminate sentences (ellipses, which usually consist of multiple periods, are also handled by this rule). We will call these characters *EOS* (end of sentence) punctuation. The hyphen and the apostrophe can appear as part of compound words and contractions, so we will call these *inwordpunct*. We will use the term *punct* to refer to any POSIX punctuation character not covered by the above rules, such as a colon, a semicolon, or a comma.

Similarly, the POSIX [:space:] classification is too broad. We potentially need to distinguish between horizontal whitespace (spaces, tabs) that separates words and vertical whitespace (carriage return, linefeed, vertical tab) that separates paragraphs. We call the latter *EOL* (end of line; carriage return, line feed, or
 tag in HTML document) or *EOP* (end of paragraph; vertical tab or <p> tag in HTML document). The term *WS* is then used for all other kinds of whitespace, such as a tab or a space. There is also a POSIX [:cntrl:] category for

nonprinting control sequences. These should not appear in the logs, except for the backspace or delete characters, which are mapped onto a special *bs* category.

The present implementation of the character classifier only considers the current event; this is a source of potential misclassification. For example, the period at the end of the abbreviation *Dr.* should be considered ordinary usually (word-terminating, but not sentence-terminating) punctuation (punct), and the periods within an abbreviation such as *P.T.O.* should be considered inwordpunct ordinarily. This is a difficult problem, which is complicated by the fact that the partial document may have uncorrected typographical errors making it more difficult to infer the writer's intent. (For example, one student typed *don"t* [*sic*]. In this case, the quotes—ordinary punctuation—were almost certainly a typographical error, and the student probably intended to type an apostrophe, in-word punctuation). Additionally, strings of punctuation may change the classification. For example, multiple hyphens may be shorthand for an em-dash or en-dash, and should be interpreted as ordinary punct and not inwordpunct. Similarly, various combinations of punctuation designed to produce emoticons (smilies) may be used by student writers to terminate sentences.[5]

These matters are not considered in the current implementation because they would require taking an additional pass through the event log to properly classify them. It is hoped that the number of misclassifications due to ignoring these compound cases is small.

The current classification system works by taking two passes through the event log. The first pass reads the event log from the XML file (Appendix A), translating HTML markup into the corresponding characters (Appendix A.4). Character class annotations are added at this time, and the result is stored as a vector of KeyLogEvent objects (Appendix B). The second pass (Section 3.4) does the work of actually classifying the event types.

3.4 Event Classification

The event classification works by assuming that InWord*;* (Between) Word, Sentence, Paragraph, BackSpace, and Edit are states that the writer could be in. The classifier is then a finite state machine that works through the event stream classifying the events according to what state it is currently in.

Implementing the state machine requires producing formal definitions for the states. We also add special BEGIN, END, and Error states for a complete description of the algorithm. For analyses of the Spring 2008 data, we used the following definitions:

- *BEGIN*—When the essay part of the assessment starts.
- *END*—When the log has finished.[6]
- *InWord*—Before the student has typed a word constituent character (upper, lower, digit, or inwordpunct).
- *Word*—Before the student has typed punctuation or whitespace between words within a sentence.
- *Sentence*—Before the student has typed EOS punctuation or whitespace between sentences.
- *Paragraph*—Before the student has typed vertical whitespace (EOL or similar characters).
- *Edit*—Before the student jumped to a new position in the document or performed an operation involving a large amount of text (replace operation or cut/paste operation involving three or more characters).
- *BackSpace*—Before the student deleted a small amount of text (one to three characters).
- *Error*—Temporary state if unexpected character (e.g., control character) is found in log. The goal is to flag unusual data and try to resynchronize the classifier and classify as much of the document as is possible.

Table 1 shows the kind of events that will move the state machine from one state type to another. The rows indicate the current state of the state machine, and the columns indicate the new state, which is also the classification assigned to the event. For the most part, the properties of the current event determine the value of the state assigned to the event; however, whitespace events depend on context (between word or sentence).

Table 1

State Machine Transitions

From/to	Edit	BackSpace	InWord	Word	Sentence	Paragraph	END
BEGIN	~~Jump~~, Cut, Paste, Replace	Delete	Wordchar EOS Punct			EOP, EOL, WS	OK or Timeout
InWord	Jump, Cut, Paste, Replace	Delete	Wordchar[a]	WS, Punct	EOS	EOP, EOL	OK or Timeout
Word	Jump, Cut, Paste, Replace	Delete	Wordchar[a]	WS, punct	EOS	EOP, EOL	OK or Timeout
Sentence	Jump, Cut, Paste, Replace	Delete	Wordchar[a]		WS, punct, EOS	EOP, EOL	OK or Timeout
Paragraph	Jump, Cut, Paste, Replace	Delete	Wordchar, EOS, punct[a]			EOP, EOL, WS	OK or Timeout
BackSpace	Jump, Cut, Paste, Replace	Delete	Wordchar[b]	WS & inSentence, punct	WS & inPara, EOS	WS & !inSentence, !inPara, EOP, EOL	OK or Timeout
Edit	Jump, Cut, Paste, Replace (new event)	Delete	Wordchar[c]	WS, punct	EOS	EOP, EOL	OK or Timeout

Note. Exclamation points preceding a flag = the condition should be negated, EOL = end of line, EOP = end of paragraph, EOS = end of sentence, inPara = flag indicating that writer is typing within a paragraph, inSentence = flag indicating writer is typing within a sentence, punct = punctuation, Wordchar = number, letter or in-word punctuation, WS = whitespace.

[a]Time is logged with the previous event (row name). [b]Time is logged to between words, between sentences, or between paragraphs, depending on value of flags. [c]This requires special handling, as the time should be logged to between word time, not InWord time.

Several decisions were made to aid in the classification. First, it is necessary to distinguish between events in which characters are added through typing (or removed with the backspace/delete key) and those in which larger chunks of text are added through the clipboard (cut and paste). Based on inspection of the first few keystroke logs, we set three characters as a threshold. Additions of more than three new characters are classified as *Paste* events, while additions of three or fewer new characters are classified as *Insert* events (and the processing depends on the characters inserted). Similarly, removal of three or fewer old characters is counted as *Delete* events, and removal of more than three is categorized as *Cut* events. Any event with both old and new characters is counted as a *Replace* event. *Jump* events are determined by a change in the position that cannot be accounted for by effects of the previous event, as that indicates the user has moved the cursor. For Insert events, the type of event is determined by the character category of the last character in the new string (this is appropriate as we are generally concerned with the state that the writer is in after completing the operation).

There is also a question of priority of the rules. The highest priority is given to the rules that transition to the Edit state. This is indicated either by a jump (a change in cursor position not caused by the previous event) or by any Cut, Paste, or Replace event. The second highest priority is given to the Delete event, which will always put the state machine in the BackSpace state. And then the priority is given to the transitions to InWord, Word, Sentence, and Paragraph states in that order.

The transition out of the BackSpace state takes some care, as it requires knowing where the user was before starting the backspacing. Generally speaking, the character after the backspace should be treated as if it were in the state that the system was previously in. To that end, the state machine keeps track of three flags, inWord, inSentence, and inParagraph, which are used to indicate whether the writer is currently typing within a sentence or a paragraph. When the BackSpace event is followed by WS, the value of the last two flags is used to indicate whether the transition should be to the Word, Sentence, or Paragraph states.

Another critical issue is that the first character of a new word, sentence or paragraph (usually a letter, but new paragraphs could start with punctuation) should really be grouped with between Word, Sentence, or Paragraph time. In particular, when the state machine transitions into the InWord state, for the most part it associates the event with the previous state of the state machine. The BackSpace and Edit states are two exceptions to the general rule. For the

BackSpace event, the transition is determined by the values of the InWord, InSentence, and InParagraph flags. For Edit events, the following typing is always credited to Word time.

The classification produced by the state machine is not perfect and does not necessarily correspond to the classification we would get by looking at the final document. The Words, Sentences and Paragraphs identified by the state machine are not necessarily the same as the ones in the final document. In particular, they are chunks of text punctuated by editing operations (jump, cut, paste, replace). Later editing operation could modify the unit to provide its final appearance.

- A Word could contain embedded spaces if they are typed quickly enough; for example, the logger might identify *of the* as a single word if it was typed very quickly.
- A Sentence or Paragraph could be interrupted by an edit operation. Thus, what would appear as a single paragraph in the document may be multiple units in the log file.

Even so, the definitions provided here should be good enough for a preliminary analysis. Based on what appears to be useful, we can later refine the definitions.

The classifier does two things. First, it assigns a type (one of InWord, Word, Sentence, Paragraph, BackSpace, or Edit) to each event in the event log. Second, for each event type, it builds a list of pause times for each category. For InWord and Edit pauses, this is simply the pause before the current event. For Word, Sentence, and Paragraph events, the pauses for the punctuation and whitespace at the end of the unit are added to the pause length, as is the pause before the start of the next unit (Word, Sentence, or Paragraph). For a single BackSpace, the pause time before each backspace is recorded. For a multiple BackSpace, the pause times before each backspace are summed up. Pauses are only placed in one of the categories (i.e., Sentence pauses are not included in the list of Word pauses). In addition to the collection of pauses, the state machine also identifies Bursts sequences of events not classified as edit events, in which the inter-event time is less than two-thirds[7] of a second. In other words, a burst is uninterrupted typing that does not include pauses longer than two-thirds of a second and is uninterrupted by cut, copy, paste events or the use of the mouse to jump to a new location in the document. A burst can include an event like BackSpace if it is shorter than two-thirds of a second. The state machine records the length of each burst, that is, the number of events (this is close to the

number of characters; however, if some events contain multiple characters, this will be slightly less than the number of characters).

The parser also attempts to distinguish between append events, which occur at the end to the currently produced text, and edit events, which occur at some point other than the end. The classification algorithm is not perfect and can get fooled by problems in the data, including problems with the algorithm that adjusts document length for HTML codes. In particular, if the partial document contains terminal whitespace (e.g., a blank line at the end), and the student inserts text just before that terminal whitespace, the algorithm will classify this as an edit event, even though the student's intentions are closer to the meaning of append. If these measures show promise, then a more sophisticated algorithm will be needed. Thus, in addition to the pause sequences, the parser calculates the following values:

- *appendTime*—text production time (sum of InWord, Word, Sentence, and Paragraph pauses)
- *editTime*—revision/editing time (sum of Edit and BackSpace pauses)
- *startTime*—the timestamp of the first event, planning/reading time (before typing or pasting)
- *totalTime*—total writing time (appendTime + editTime)

The Spring 2008 data were classified using a collection of specially designed functions written in the R language (R Development Core Team, 2009). The processed logs were stored in R objects as described in Appendix B. The Fall 2007 data were classified using a Perl script that did not produce an annotated log, only the lists of timing data. As the data formats for the logger differed between the two systems, the definitions of the pause types were slightly different between the two groups. However, the intended definition was similar, so analyses with both data sets are discussed in the following section.

The classifier itself contained a logging system that showed details of how each event was classified. This was hand-checked for a few selected logs. In addition, the log was consulted when the classifier generated an error. This latter testing identified a few places where the classification algorithm needed refinement.

4. Preliminary Analysis

4.1 Qualitative Observations

There were 79 essays that were collected with the keystroke logging turned on; however, three of those essays had empty logs. As an initial pass, we eliminated 11 essays with fewer than 50 log entries, resulting in a final data set of 68 essays and logs.

At least part of the difficulty was the presence of a planning tool that the students could use for a variety of purposes, including writing draft text. In many cases, the initial event was a rather large paste, and, in at least one case, it contained an obvious typographical error (quote substituted for an apostrophe), indicating that it was a paste of student text. Unfortunately, events in the planning tool are not logged, so it is difficult to obtain insight into the students' cognition during that time.

A technical problem with the treatment of quotation marks required us to examine the use of quotations by hand. A fair number of students used quotations from the source material, in particular, from persons quoted in the source material, to support their argument. This is partially due to the nature of the prompt, which called for them to support a position on whether or not the school day should start later. Students selected actors in the story who offered arguments for one side or the other of the issue. They often used quotation marks when quoting these persons.

To assess the extent to which quotation from the source material was present in student essays, Joel Tetreault ran the essays through his essay similarity detection software (Tetrault & Chodorow, 2009). The amount of overlap ranged from 0 to 100%. The use of quoted material cut and pasted from the source material likely has an effect on automated scoring algorithms. In particular, the Grammar, Usage, Mechanics, and Style features of the *e-rater*® system (Attali & Burstein, 2006) are all based on error counts normalized by the document length. Adding copied error-free text from the source material adds to the denominator without adding to the numerator, thus yielding more favorable values for those features. Similarly, the quoted material adds to either the number of discourse units or to the average length of a discourse unit, inflating the organization and development features. In regression models used to predict human scores from linguistic features, the percentage of overlap came in with a negative coefficient, suggesting that it was correcting for otherwise inflated feature values.

Finally, the students knew that they would not be graded on this experimental assessment. The length or content of several of the essays indicated that some students did not make a serious effort at addressing the required task.

4.2 Shape of Pause Distributions

The classifier distinguished between seven different types of pause distributions, as well as the bursts. Running the classification algorithm resulted in eight different data vectors for the 68 students in the reduced sample. The length of these data vectors varied from person to person. Some indications of the distributions of the lengths are given in Table 2.

One thing that is striking from Table 2 is how few paragraphs are appearing in the final essay. More than half of the students are turning in single-paragraph essays (zero between-paragraph breaks identified) and three-quarters of the students are providing at most two paragraphs. It is possible that there are issues with the classifier missing paragraph boundaries embedded in paste events or multicharacter sequences; however, this seems to be borne out by an examination of the essays. Note that the 316 paragraphs are from a log that had a large number of stray blank lines added to the end, and that data point is probably a problem with the capture tool. The numbers in the parentheses give the value of the mean and maximum with that outlier removed.

Table 2

Distribution of Lengths of Pause Data Vectors

	Bursts	InWord	(Between) Word	(Between) Sentence	(Between) Paragraph	(Single) BackSpace	(Multiple) BackSpace	Edit
Min	6.00	18.0	11.0	0.0	0.0	0.00	0.00	0.0
1st Q	53.50	234.2	88.5	4.0	0.0	7.75	8.75	23.0
Median	98.50	451.5	163.5	11.0	0.0	18.00	20.00	39.5
Mean	104.00	557.0	187.8	11.3	5.4 (0.78)	23.00	26.56	51.8
3rd Q	145.25	812.0	280.8	18.0	1.0	35.00	40.25	74.25
Max	325.00	1,747.0	479.0	28.0	316.0 (5.0)	97.00	112.00	192.0

Note. This table shows the counts of the number of events of each type, as well as the number of bursts. Q = quartile.

To look at the distributional shapes of the eight pause distributions (including bursts), we produced a set of box plots for each measure, showing the distribution for that pause type for that student. Initial looks at the data on the natural scale showed that the data were too highly skewed to show details of the distribution. Therefore, we produced boxplots of the log of the pause times. For burst lengths, which are a measure of text production fluency, we also investigated square roots, but found that even the square root distribution was highly skewed. Figures 2 and 3 show the distributions of the (natural) log of the pause times.

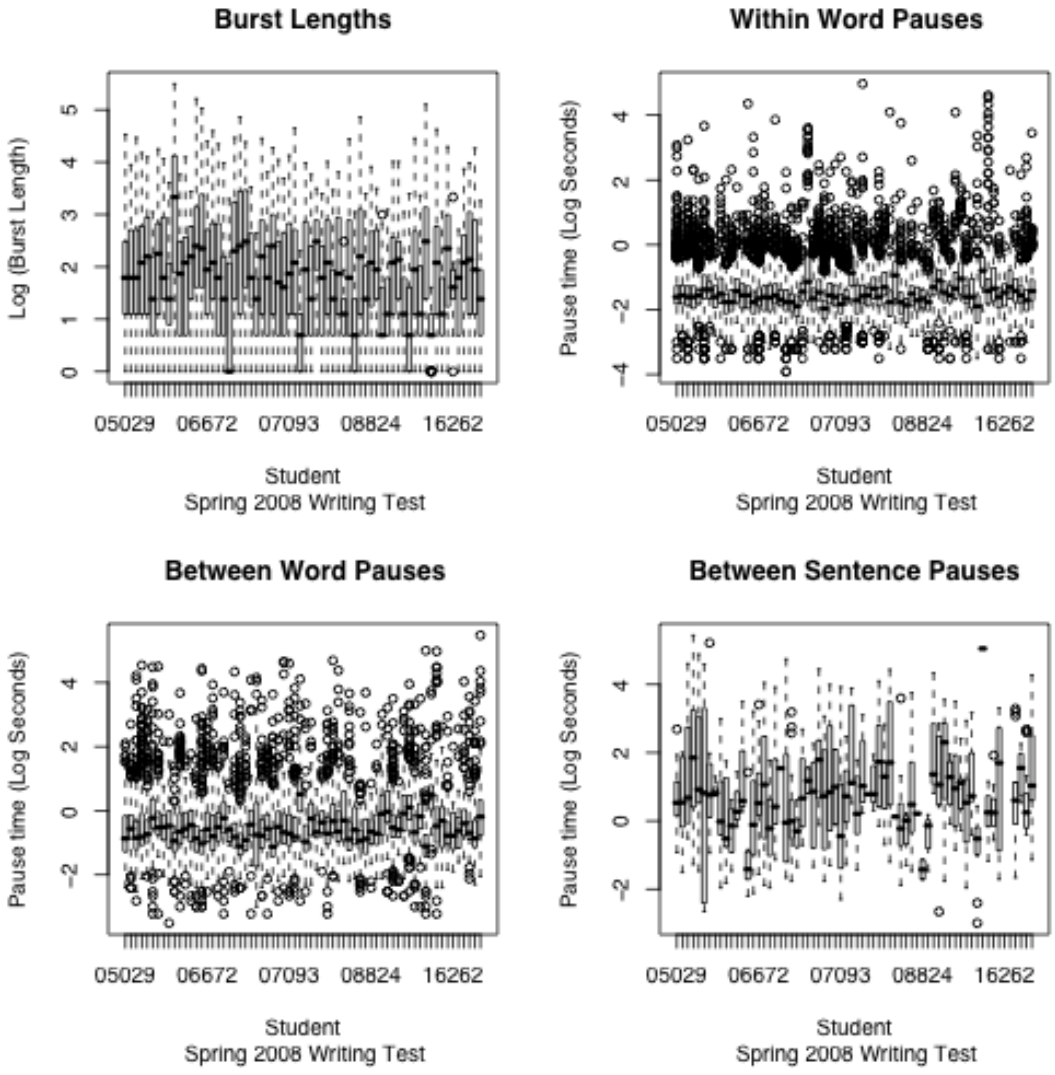

Figure 2. **Box plots of log pause lengths within words, between words, and between sentences (or log burst length).**

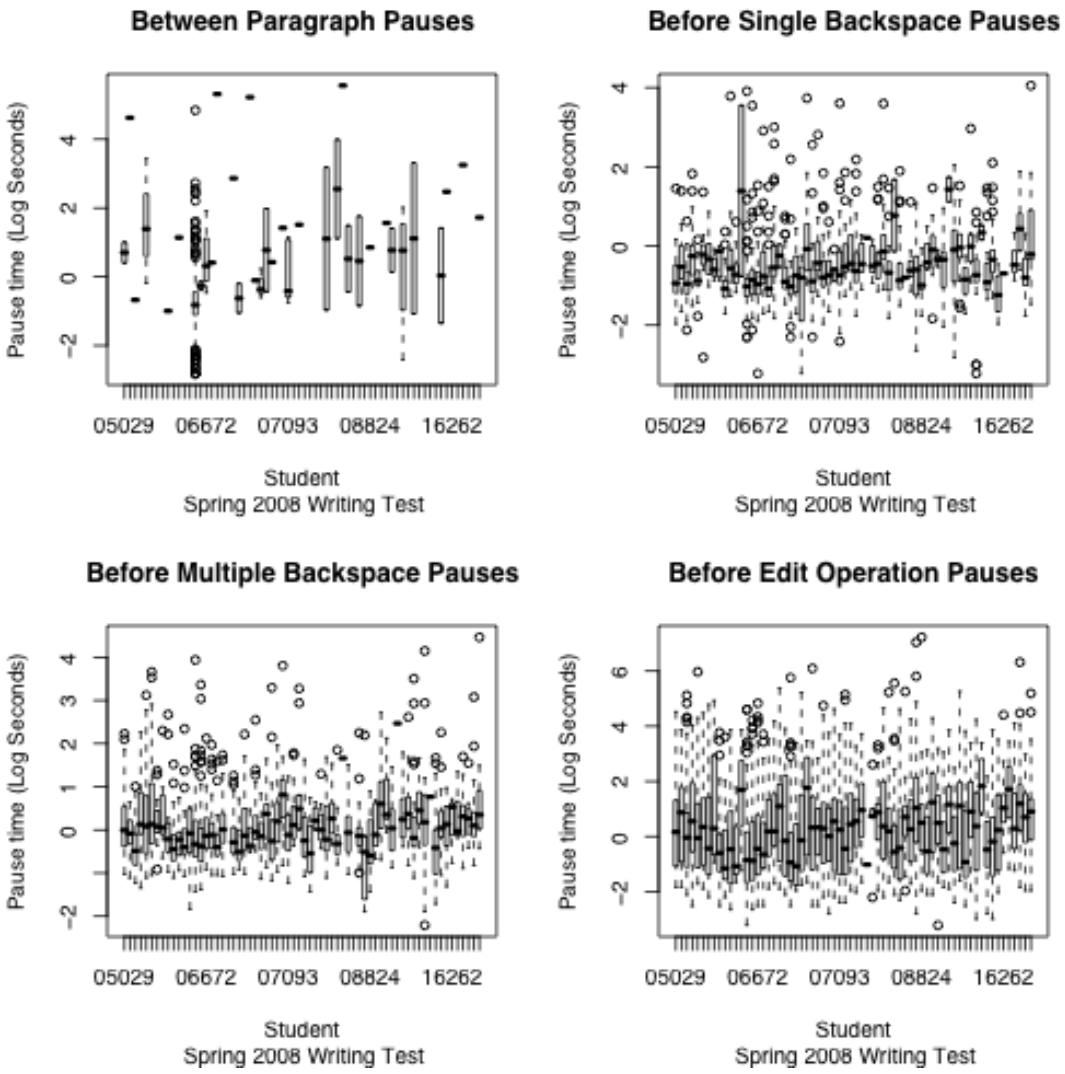

Figure 3. **Box plots of log pause lengths for between paragraphs, before a single backspace, before multiple backspaces, and before edit operations.**

All of the plots show a large number of outliers (plotted as individual circles). This is especially true in the InWord and (Between) Word pauses. For the other pause types, the number of pauses occurring in a typical essay is much smaller, and hence the outlier detection is much weaker. This indicates that the pause distributions are highly leptokurtic (much heavier tails than the normal distribution). To confirm this, Figures 4 and 5 look at density plots for the log pause times for the first nine keystroke logs. Because of the lower number of events of the other five types, the density plots for those distribution types were judged to be too likely to be showing artifacts of the particular observed data, rather than general underlying trend.

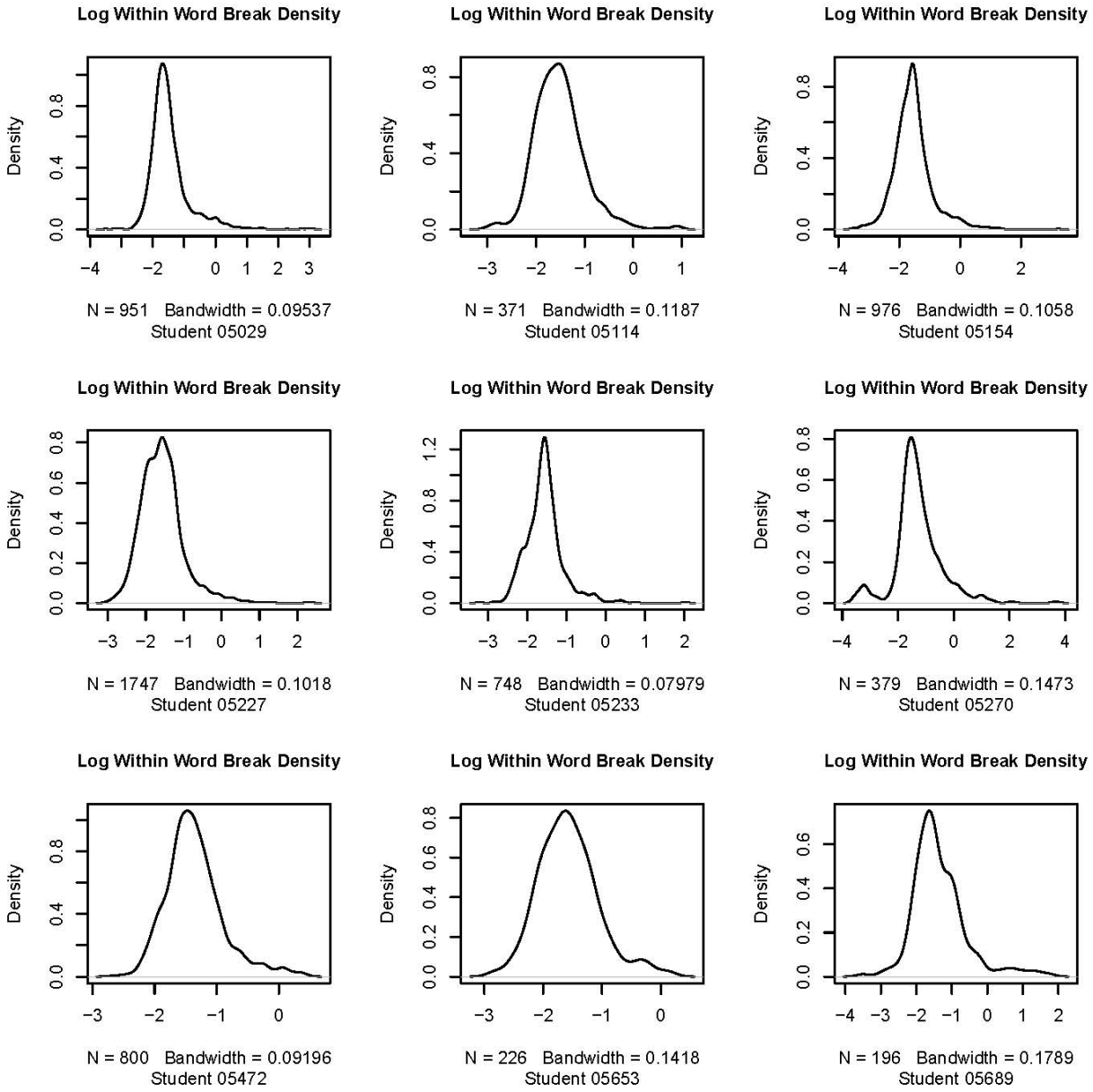

Figure 4. **Density plots of within-word break time for first nine student logs.**

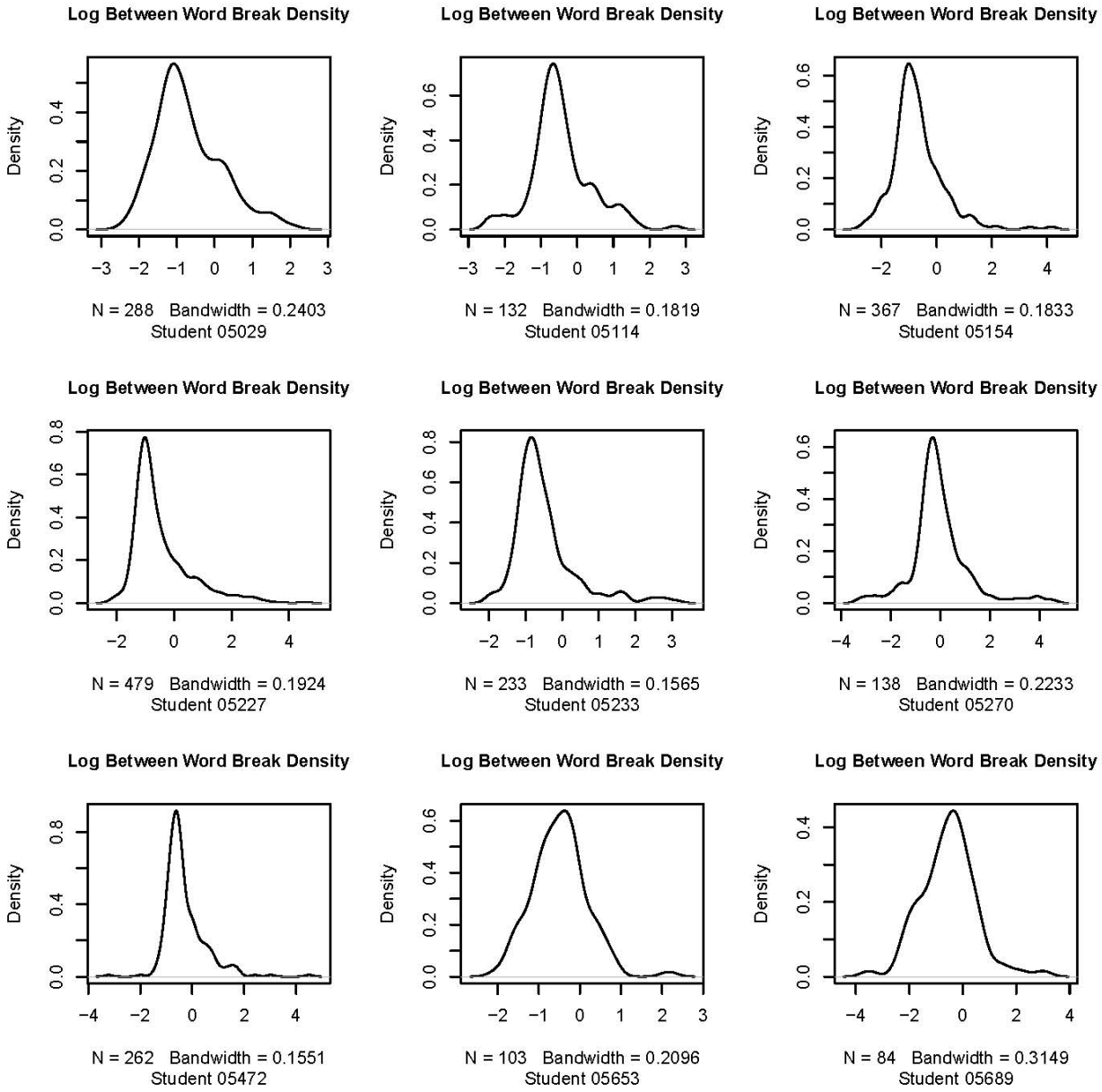

Figure 5. **Density plots of between-word pauses for first nine student essays.**

Note that in both Figures 4 and 5, the distributions are sharply peaked, another sign of a leptokurtic distribution. There are also some indications that the distributions might be bimodal.

One kind of distribution type that could generate such heavy tails is the mixture of normals or, in this case, a mixture of lognormals. This distribution type corresponds to an interesting cognitive hypothesis about how the pauses are generated. There is one pause type (with the shorter average duration) that corresponds to the mechanics of typing; there is a second pause type (with the longer average duration) in which the writer is attending to other deeper cognitive processes. This includes mechanics related processes, such as spelling and punctuation, and deeper word choices, such as organization, word choice, and critical thinking.

Figure 6 shows how this works. The dashed line is a standard normal distribution (representing the pure typing log pauses). The dotted line is a normal distribution with mean 1 and standard deviation 4. These are scaled by the mixing weights (.7 for the standard normal and .3 for the larger distribution). In the tails, the mixed distribution takes the shape of the second component. In the middle, we have contributions from both components.

To test if the mixture model hypothesis was a plausible explanation for the observed kurtosis, we generated random samples of size 500 from a mixture of normal distributions. For each data set, we generated 500 random draws from a standard normal distribution, 500 draws from a normal distribution with mean 1 and standard deviation 4, and 500 uniform random numbers. If the uniform random number was less than the mixing parameter (set at 0.5, 0.3, and 0.7 for different samples), we selected the data point from the standard normal, and if it was above, we selected the data point from the alternative distribution. Figure 7 shows the results. The kurtosis for these samples is similar to what is observed in the InWord and Word pause density plots.

To gain more insight into the mixture model hypothesis, we used the *flexmix* software package (Grün & Leisch, 2008) to fit one-, two-, and three-component mixture models to the log pause data for InWord, Word, and Bursts (the number of events per student for the other event types is too small for the mixture modeling estimation). This package uses the EM algorithm (Dempster, Laird, & Rubin, 1977) to estimate the mean and variance of each mixture component, as well as estimating the probability that each data point comes from each cluster. The EM algorithm is an iterative algorithm that has good theoretical convergence results; however, it is

not guaranteed to converge within a finite series of iterations. Also, it is possible for two or more of the mixture components to have sufficiently close mean and variance that the algorithm will combine them into a single component; this results in a fitted model with fewer than the requested number of components.

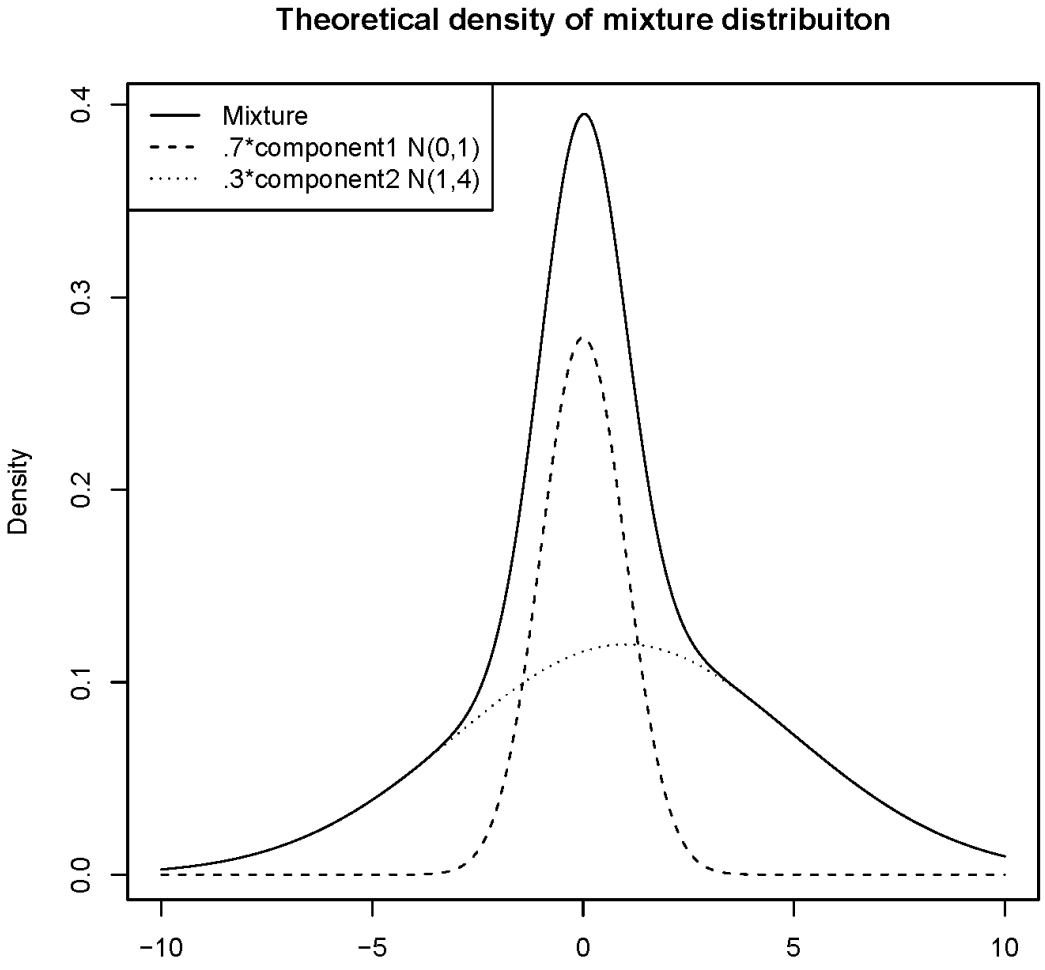

Figure 6. **Theoretical model of a mixture distribution.**

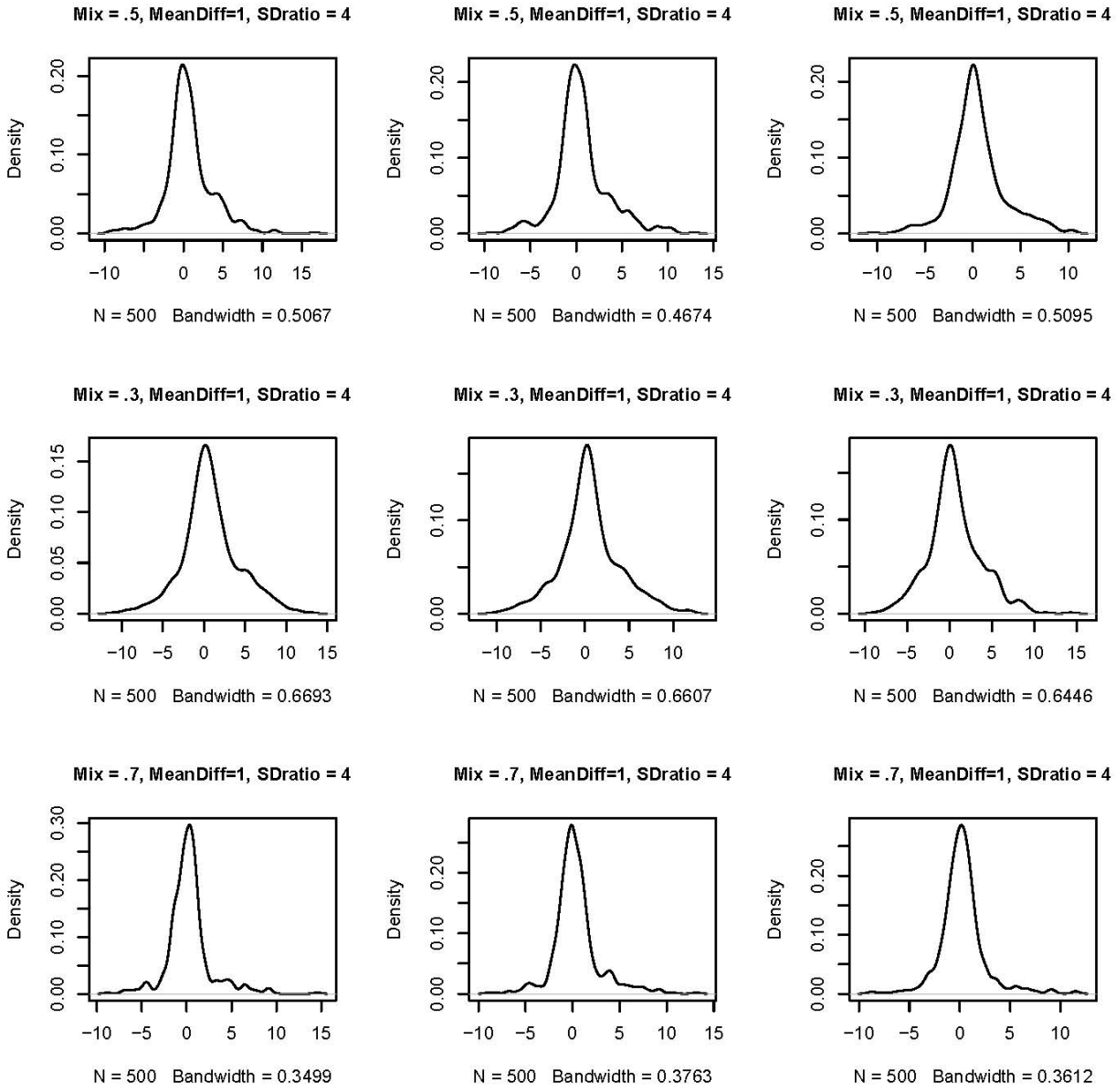

Figure 7. **Density plots for nine random data sets generated from mixture distributions.**

The flexmix model–fitting software offers two different statistics useful for model comparison: Akaike's information criterion (AIC) and Bayesian information criterion (BIC). Both statistics are similar in that they use the log likelihood to measure the fit of the model to the data and add a penalty for extra parameters (as adding more parameters should improve the fit). They differ mainly in the weight given to the penalty term (BIC has a somewhat higher penalty for model complexity). Usually both statistics should favor the same model, and cases where the simpler model is favored by BIC and not AIC are probably too close to call.

Looking first at the InWord data, the two-component model converged for 58 out of 68 student essays, and seven of the 58 converged to a single component solution. Of the 51 cases for which the two-component solution converged, the BIC statistic favored the simpler model in only four cases, and the AIC statistic never favored the one-component model. The three-component model converged for 48 out of 68 essays (seven of the 10 cases that did not converge for the two-component model also did not converge for the three-component model), and 15 of the 48 models converged to a one- or two-component solution. Of the 33 remaining cases, the AIC statistic favored the simpler two-component model in 21 cases, and the BIC statistic favored the simpler model in 28 cases. These findings seem to indicate that in most cases, the two-component model provides an adequate description of the data.

Looking next at the Word data, the two-component model converged in 65 cases, but in five of those cases converged to the simpler one-component model. Of the remaining 60 cases, the AIC statistic favored the simpler model five times, and the BIC statistic favored the simpler model 12 times. The three-component model converged in 52 cases, but in 16 of those cases, it converged to a one- or two-component solution. Of the remaining 36 cases, the AIC statistic favored the simpler model in 19 of those cases, and the BIC statistic favored the simpler model in 31 of those cases. Again, it seems as if the two-component solution provides an adequate description of the data.

One of the more interesting parameters of the mixture model is the mixing parameter, which indicated the fraction of each data set that comes from the first component (the one with the shortest average time). Figure 8 shows the distribution of this statistic for students for whom the two-component model converged to a two-component solution. Note that the first quartile for both distributions is above 0.5, indicating that for most students more than half of the pauses are from the smaller distribution.

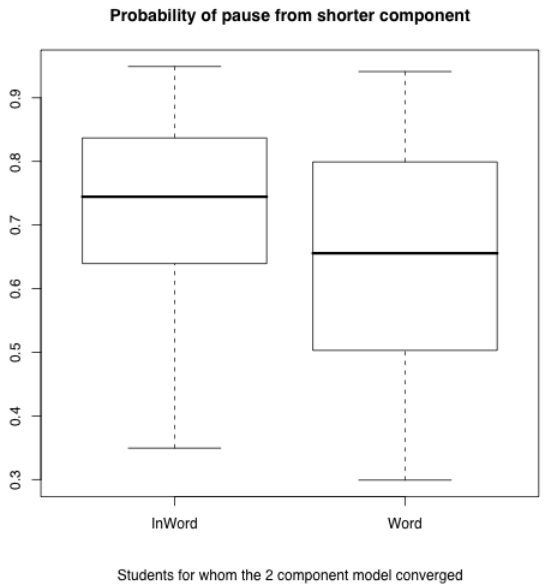

Figure 8. **Box plot of the mixing parameter.**

This study is far from conclusive. There are still a number of cases, especially in the shorter data sets, where the simpler model is favored. It also could well be that one of the components should be split into multiple groups. In particular, long pauses, where the writer is attending to issues of mechanics and word choice, and very long pauses, where the writer is attending to issues of material selection and organization, could be separate cognitive processes. However, there might be only two or three of those kinds of events in the shorter writing tasks, and thus the third component might not be identifiable from the data we have.

We carried out a similar distributional analysis with the log of the burst length data, eliminating two logs that contained fewer than 10 bursts. Figure 9 shows the density plots for the first nine logs. The two-component model converged for all but two of the logs, but for 18 of the 64 logs it converged to a one-component solution. Of the remaining 46 logs, the AIC statistic favored the one-component model in 32 cases, and the BIC statistic favored the simpler model in 41 cases. It is likely that what appears to be multiple modes in the plots in Figure 9 is an artifact of the relatively small sample size.

The sample sizes for the other types of pauses were not large enough for us to be comfortable with the mixture modeling. One possible solution is to try and build a hierarchical model defining a population distribution for the parameters of the two components across students. A simpler solution is to assign each of the pauses to one of the two mixture components

based on an arbitrary cut score. One possibility is to use the same two-thirds of a second cut point used in defining bursts. Another is to use a person-dependent cut point, such as the median Word pause for splitting the InWord sample and the median Sentence pause for splitting the Word sample.

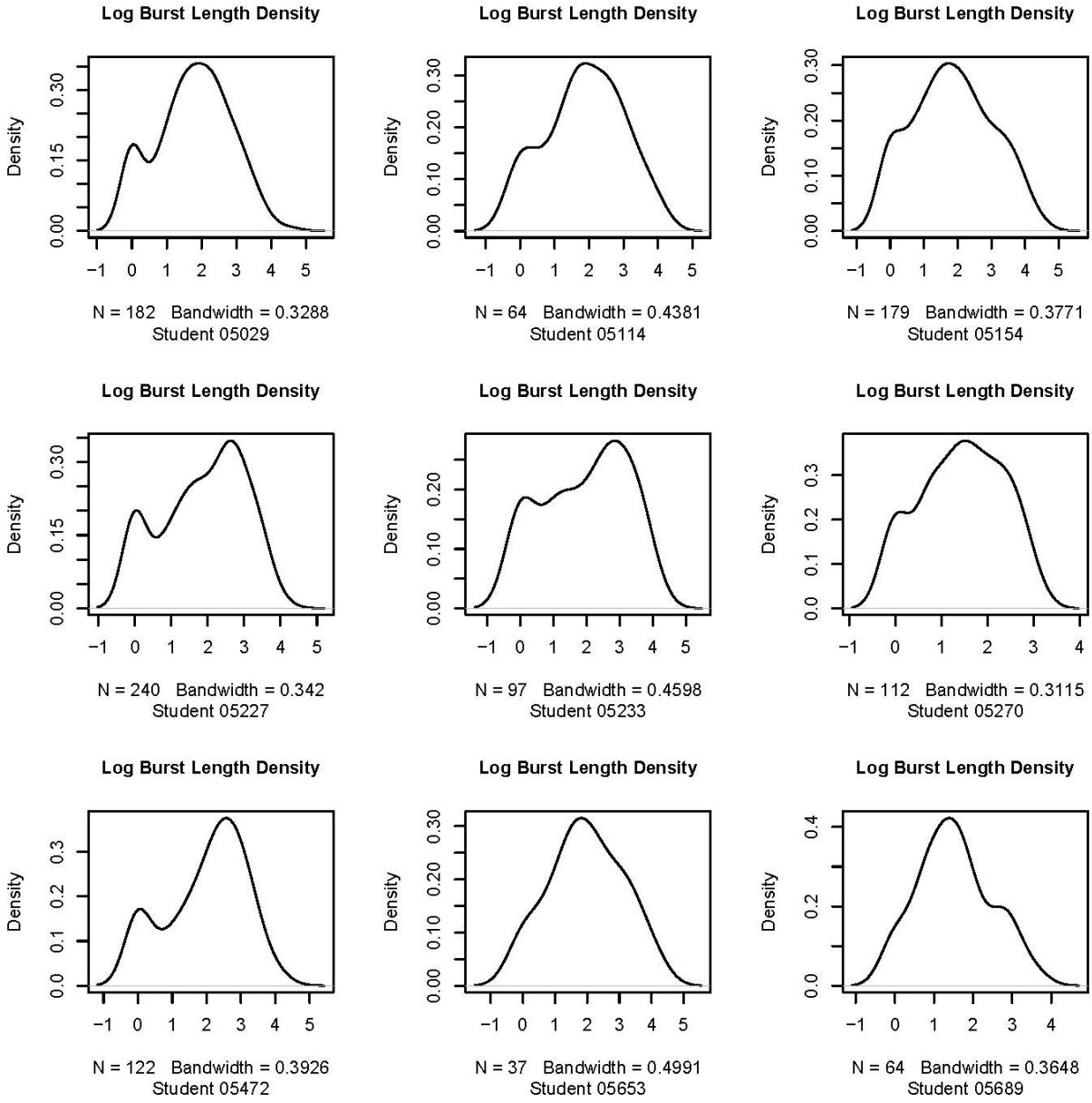

Figure 9. **Densities of log burst length.**

An even simpler solution is simply to look at the mean and variance of the pause time data (on the log scale). Note that if the two-component mixture model is a reasonable explanation for how the data are generated, then the sample mean is a function of three parameters: the means of the two components and the proportion of pauses that are drawn from each component. The standard deviation is a function of all five parameters; in particular, it is strongly influenced by both the standard deviation of the second component and the separation between the two means. However, these simple statistics may provide adequate summaries of the pause data without the convergence issues that arise with the more complex models.

Finally, it is worth noting that a similar analysis was performed using the Fall 2007 data set. Although software used to capture and classify the pauses was different, the distributional shapes were similar. In particular, the mixture of lognormal models described above seemed to fit the pause time distributions for the InWord and Word pauses.

4.3 Correlations Between Key Log Features and Human Scores

The keystroke logging features and the human essay scores are measuring different parts of the writing construct. In particular, the human scorers are looking exclusively at the product of the writing, while the keystroke logs record part of the process of the writing. Nevertheless, we expect that there should be a relationship between the two measures: students who exhibit signs of a better (more fluent) writing process should produce better essays. This section explores the relationships between various summaries of the keystroke log data and the human scores.

The Spring 2008 and Fall 2007 data were both scored using rubrics drawn from the three-strand CBAL Writing Competency. Strand I is based on the students' control of sentence-level construct: grammar, usage, and mechanics. Strand II is based on the student's control of document-level constructs: organization and development. Strand III is related to the degree to which the student displays critical thinking in the essay. Figure 10 shows the correlations among the strands. (Note that these correlations are lower than the official correlations given in the CBAL data analysis report because the data set only includes essays that had at least 50 log entries, eliminating all of the essays receiving a score of zero.) The correlation between strands on this assessment was lower than for some of the other assessments because the raters were teachers who only had time for minimal training before scoring the essays.

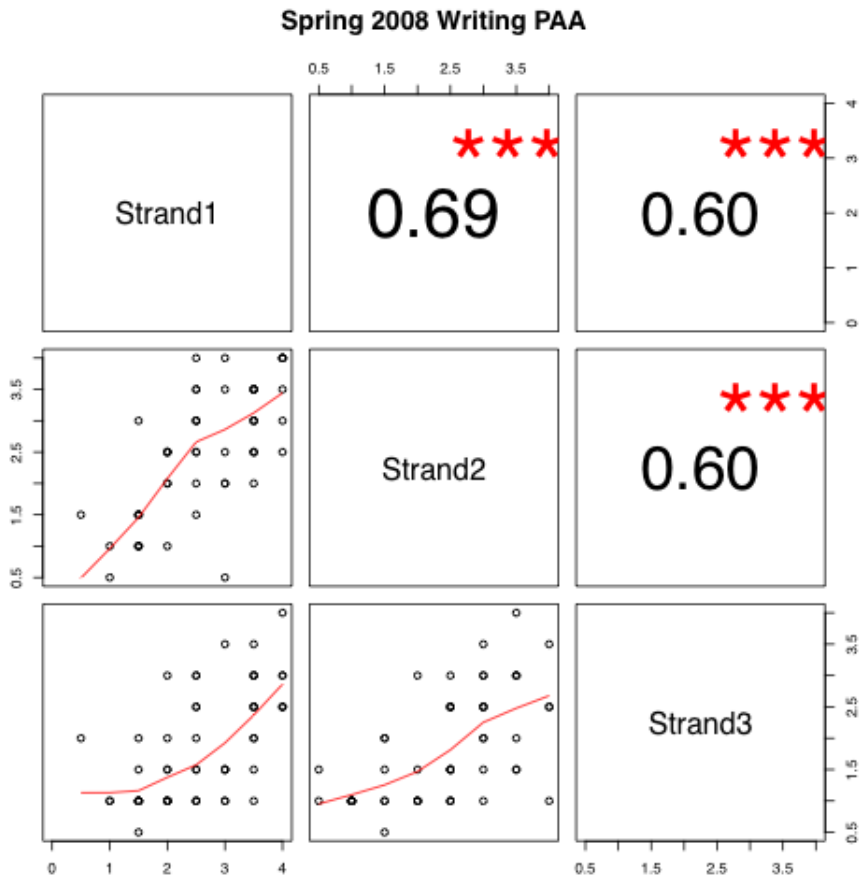

Figure 10. **Correlations among score strands. PAA = periodic accountability assessment.**

First, we will explore the relationship between the various timing features and the scores by looking at the correlations between the three-strand scores and the various timing features. Although both the small sample size and lack of rater reliability lowers our confidence generalizing the observed relationships to larger data sets, this exploratory analysis should indicate relationships that are worth studying when more data becomes available.

Figure 11 looks at three total-time summaries of the key log. The appendTime variable represents the total amount of time that the student spent appending text to the end of the document. The editTime variable represents the total amount of time the student spent performing editing operations or appending text somewhere other than the end of the document. Neither of these features shows a strong correlation with the strand scores. This may be at least partially due to an issue with how the events are classified: in particular, if there was text after the insertion point that was whitespace, this would be classified as an edit event, not an Append

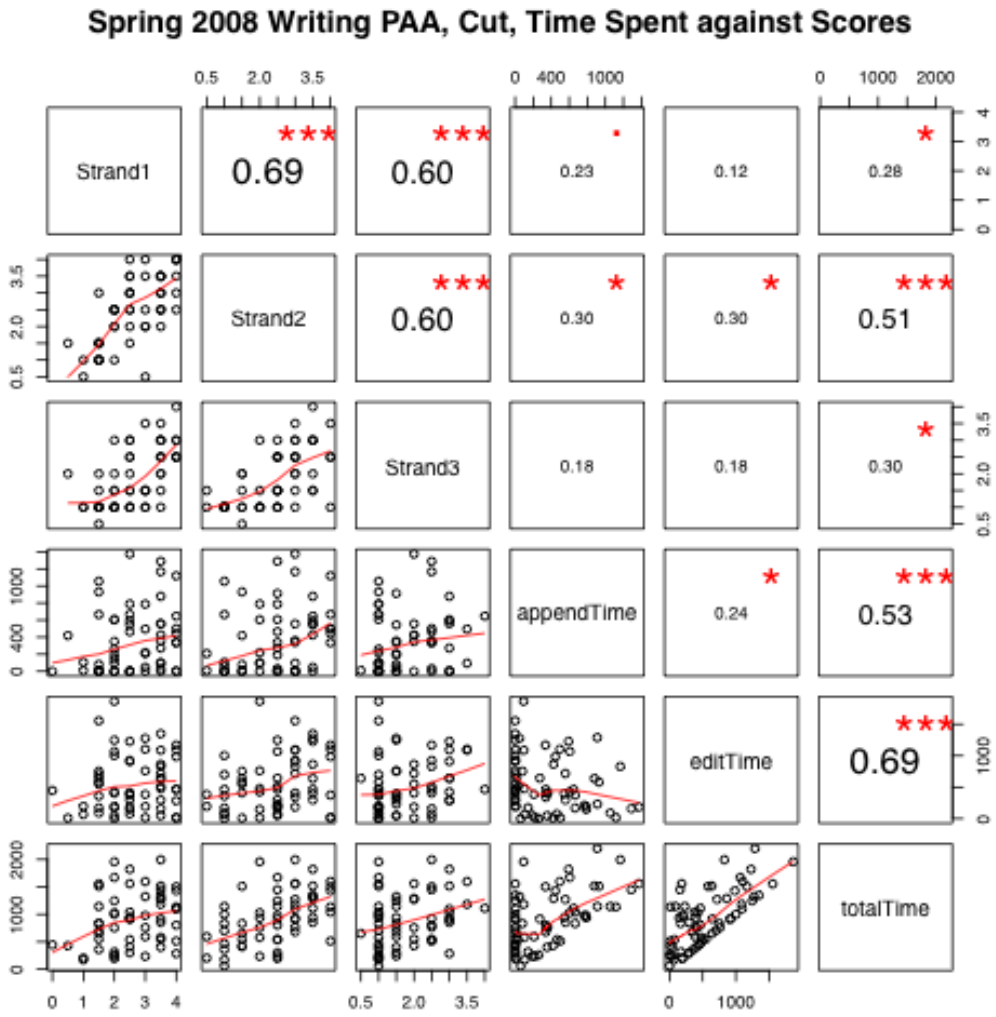

Figure 11. **Correlations of strand scores with time spent. PAA = periodic accountability assessment.**

event, even though it was logically appended to the end of the document. Note that the total time does show a moderate correlation with Strand II, Organization and Development. This indicates that students who spend more time tended to produce more developed (i.e., longer) documents. Figure 12 shows the relationship between the strand scores and two statistics related to the length of the bursts (a series of typing events with less than two-thirds of a second pause between them). Because the distribution is highly skewed, we worked with the log of the burst lengths. The two chosen statistics are the mean (on the log scale) and the standard deviation (on the log scale). The latter should separate people who have both long and short bursts of typing from

people whose bursts are relatively uniform. Note that the two measures are highly correlated. There is also a general correlation between burst length and scores on all of the strands. This is likely a function of a general fluency effect: students who can produce text in long bursts generally have better control over sentence- and document-level features and can spend more time on critical thinking (Strand III). Note that the correlation between Strand III and mean burst length is lower than the other correlations. Note also that the correlation is slightly higher for the relationship between Strand I (sentence-level control) and mean burst length, and Strand II (document-level control) and the variety of burst lengths. It will be interesting to see if this relationship persists with better data.

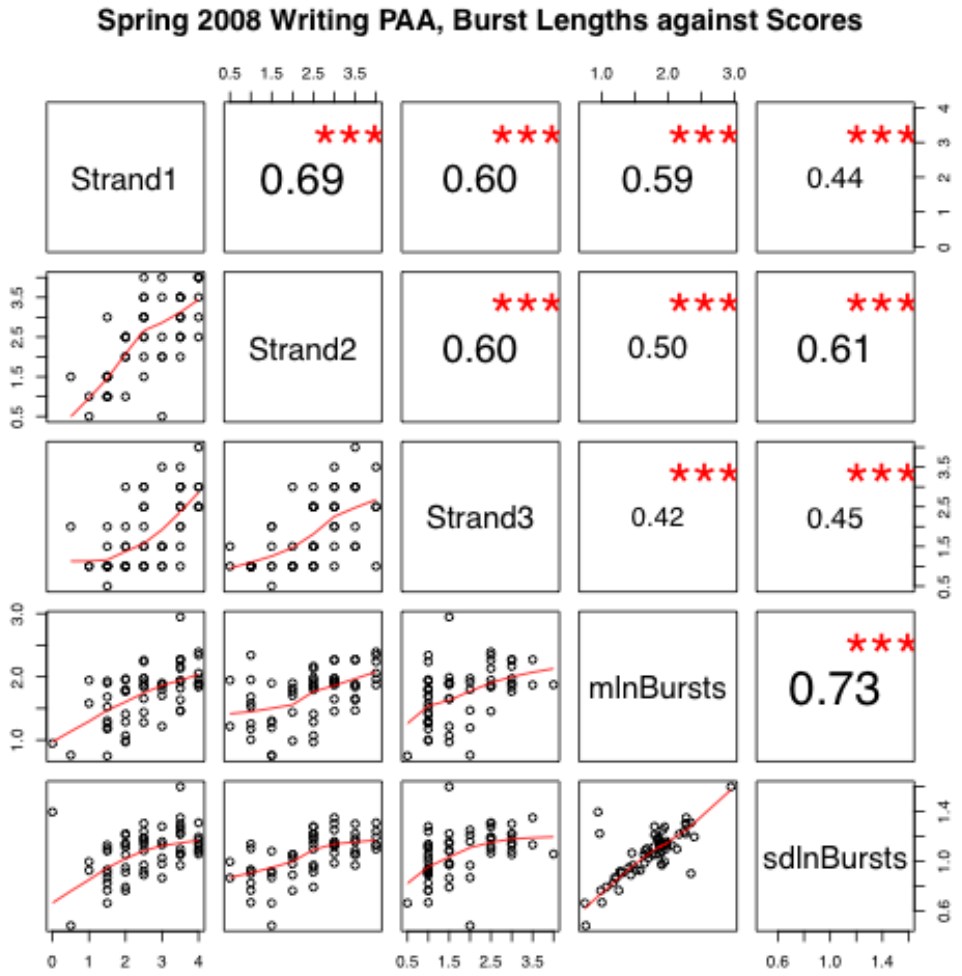

Figure 12. **Relationship between strand scores and bursts. PAA = periodic accountability assessment.**

Of the various pause types, the majority of pauses are within word pauses. Figure 13 shows the relationship between the average and standard deviation of log pause length and the human scores. There is a moderately strong negative correlation between the average within-word time and the various strand scores. This is not surprising, because this probably is a measure of typing speed and text production fluency. The relationship between the strand scores and the variance of the within-word pauses does not appear to be strong. There are two outlying cases (for whom the standard deviation of log pause length is above one); however, removing those outliers does not substantially change the correlations.

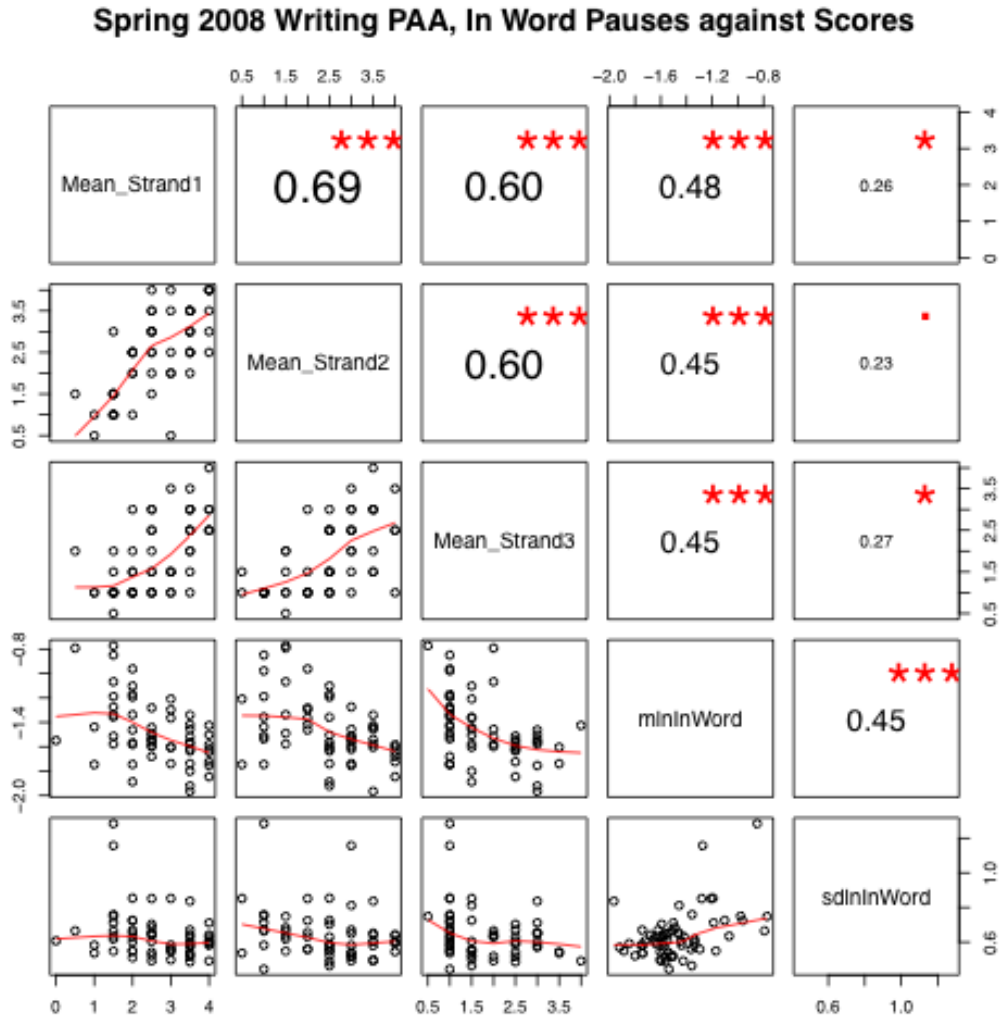

Figure 13. **Relationship between strand scores and within-word pauses. PAA = periodic accountability assessment.**

Figure 14 shows the relationship between the strand scores and the between-word pauses. The story is similar, indicating that this may again be a general fluency measure. Again there are two outliers in the standard deviation measure, which are the same two essays that were flagged for the within-word pauses.

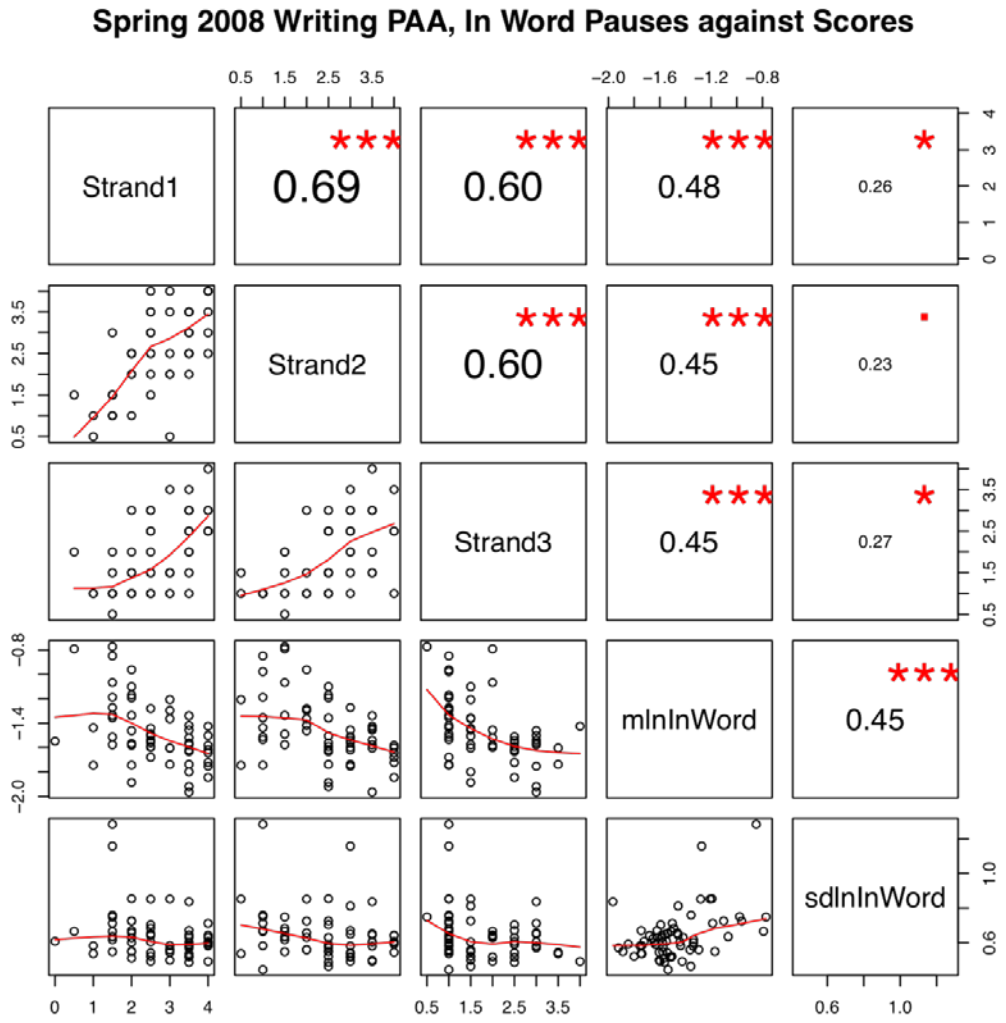

Figure 14. **Relationship between strand scores and between-word pauses. PAA = periodic accountability assessment.**

Between-sentence and between-paragraph pauses are more difficult to work with, for two reasons. Relative to between-word junctures, there are a lot fewer of sentence and paragraph breaks in a typical document. Further, relative to experienced writers, less-experienced writers tend to have more difficulties with terminal punctuation. The median number of between-sentence pauses in the Spring 2008 data was 11 (indicating that half the essays had 12 or fewer sentences). This makes them less stable as estimates of student performance. Figure 15 shows the relationship between the pause times and the strand scores. Note that there were two outliers, one

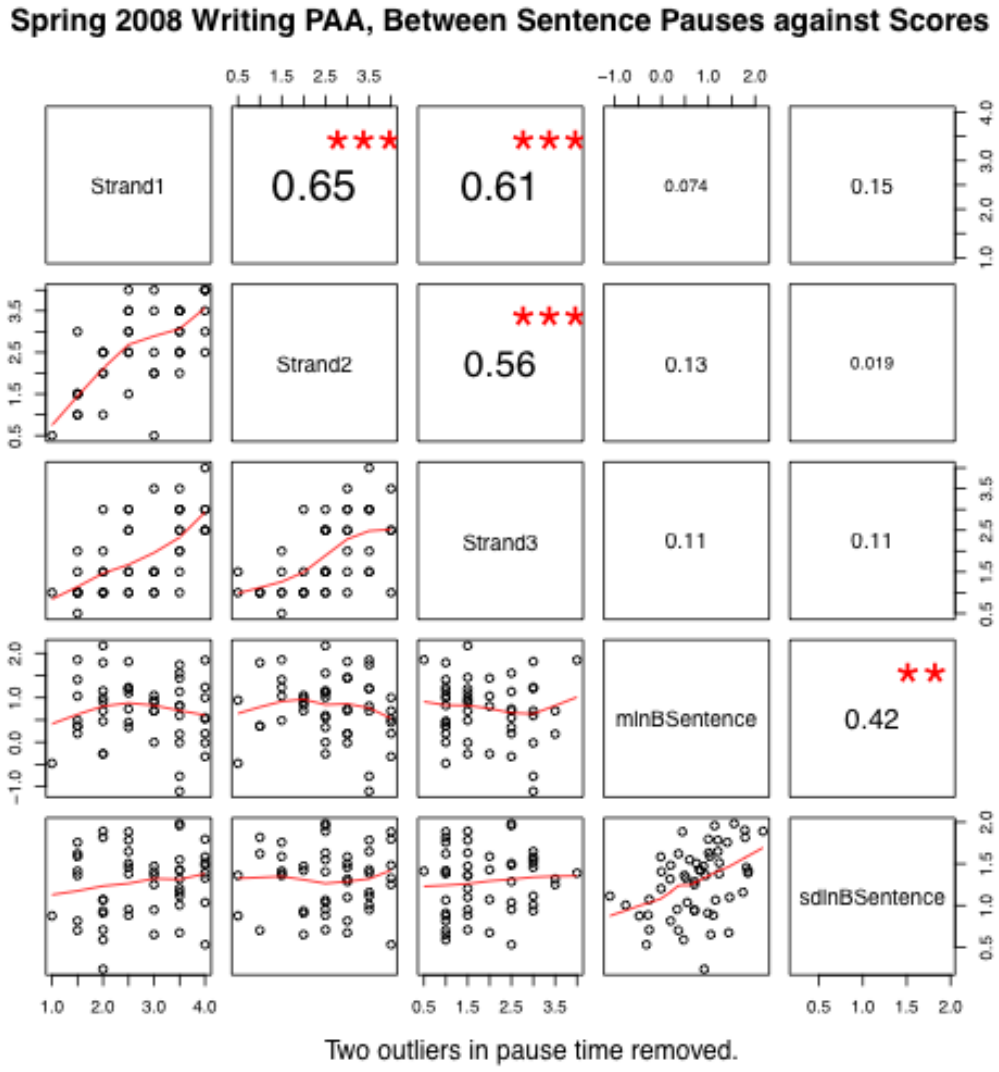

Figure 15. **Relationship between strand scores and between-sentence pauses. PAA = periodic accountability assessment.**

in the mean of the log pause time and one in the standard deviation, which have been removed from this plot. The outlier on the mean log pause time had a low score, and including it induces a small correlation between the mean of the log between sentence pauses and Strand I.

A manual examination of the two outlying essays showed that they were both collections of isolated sentences, which did not constitute a coherent paragraph. The key log for the essay with the large mean time sentence pauses showed a lot of paste events, suggesting that long pauses could be based on selecting text from the source material. It is possible that this kind of pause data could be useful diagnostically, but a larger sample size is required to see if similar events emerge again.

Because there are so few paragraph breaks, the statistics from this type of pause are almost meaningless. The median number of between-paragraph pauses is 0, and the third quartile is one: that is, more than half of the students are producing one-paragraph essays and three-fourths of them are producing no more than two paragraphs. The number of paragraphs is highly predictive of the overall score and may be an important aspect of feedback to both teachers and students. Note that if the key logs were aggregated over several writing assignments, it may be possible to produce more useful diagnostic information from these kinds of pauses.

The analysis of the normal typing events above does not include information about the number of events because that is strongly related to document length. However, the number of editing events is interesting in that this is a count of editing behavior, rather than overall text production. Figure 16 shows the counts of editing behaviors and the mean pause length before edits. Note that the correlation between the number of single and multiple backspace events is very high. This indicates that the distinction between single and multiple backspaces is somewhat artificial (changing a single mistake that occurred several characters previously is often done with multiple backspaces). The two types of backspaces are collapsed for the subsequent analysis.

Note also that there is a negative correlation between the number of editing events and the average pause time before the event. This suggests that students who are making fewer edits are thinking or proofreading more before starting the edits.

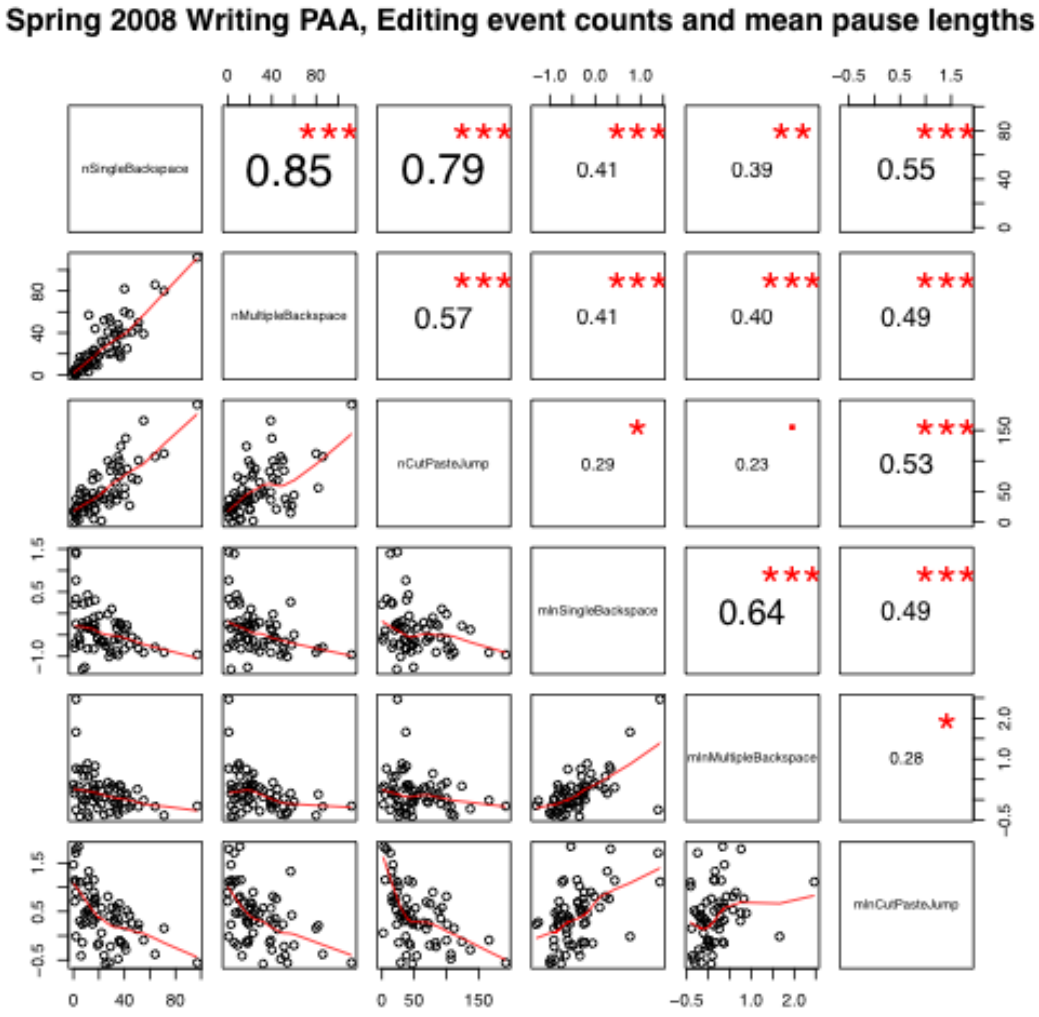

Figure 16. **Relationship between Editing event counts and mean pause lengths. Variables are number of Single BackSpaces; number of Multiple BackSpaces; number of Cuts, Pastes, and Jumps; mean of log Single BackSpace pauses; mean of log Multiple BackSpace pauses; and mean of log Cut, Paste, or Jump pauses. PAA = periodic accountability assessment.**

Figure 17 shows the relationship between backspaces (pooled across single and multiple backspaces) and overall score. There appear to be moderate correlations between both the number of backspaces and the average length of the backspace with the strand scores. The correlations for each of the two types of backspaces look similar.

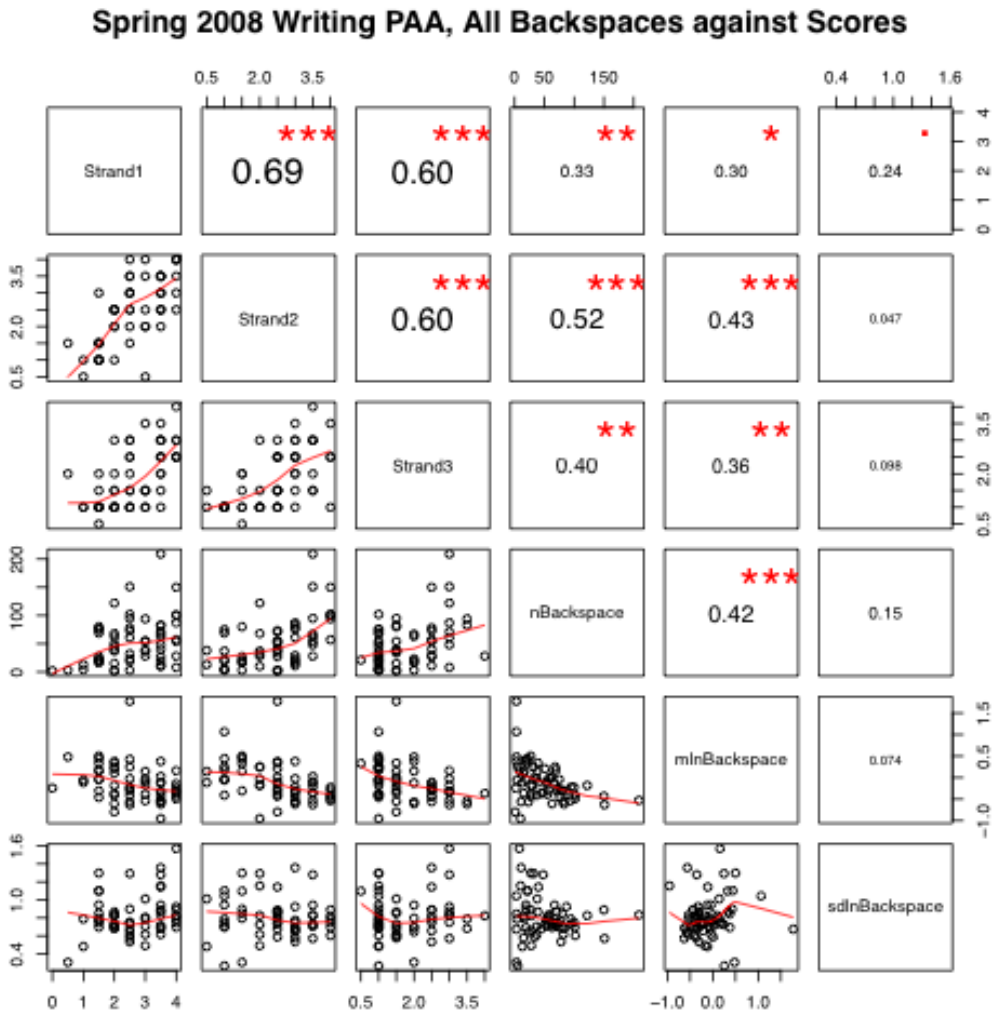

Figure 17. **Relationship of backspaces to strand scores. PAA = periodic accountability assessment.**

Turning to the Cut, Paste, and Jump events, Figure 18 shows their relationship to the strand scores. Again, there is moderate correlation with the number of editing events and the overall score. Also, there appears to be a nonlinear relationship between the mean of the log pause time before an editing event and the Strand II score. This is interesting, as it suggests that there may be multiple mechanisms at play. However, some caution is needed because in the Spring 2008 pilot, many students did a large amount of pasting from the source text.

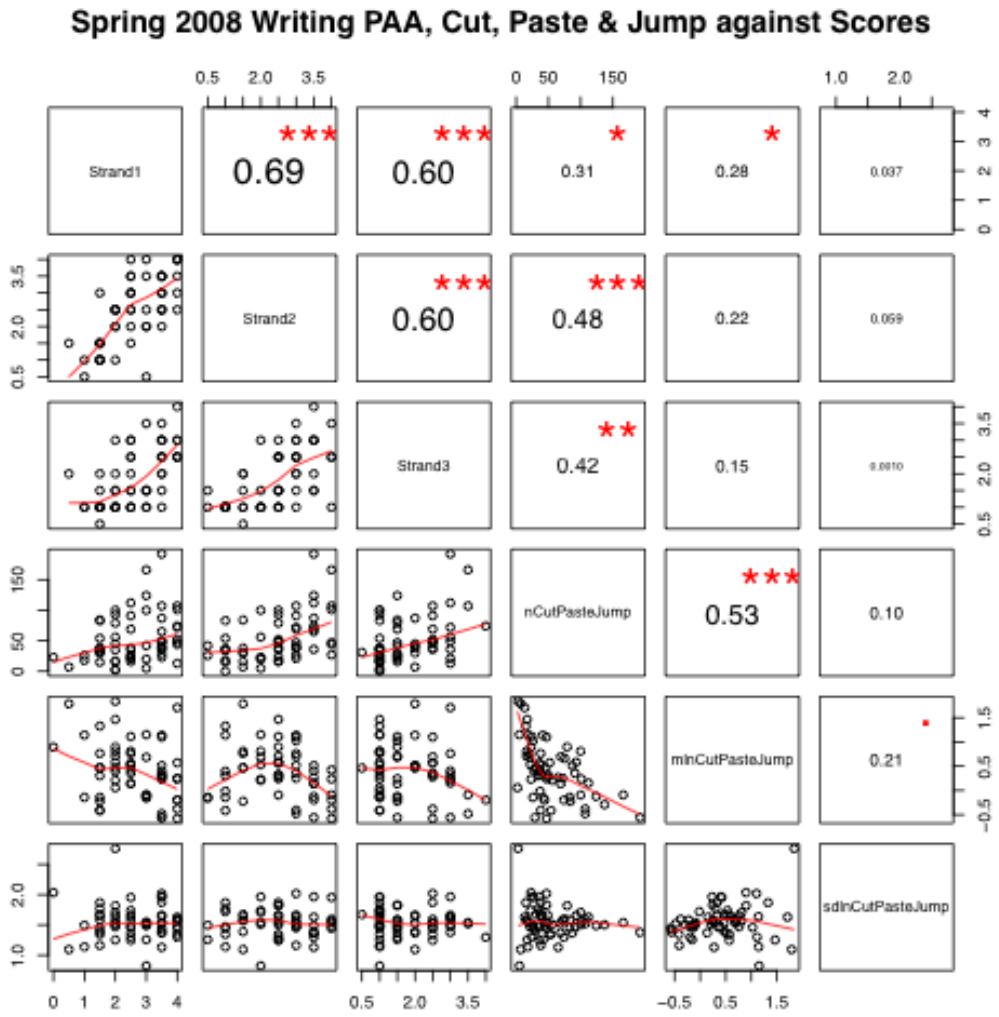

Figure 18. **Relationship between Cut, Paste, and Jump events and strand scores. PAA = periodic accountability assessment.**

One note of caution is in order. Even though the number of editing events seems predictive of the strand scores, it could still be an artifact of the document length and, hence, a general fluency effect. Longer documents require more editing. To explore this, we normalize the number of editing events by dividing by the number of characters in the final essay. Figure 19 shows the result. The strong relationships between the number of editing events and the scores have largely disappeared.

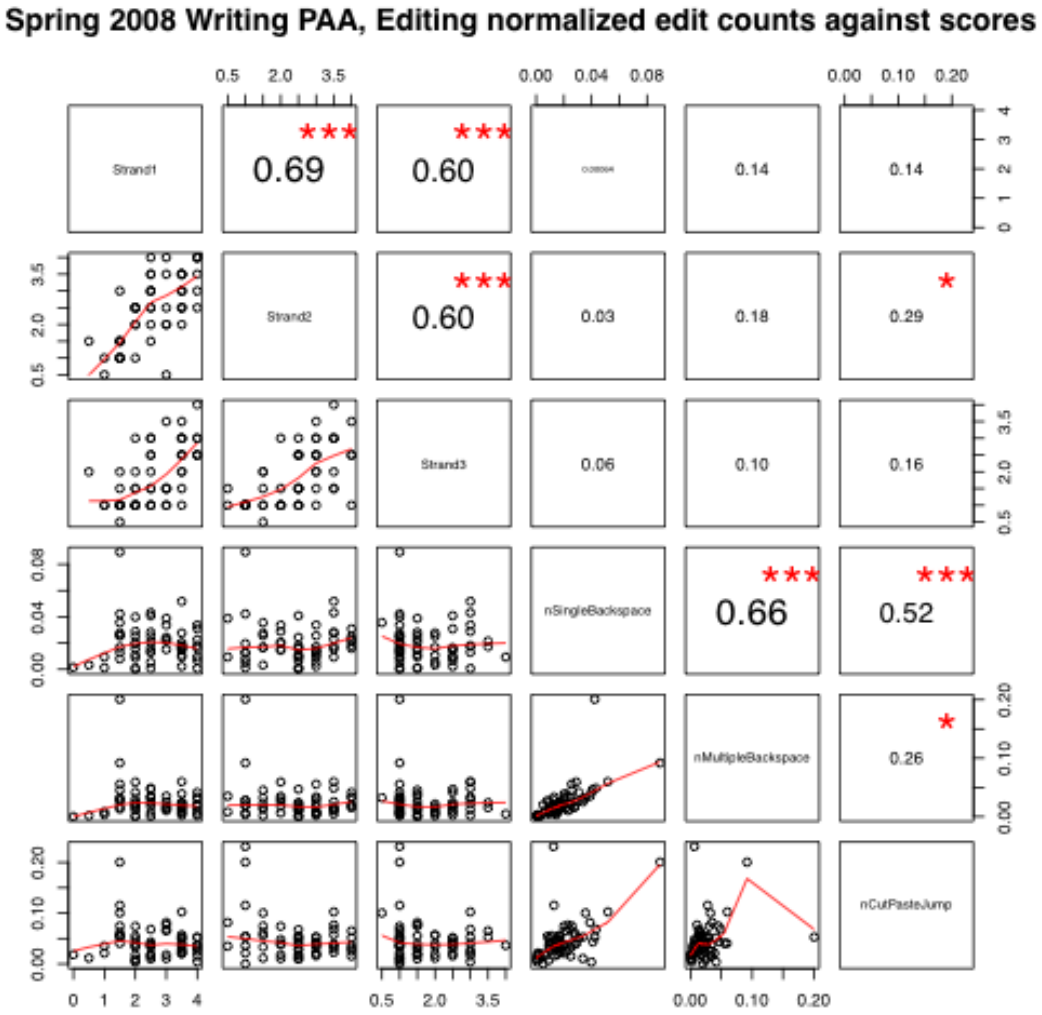

Figure 19. **Relationship between normalized event counts and strand scores. PAA = periodic accountability assessment.**

Finally, to take a quick look to see the value of timing features in automated scoring, we examine some regression models to predict the human scores from automated features. In this analysis, we used three sets of features: the eight generic e-rater features (Attali & Burstein, 2006), eight factors arising from the Source Finder features identified by Deane, Quinlan, and Kostin (2011), and the timing features described above. We used the R (R Development Core Team, 2009) *step* function to search for the model with the lowest BIC[8] statistic. Table 3 compares the best model with and without the timing features.

Table 3

Best Models for Predicting Strand Scores From NLP and Timing Features

Search space	Model	Parameters	BIC	R^2	Adjusted R^2
	Strand I (Sentence-level control)				
NLP only	Strand1 ~ √Grammar + √Mechanics + √Style + ln(DTU) + ln(DTA) + median(WF) + AcademicOrientation	8	-13.33	0.53	0.46
NLP + timing	Strand1 ~ √Usage + √Style + WordLength + AcademicOrientation + SpokenStyle+ OvertPersuasion + nBursts + mlnBursts + mlnInWord	11	-36.44	0.74	0.69
	Strand II (Document-level control)				
NLP only	Strand2 ~ ln(DTU) + ln(DTA) + AcademicOrientation + SentenceComplexity	5	-38.98	0.64	0.61
NLP + timing	Strand2 ~ ln(DTU) + ln(DTA) + AcademicOrientation + totalTime + sdlnBursts	6	-39.72	0.66	0.64
	Strand III (Critical thinking)				
NLP only	Strand3 ~ ln(DTU) + ln(DTA)	3	-32.66	0.36	0.33
NLP + timing	Strand3 ~ √Grammar + ln(DTU) + WordLength + mlnBWord	5	-31.87	0.43	0.39

Note. WF = word frequency; DTU = discourse units; DTA = average discourse unit length.

For Strand I, the timing features lead to strikingly better predictions (adjusted-R^2 increases by .2). For Strands II and III, they do not seem to make much of a difference (the BIC scores are nearly identical). It is possible (especially for Strand III) that the timing features are providing an independent measure of fluency and, hence, the observed predictive power is due to the general writing fluency factor. However, the finding with Strand I is intriguing, and the timing features, particularly average burst length and average within-word pause, are particularly focused on word- and phrase-level writing fluency, which is also the focus of that strand score.

4.4 Pause Features From Mixture Model Analysis

The analysis of the previous section summarized the within-person pause distributions with two statistics: the mean of the log pauses and the standard deviation of the log pauses. While this summary would be adequate if the pauses followed a lognormal distribution, the analysis of Section 4.2 shows that the lognormal distribution does not capture everything that is going on within the key logs.

The alternative proposed in Section 4.3 is a mixture of lognormal distributions. This distribution is summarized by five statistics: the mean and standard deviation of the "low" component, the mean and standard deviation of the "high" component, and the proportion of pauses in the low component. For the sake of identifiability, the mixture component with the smaller mean is called the low component. The question arises: do these components provide additional information about the student performance?

The first question that arises is how do the observed mean and standard deviations of the log pauses (analyzed in Section 4.3) relate to the mixture components. Figures 20 and 21 show these relationships. In both cases, there is a strong correlation between the overall mean and the mean of the low-mixture component. There is also a strong to moderate correlation between the standard deviation of the complete data and the mean and standard deviation of the high-mixture component. As the mean of the log pauses was the component that had the highest correlation with the strand scores, this seems to reinforce the idea that those correlations are based on fluency effects.

Looking at the relationship between the five statistics of the mixture components and the strand scores (Figures 22 and 23), we see a negative correlation between the mean length of pause in the lower mixture component and the strand scores. This is likely to be once more a fluency effect. For the within-word pauses (Figure 22), there is also a correlation with the proportion of pauses in the lower mixture component and the score. This is interesting, as long within-word pauses are likely related to specific dysfluency with orthography. The same effect does not appear in the between-word data (Figure 23), possibly because even the better writers are frequently pausing within sentences, and pause length alone does not distinguish between pauses to consider mechanical issues (e.g., punctuation) and pauses to consider high-level writing problems (e.g., nuanced word choice).

The method involved in coming up with the parameters for the mixture components is difficult to automate. It involves assessing the convergence of the model-fitting process for every student; recall that the two-component model did not converge in a number of cases. While that is not a substantial issue for a trained statistician, it would be problematic in the context of automated scoring where human intervention would slow the delivery of scores. The question arises: is there a statistic that is simpler to calculate, but which would provide much the same information?

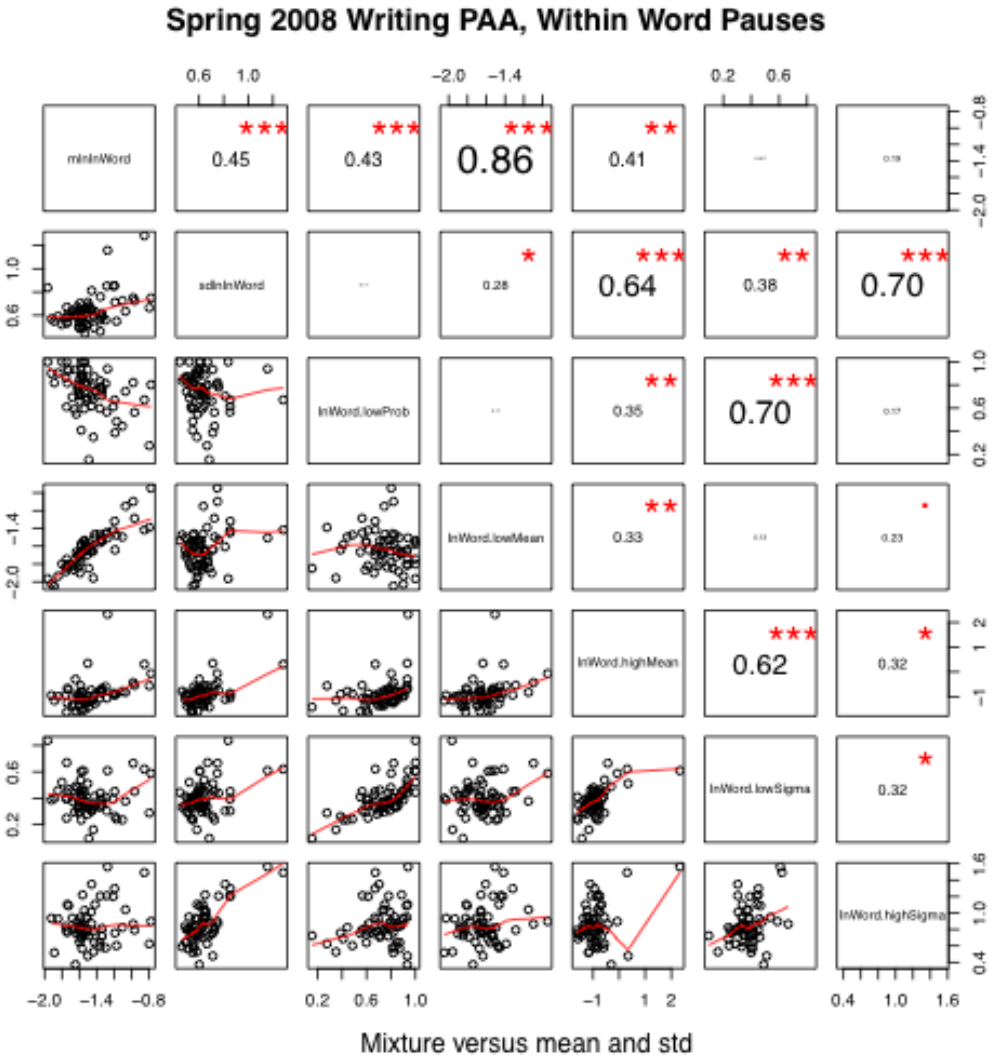

Figure 20. **Within-word pause mixture components and whole sample mean and SD. Variables are mean and SD of the log InWord data (one-component model), probability of being in the high component model, mean of the low and high components, SD of the low and high components. PAA = periodic accountability assessment.**

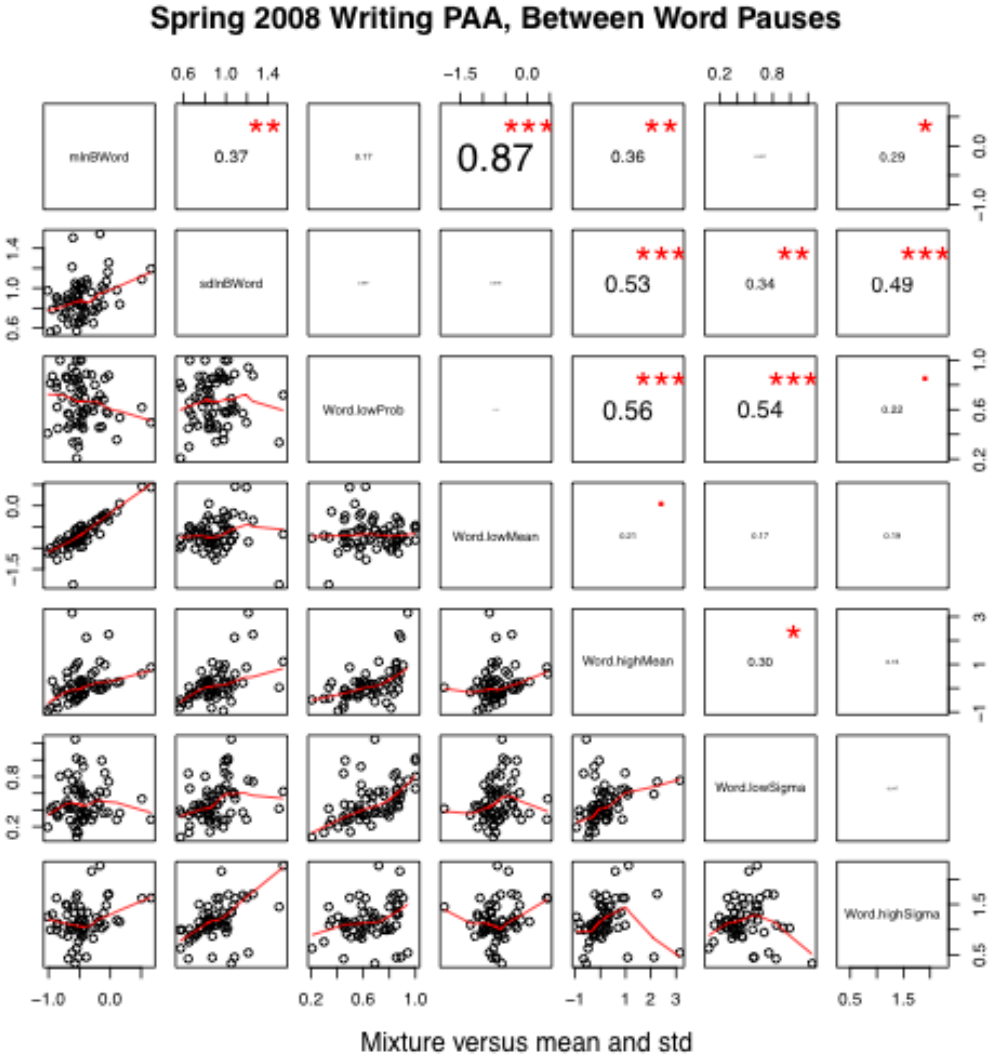

Figure 21. **Between-word pause mixture components and whole sample mean and SD. PAA = periodic accountability assessment.**

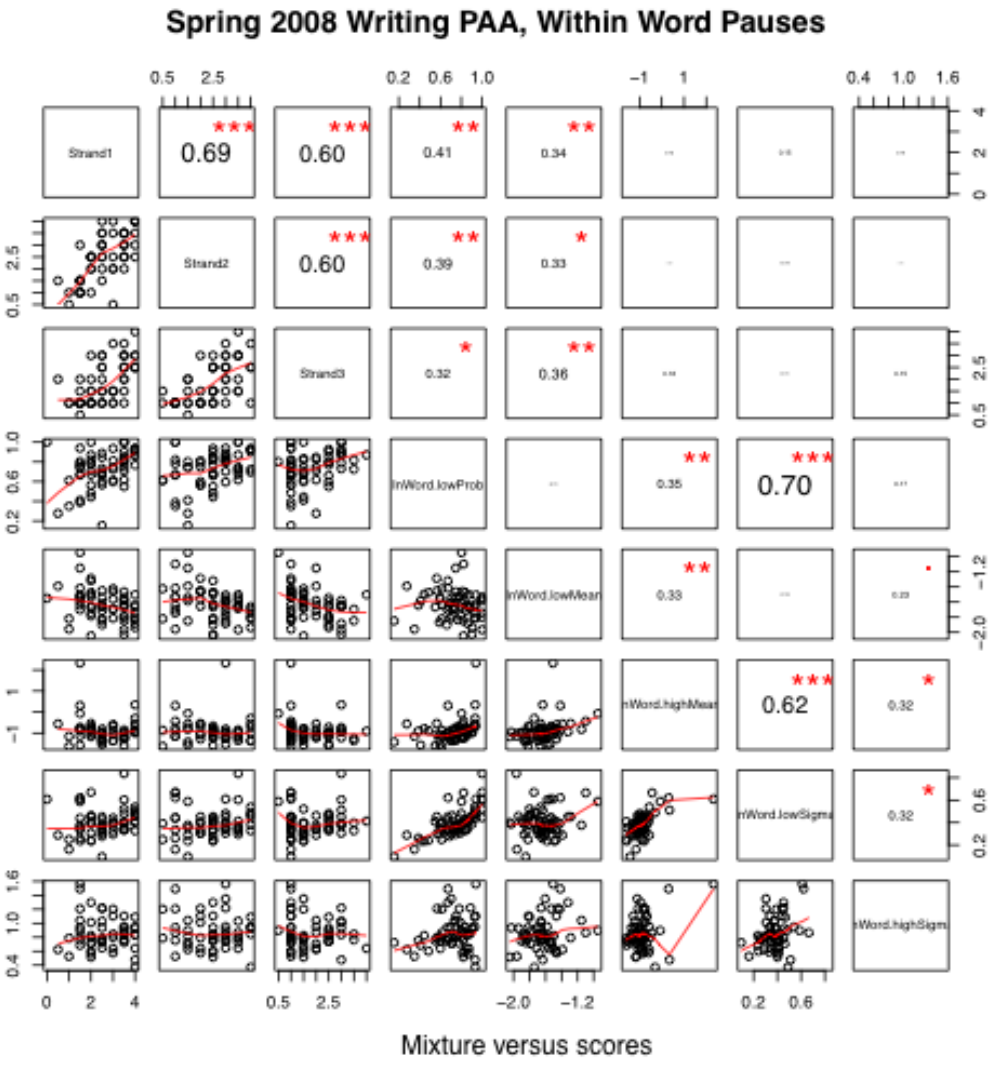

Figure 22. **Within-word mixture components and strand scores. Variables are scores on Strands I, II, and III; probability of being in low component; means of low and high components; and SD of low and high components. PAA = periodic accountability assessment.**

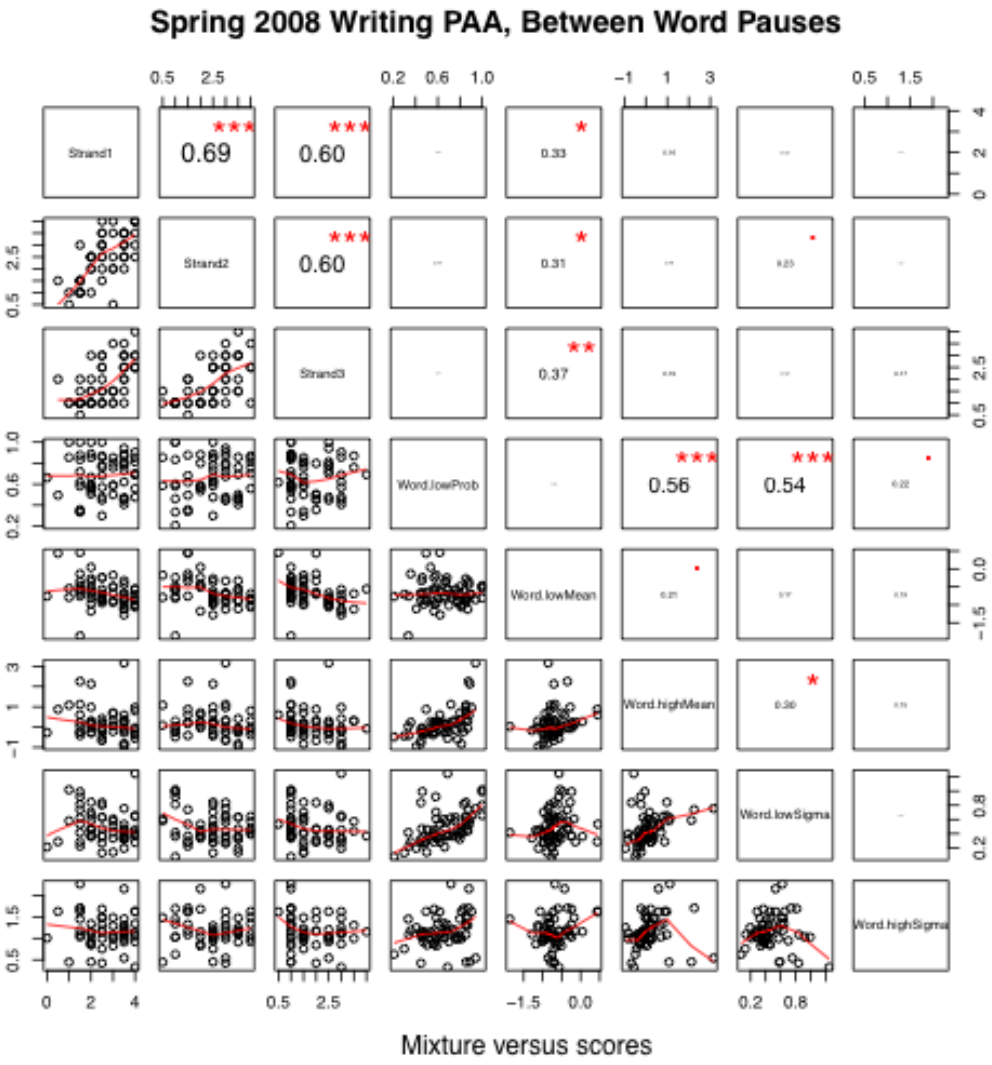

Figure 23. **Between-word mixture components and strand scores. Variables are scores on Strands I, II, and III; probability of being in low component; means of low and high components; and SD of low and high components. PAA = periodic accountability assessment.**

If we knew the parameters of the mixture component, we could devise a simple rule for classifying pauses into the high or low component. This rule would take the form of a cut point; pauses shorter than the cut point would be more likely to be from the low component, and pauses longer than the cut point would be more likely to be from the high component. If we knew the

parameters, the cut point could be determined analytically (given a relative weight to false-positive and false-negative errors). However, if we simply used a fixed cut point (possibly derived by examining similar student essays), we may be able to get an approximate classification of the pauses without the need for formal model fitting.

To explore this, we separated the within-word and between-word pauses into high and low components based on the two-thirds of a second cut point used in determining breaks. We then calculated the mean and standard deviation of the log pause time for the high and low components for each student in the sample (note that one student did not have any within-word pauses longer than two-thirds of a second). This yielded five scores similar to the parameters of the mixture model. Table 4 shows the correlations between the mixture model parameters and the cut point–based statistics for the within-word data, and Table 5 shows the correlations for the between-word data. In both tables, strong correlations along the main diagonal indicate that the cut point–based statistics are producing a good approximation to the mixture component parameters. In the case of the within-word pauses (Table 3), the correlations are very high, indicating that the cut point–based statistics may be a good approximation. In the case of the between-word pauses (Table 4), the correlations are more modest. Perhaps the two-thirds of a second is not the ideal cut point, and a better cut point needs to be found.

Table 4

Correlations of Mixture Parameters With Cut Point Statistics for Within-Word Pauses

Mixture component	Mixture component				
	P(low)	High mean	Low mean	High SD	Low SD
P(low)	**0.35**	***-0.44***	***-0.69***	-0.20	-0.27
High mean	0.25	**0.67**	0.20	***0.84***	0.45
Low mean	**-0.37**	**0.24**	***0.88***	-0.05	0.03
High SD	*0.26*	***0.50***	0.09	***0.88***	**0.37**
Low SD	-0.16	0.14	-0.07	**0.44**	0.31

Note. Font is based on *p*-value for test of difference from zero. ***Bold-italic*** indicates value less than 0.001. **Bold** indicates value less than 0.01. *Italic* indicates value between 0.05 and 0.1. Roman type indicates value above 0.05.

Table 5

Correlations of Mixture Parameters With Cut Point Statistics for Between-Word Pauses

Mixture component	Mixture component				
	P(low)	High mean	Low mean	High SD	Low SD
P(low)	0.18	*-0.29*	***-0.85***	-0.21	-0.18
High mean	0.03	***0.47***	0.07	***0.48***	0.10
Low mean	0.00	0.00	***0.44***	-0.08	**-0.35**
High SD	**0.35**	***0.64***	-0.07	***0.59***	0.09
Low SD	-0.17	-0.09	0.20	0.24	**0.43**

Note. Font is based on *p*-value for test of difference from zero. ***Bold-italic*** indicates value less than 0.001. **Bold** indicates value less than 0.01. *Italic* indicates value between 0.05 and 0.1. Roman type indicates value above 0.05.

5. Preliminary Conclusions and Future Research Directions

The sample sizes for both the Fall 2007 and Spring 2008 CBAL Writing pilots were small. Some of this was intentional (they were never intended to be large samples), and some was due to technical difficulties. As a consequence, almost any of the results described above could be artifacts of the small sample size. In addition, as these were early pilot tests, several problems with the task had not yet been identified. In particular, the way the Spring 2008 task was worded encouraged the students to quote extensively from the source material for the task. These quotations distorted both natural language features and timing features.

The first steps for future research are obviously to try and replicate the findings with a larger data set. Fortunately, a nationally representative sample with about 1,000 students per essay was collected in the fall of 2009. The technical problems with the keystroke logging were solved, and full keystroke logs will be available for those data. Although the analyses if these data is not yet complete, preliminary results appear similar to the ones presented in this paper.

The approach to capturing events used in the Spring 2008 data collection appears promising, and it was the one chosen for use with the larger sample. The definitions of the within-word and between-word features appear to be reasonably useful, as does the definition of the burst-length features. The between-sentence and between-paragraph features are probably sound, but the short length of the essay means that they yield little information about the examinee. The editing features need more thought; in particular, the distinction between single and multiple backspaces does not seem to be useful in distinguishing between examinees.

Although the definitions of the features appear promising, the actual software used to isolate the features needs to be reworked to handle the higher volume of data generated by a large sample.

One of the most interesting finding of the analysis of the Fall 2007 data was replicated in the analysis of the Spring 2008 data. That was the highly leptokurtic distributions for the pause times which could be modeled as mixtures of lognormals. The mixture of lognormal distribution is interesting because it corresponds to a cognitive model of text production in which several different processes are operating. The majority of the pauses (especially within-word and between-word) are likely generated in the course of inscription (i.e., typing the words into the essay). The second mixture component corresponds to more complex cognitive activity associated either with attending to issues of grammar, mechanics, usage or style, or those of organization and planning. The cognitive model would lead us to believe that there may be more than two mixture components; however, the short length of the essay does not allow the additional components to be distinguished from data. Accumulating data across many essays (possibly through the use of keystroke capture tools for routine class assignments) may generate data that can distinguish these higher-order mixture components.

The correlations between various timing features and the strand scores are high enough to be promising. The better prediction for Strand I is also encouraging. Unfortunately, both the small sample size and the unreliability of the scoring make it difficult to draw conclusions. It is possible that the timing features are just another measure of fluency; however, we may be able to look at fluency for specific parts of the construct.

Perhaps more interesting is the presence of some distinct outliers in the timing features. The unusual ways that these students are spending their time may be a symptom of a deeper cognitive difficulty or poor test-taking strategies. Being able to alert teachers to these unusual patterns for follow-up may be a beneficial side-effect of the work on studying timing during writing.

References

Ahlsén, E., & Strömqvist, S. (1999). ScriptLog: A tool for logging the writing process and its possible diagnostic use. In F. Loncke, J. Clibbens, H. Arvidson, & L. Lloyd (Eds.), *Augmentative and alternative communication. New directions in research and practice* (pp. 144–149). New York, NY: Wiley.

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater v.2.0. *Journal of Technology, Learning, and Assessment, 4*(3). Retrieved from http://ejournals.bc.edu/ojs/index.php/jtla/article/view/1650/1492

Bennett, R. E., & Gitomer, D. H. (2009). Transforming K-12 assessment: Integrating accountability testing, formative assessment, and professional support. In C. Wyatt-Smith & J. Cumming (Eds.), *Educational assessment in the 21st century* (pp. 43–61). New York, NY: Springer.

Berninger, V. W., & Swanson, H. L. (1994). Modifying Hayes and Flower's model of skilled writing to explain beginning and developing Writing. *Advances in Cognition and Educational Practice, 2*, 57–81.

Bourdin, B., & Fayol, M. (1994). Is written language production more difficult than oral language production? A working memory approach. *International Journal of Psychology, 29*(5), 591–620.

Bourdin, B., & Fayol, M. (2000). Is graphic activity cognitively costly? A developmental approach. *Reading and Writing, 13*(3–4), 183–196.

Calfee, R. C., & Miller, R. G. (2007). Best practices in writing assessment. In S. Graham, C. A. MacArthur, & J. Fitzgerald (Eds.), *Best practices in writing instruction* (pp. 265–286). New York, NY: Guilford Press.

Chambers, J. M. (2004) *Programming with data: A Guide to the S language*. New York, NY: Springer.

Cochran Smith, M. (1991). Word processing and writing in elementary classrooms: A critical review of related literature. *Review of Educational Research, 61*(1), 107–155.

Deane, P. (2012). Rethinking K-12 writing assessment. In N. Elliot & L. Perelman (Eds.), *Writing assessment in the 21st century: Essays in honor of Edward M. White* (pp. 87-100). New York, NY: Hampton Press.

Deane, P. (2011). *Writing assessment and cognition* (ETS Research Report No. RR-11-14). Princeton, NJ: ETS.

Deane, P., Fowles, M., Baldwin, D., & Persky, H. (2011). *The CBAL summative writing assessment: A draft 8th-grade design* (ETS Research Memorandum No. RM-11-01). Princeton, NJ: ETS.

Deane, P., Quinlan, T., & Kostin, I. (2011). *Automated scoring within a developmental, cognitive model of writing proficiency* (ETS Research Report No. RR-11-16). Princeton, NJ: ETS.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royals Statistical Society, Series B, 30,* 205–247.

Grün, B., & Leisch, F. (2008). FlexMix Version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, *28*(4), 1–35. Retrieved from http://www.jstatsoft.org/v28/i04/

Hayes, J. R., & Flower, L. S. (1980). Identifying the organization of writing processes. In L. Gregg & E. R. Steinberg (Eds.), *Cognitive processes in writing* (pp. 3–30). Hillsdale, NJ: Lawrence Erlbaum Associates.

International Organization for Standardization. (1993). *ISO/IEC 9945-2:1993: Information technology—Portable Operating System Interface (POSIX)—Part 2: Shells and utilities.* Retrieved from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=17841

Matsuhashi, A. (1981). Pausing and planning: The tempo of written discourse production. *Research in the Teaching of English, 15*(2), 113–134.

McCutchen, D., Covill, A., Hoyne, S. H., & Mildes, K. (1994). Individual differences in writing: Implications of translating fluency. *Journal of Educational Psychology, 86*(2), 256–266.

R Development Core Team. (2009). R: A language and environment for statistical computer [Computer software]. Vienna, Austria: R Foundation for Statistical Computing.

Schilperoord, J. (2002). On the cognitive status of pauses in discourse production. In T. Olive & C. M. Levy (Eds.), *Contemporary tools and techniques for studying writing* (pp. 61–90). Dordrech, Netherlands: Kluwer Academic Publishers.

Shepard, L. (2006). Classroom assessment. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 624–666). Westport, CT: Praeger.

Tetreault, J., & Chodorow, M. (2009). *Essay similarity detection: 2009 final report.* Unpublished manuscript.

Van Waes, L., & Leijten, M. (2006). Logging writing processes with Inputlog. In L. V. Waes, M. Leijten, & C. Neuwirth (Eds.), *Writing and digital media* (Vol. 17, pp. 158–166). Oxford, England: Elsevier.

Woodcock, R., & Johnson, B. (2001). *Woodcock-Johnson Psychoeducational Battery III tests of achievement*. Chicago, IL: Riverside Publishing.

Notes

[1] Russell Almond conducted the research in this report while employed by ETS. He is currently an associate professor in the Department of Educational Psychology and Learning Systems in the College of Education at Florida State University. He can be reached at ralmond@fsu.edu.

[2] Thomas Quinlan conducted the research in this report while employed ETS. He is currently an adjunct at the Department of Psychology, Rutgers University.

[3] Additional CBAL writing pilots took place in the fall of 2008 and the spring of 2009, but the keystroke logging system was not used in those pilots. A large-scale national pilot study was collected in the fall of 2009. This pilot included keystroke logging data and substantially larger sample sizes, but the data analysis is still underway at the time of this report.

[4] For example, in a Swedish locality, the accented letters used in Swedish are accepted as part of the designation of [:upper:] and [:lower:] without additional work on the part of the programmer using the POSIX designations. Unfortunately, the POSIX [:punct:] class does not work sufficiently well for our purposes, so this part of the design would need to be revisited in porting this software to work with a non-English language.

[5] The use of emoticons to terminate sentences is not considered proper usage in formal academic writing, but it is common practice in chat, instant messages, and other common online text communication. As the students are developing writers, they may not properly recognize the differences in the genres and use the emoticons to terminate sentences.

[6] There is no entry in the key log for the writer pressing the final "submit answer" button, or for the assessment timing out, so we have no idea how long the writer spent reviewing the essay before submitting or whether or not the writer considered the essay complete.

[7] This number was chosen after a preliminary look at the data. The median (across students) of the median (within student) between-word pause times was around 0.6 seconds. This number was chosen to be slightly larger than that value; thus, about half of between-word pauses should be classified as within a single burst.

[8] We also looked at the model with the lowest AIC statistic. As the BIC statistic generally chose more parsimonious models, only those results are reported.

[9] Line breaks and tabs have been added to improve readability. In the original file, all the data appears as a single line of text (unless linefeeds are embedded in the text).

[10] Although XML and HTML have similar syntax, XML is much stricter about certain aspects: in particular, all opening tags must have a corresponding close tag (starting with a slash) or be self-closing (ending with a slash). XHTML is a modification of HTML to make it follow XML syntax rules. Properly speaking, the close paragraph tag `</p>` is XHTML and not HTML, but most HTML display engines simply ignore it and the document appears correctly formatted. Curiously, while the logger and essay use the XHTML `</p>`, they use the plain HTML `<br>` and not the XHTML `<br/>` tag.

[11] Earlier versions of the S programming language support a very loose object model. Chambers (2004) introduces a more formal object model to the S language, which has become known as the S4 object model. The current R distribution contains a partial implementation of S4.

Appendix A

Keystroke Log Formats

This appendix documents the data formats used for capturing the keystroke logs. Section A.1 shows the format used in the Fall 2007 study. Section A.2 shows the XML format used for all data in the Spring 2008 study. Section A.3 shows the format used in just the keystroke logging. Section A.4 summarizes some of the HTML codes that can appear in essays and log files.

A.1 Fall 2007 Log Format

Keystroke logs for the Fall 2007 pilot study were placed in separate text files. The names encoded both the student ID and the form (Form M or Form K). The logs themselves have the following format:[9]

```
{keys:[{t:"2376462",k:32,c:" ",m:0},
        {t:"2376225",k:115,c:"s",m:0},
        {t:"2376075",k:104,c:"h",m:0},
        {t:"2375705",k:111,c:"o",m:0},
        {t:"2375545",k:117,c:"u",m:0},
        {t:"2375365",k:108,c:"l",m:0},
        {t:"2375135",k:32,c:" ",m:0},
        {t:"2374645",k:92,c:"\",m:0},
        {t:"2374035",k:8,c:"&#x08;",m:0},
        {t:"2373835",k:8,c:"&#x08;",m:0},
        {t:"2373485",k:8,c:"&#x08;",m:0},
        {t:"2373181",k:100,c:"d",m:0},
        {t:"2372315",k:8,c:"&#x08;",m:0},
        {t:"2371806",k:108,c:"l",m:0},
        {t:"2371716",k:100,c:"d",m:0},
        {t:"2371506",k:32,c:" ",m:0},
        {t:"2370156",k:110,c:"n",m:0},
        {t:"2370086",k:111,c:"o",m:0},
        {t:"2370017",k:116,c:"t",m:0}],actions:[]}
```

There are two sections, keys and actions, and the actions section is always empty.

Within keys, each expression in braces refers to a keystroke. The number following `t:` is a timestamp. (It is unclear when the origin is; only differences are used in the analysis.) The number following `k:` gives the ASCII (or Unicode) code for the character, and the string following `c:` gives the actual character. The number following `m:` is always zero or one, and it

may be a flag indicating mouse movement. However, as the actions area which would have indicated the mouse action is empty, it was not used in any of the analyses.

The special sequence `&#x08;` is used to indicate a backspace (this is an XML/HTML escape sequence for a nonstandard character). Quotation marks are indicated with an escaped quote (`k:34,c:"\""`); however, backslashes are not properly escaped (`k:92,c:"\"`). This causes some difficulty for parsing code.

A.2 Spring 2008 XML Format

The Spring 2008 data are stored in an XML format that combines the final essay, the event log, and the responses to the multiple choice portions of the test.

```
<responses>
  <response ITEMRESULTID="Mini1" CANDIDATEID="Mini">
    <responsepart
ID="W_SLEEP03_MC01"><![CDATA[1,1,1,1,1,1]]></responsepart>
    <responsepart ID="W_SLEEP03_TX01"><![CDATA[<p></p>Teens
need.  <p></p>]]></responsepart>
         <responsepart ID="W_SLEEP03_TX01-
actions"><![CDATA[{0:{p:0,o:"",n:"<p></p>",t:"173.18"},…]]></res
ponsepart>
  </response>
</responses>
```

The highest level element is `<responses>`, which serves as a container for the responses for all students in the file. It contains any number of `<response>` elements, which contain the results for an individual student. The `<response>` element has two attributes: ITEMRESULTID, which provides an identifier for the prompt used, and CANDIDATEID, which provides an identifier for the student. It also contains any number (number is determined by the task model for the task identified by ITEMRESULTID) of `<responsepart>` elements.

The `<responsepart>` element has an ID attribute that identifies the parts of the response. In the Spring 2008 assessment, `W_SLEEP03_MC01` contains the multiple choice responses, `W_SLEEP03_TX01` contains the student's essay, and `W_SLEEP03_TX01-actions` contains the keystroke log. The value of the `<responsepart>` element is wrapped in the `<![CDATA[…]]>` to protect any HTML tags within the essay or keystroke log (used to indicate line and paragraph breaks). Appendix A.4 describes commonly occurring HTML codes that could occur.

A.3 Spring 2008 Log Format

The Spring 2008 keystroke log format differs from the Fall 2007 format. One immediately visible change is that the log entries are preceded by a sequence number. This would help identify gaps in the log (e.g., due to a network connection problem). Additionally, the entries in the log have now changed. They are: `p:` for the position of the change, `o:` for the old [deleted] text (as a character string with possible embedded HTML control sequences), `n:` for the new [inserted] text (as a character string with possible embedded HTML control sequences), and `t:` for a timestamp in units of seconds. This is now a real number rather than an integer; however, the origin is still unspecified. Only differences are used in the analysis.

Some sample log entries:

```
0:{p:0,o:"",n:"<p></p>",t:"173.18"},
1:{p:7,o:"",n:"T",t:"175.91"},
2:{p:8,o:"",n:"e",t:"176.19"},
3:{p:9,o:"",n:"e",t:"176.37"},
4:{p:10,o:"",n:"n",t:"176.53"},
5:{p:11,o:"",n:"s",t:"176.76"},
6:{p:11,o:"s",n:"",t:"176.93"},
7:{p:10,o:"n",n:"",t:"177.10"},
8:{p:9,o:"e",n:"",t:"177.29"},
9:{p:8,o:"e",n:"",t:"177.48"},
10:{p:8,o:"",n:"e",t:"178.30"},
11:{p:9,o:"",n:"e",t:"178.50"},
12:{p:10,o:"",n:"n",t:"178.66"},
13:{p:11,o:"",n:"s",t:"178.82"},
14:{p:12,o:"",n:" <br>",t:"179.38"},
15:{p:13,o:"",n:"s",t:"180.92"},
34:{p:39,o:"",n:"to ",t:"399.30"}
146:{p:137,o:"",n:" t",t:"437.15"},
147:{p:139,o:"",n:"h",t:"437.25"},
148:{p:140,o:"",n:"e",t:"437.29"},
1050:{p:622,o:"<br>",n:"the first class of the morning is often
a waste, with as many as 28
percent of students falling asleep.  Some are so sleepy
they don't
even show up.\"",t:"723.13"}
```

A.4 HTML Codes and Entities

As both the essays and the keystroke logs are transported within an XML wrapper, the logging program uses conventions from XML and HTML to address issues with special characters. This is not actually required, as the essay and key log are wrapped within the

<![CDATA[...]]> escape sequence; the XML parser does not check for errors in that region. The character classifier looks for these special sequences and translates them into ASCII characters for further processing.

New lines generated by the writer are marked with one of two HTML tags. The first is the new paragraph tag . This often appears with the XHTML[10] close paragraph tag, . Unfortunately, this does not always appear consistently within the keystroke logs. It can appear as , ,
 , or /p> <. The last is not well formed XML or HTML, but was observed in several logs. The character classifier looks for all of these sequences and replaces them with an ASCII vertical tab character (\013, or \v). It does the same for and tags occurring separately. These are all assigned to category EOP.

Another HTML code that occurs is
, which is used to indicate a line break without implying a new paragraph. As there is really no reason for this code in a typical essay, it is unclear why it is generated. The character classifier replaces
 codes (not wrapped within) with an ASCII new line character (\n) and assigns them to the category EOL.

XML and HTML also use a series of *entity* declarations to handle special characters. These are multicharacter codes that start with an ampersand and end with a semicolon. All of the ones here map to a single character. The codes that are potentially of interest are: (nonbreaking space), & (ampersand, &), < (less than, <), > (greater than, >), ' (apostrophe, ‘), and " (quote, “). The character classifier checks for these codes and replaces them with their ASCII equivalents. There is also a general escape sequence—&#x*hh*;—which is used for arbitrary characters, with *hh* replaced with the ASCII or Unicode numeric equivalent in hexadecimal. This is used in the Fall 2007 logs to indicate backspace characters (ASCII Code 08), but not used in the 2008 code. The character classifier does check for this entity.

The Spring 2008 logger did not use the XML entities to escape special characters; instead, it partially implemented the c-style escape sequences (preceding special characters with a backslash, ‘\’). In particular, it indicated quotes within a string with the escape sequence ‘\″’. This was only partially implemented in the Spring 2008 version with two difficulties: (a) if multiple quotes appeared in a string (a cut or paste event), then only the first one was properly escaped, and (b) the backslash character representing itself was not escaped; thus, an event that

consisted of inserting a single backslash character would appear as '"\"' which could cause confusion with an escaped quote. Both of these issues should be corrected in future versions of the logger.

The presence of XML/HTML tags and entities in the document presents one other issue: the position data given in the log counts each character within the tag or entity in the position count. Look at the first two entries in the sample log (Section A.3). The tag (a single conceptual character) advances the character counter by 7. When the character classifier replaces this sequence with a vertical tab (a single character), it needs to adjust the position. Both the original and adjusted positions (as well as the size of any adjustment) are stored in the KeyEvent object (Appendix B). However, there is a possibility that the adjustment might not be correct for an as-of-yet unidentified problematic case. This would cause both the jump determinations and the append/edit distinctions to be inaccurate.

Another potential source of difficulty comes from the fact that the first entry usually has a special structure, which includes enclosing paragraph tags. In particular, the first entry often looks like: *text*
 , where *text* is any bit of text and may contain embedded HTML tags or entities (particularly, if the first event is a paste event). These HTML codes set up the initial paragraph and a blank line at the end of the document. Usually, the student continues adding to the end of text, so that should be treated as the end of the document. Unfortunately, not all of the first events follow this pattern. Sometimes the
 code is missing. In the first two logs, the document starts with and proceeds to append text after this paragraph mark. The code used to judge the append/edit distinction tries to take this into consideration, but it is not clear that it covers all possible cases.

Appendix B

R Object Model

The classifier for the Spring 2008 data was written in R (R Development Core Team, 2009) using the S4[11] object model (Chambers, 2004). The classifier uses three objects: A KeyLogEntry object for individual entries (corresponding to one bracketed expression in the log) and KeyLog object for the complete log, which inherits from a more general TimingLog object. Figure B1 shows the core object model for the classifier.

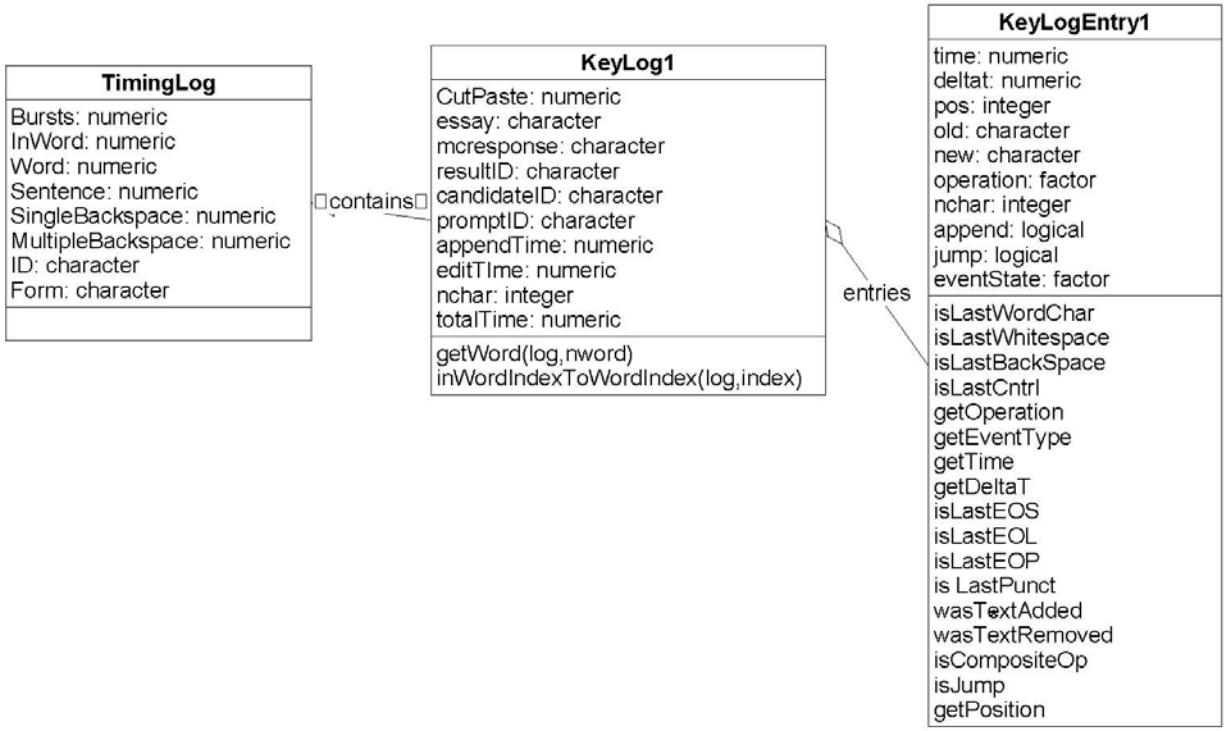

Figure B1. **Core object model for the classifier.**

The Fall 2007 classifier was written in Perl and worked differently. Rather than build up an annotated version of the key log (as Spring 2008 version does), it simply classified the events into one of the types, InWord, Word, Sentence, Paragraph, SingleBackSpace, or MultipleBackSpace. It then added the pause time to the appropriate vector (summing over white space between words and paragraphs). It also calculated bursts and stored the bursts in a similar vector. The TimingLog object was designed to work with the Fall 2007 data, and the KeyLog object has extensions to deal with the Spring 2008 data.

The biggest feature of the KeyLog object is the collection of KeyLogEvent objects that contain the individual records for the key log. It contains some minor changes as well. It contains a field for the final essay and for the multiple choice responses. It contains a new list of timing information, CutPaste, and it calculates the total time spent on text generation (appending) versus editing operations.

The KeyLogEvent contains information both about the timing event from the log record and additional annotation information. Thus it contains the old and new character strings, the position, and both the absolute and differenced time (deltat*)*. The fields operation, append, jump*,* and eventState are annotations created by the classifier. Operation is one of Insert, Delete, Paste, Cut, or Replace and is determined by the size of the old and new character strings (Insert or Delete if the number of characters is at or below the threshold; Replace if both old and new text are present). EventState is one of BEGIN, InWord, Sentence, Paragraph, BackSpace, Edit, END, or Error and is set by the classifier.

The append and jump fields are logical flags; jump is set if the position of the change is different from what would have been expected if the writer was continuing from the current cursor location. The append flag is set when the writer is adding text to the end of the document. The appendTime and editTime fields of the KeyLog object are set according to the append flag. When append = TRUE, then the time for the event is credited to appendTime; when append = FALSE, the time is credited to editTime.

Note that the KeyLogEntry object has a number of methods that query the state of the entry. For example, isLastWordChar(), isLastWhiteSpace(), isLastCntrl(), isLastEOS(), isLastEOP(), isLastEOL, isLastPunct(), wasTextAdded(), wasTextRemoved(), isCompositeOp() [true if operation is Paste, Cut, or Replace]. These are used to implement the logic of the state machine, and most of the conditions in Table 1 are checked by using these functions.