**Title:**

Exploration of a Bayesian Updating Methodology on the Data in a Newly Developed Computer-based Assessment

**Abstract:**

In a computer-based writing assessment, keystroke log data capture a student’s writing process (i.e., their typing behavior) as the student responds to a writing assignment prompt. Quantification of the writing process may contribute to establishing a writing profile for each student and influence the ongoing writing instruction. In this research, we have (1) designed a computer-based assessment to capture students’ writing process; (2) administered the assessment to students to collect the first-hand data; (3) parsed the raw data into formatted data which include pause events; and (4) modeled the formatted with a hierarchical mixture model using a Bayesian updating methodology. The highlights of our work are twofold. Firstly, we provided the source code for designing the sensor and the parser. Therefore, if future researchers intend to develop their own keystroke logging system, they can modify our callback functions, instead of creating the code from scratch. Secondly, the hierarchical mixture model, with Bayesian updating included, offers benefits over non-hierarchical alternatives. Because the standard error of the mixture component parameters is based on the number of events from that mixture in the sample, a non-hierarchical mixture fails to produce desirable estimates when the number of observations assigned to one of the components is small. Under the Bayesian model’s assumption (*exchangeability*), essays (students) are treated as similar data units, meaning that the model *borrows strength* adaptively from other essays to enhance the efficiency of estimation for shorter essays. We establish the advantage of borrowing strength, resulting from information gathered from other essays in computing posterior distributions.

***Index Terms*—Bayesian models, cognitive, computer-based assessment, process data.**

This work was supported by Shanghai Jiao Tong University under Project AF3500015. *(Corresponding author: Tingxuan Li.)*

Tingxuan Li is with School of Education, Shanghai Jiao Tong University, Shanghai, 200240, China

(email: litingxuan@alumni.purdue.edu)

Junjie Luo is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA

(email: luo330@purdue.edu)

Tingxuan Li is currently an assistant professor in the School of Education at Shanghai Jiao Tong University (Shanghai, China). She received the PhD degree in educational psychology and research methodology, College of Education, Purdue University (West Lafayette, Indiana, USA). Her research interests include psychometrics, Bayesian statistics, assessment design, and educational innovation. She is a member of the American Educational Research Association as well as the American Statistical Association.

Junjie Luo is currently pursuing the PhD degree in the School of Electrical and Computer Engineering at Purdue University (West Lafayette, Indiana, USA). He obtained his Master of Science degree in computer and information technology at Purdue University in 2020, Bachelor of Science degree in computer science at Sun Yat-sen University in 2018. His research interests include optics and computer vision.

I. INTRODUCTION

In educational research, discussions on student outcomes are often centered around writing instruction, writing assessment, and writing competency. To measure a student’s writing competency, the traditional (pencil–paper based) assessment offers a holistic score provided by a human rater (e.g., a teacher). In a computer-based writing assessment, automated essay scoring (AES) systems can generate a set of analytical scores to assess students’ writing competency. The analytical scores are (1) features reflecting grammatical and spelling errors (e.g., error counts normalized by the essay length), and (2) features based on discourse length and syntactic variety (McCarthy *et al*., 2022). The AES systems essentially rely on natural language processing, along with recently developed machine learning techniques (Bhatt *et al.*, 2020). A few relatively mature AES engines, for example the *e-rater* (Attali & Burstein, 2006) , *IntelliMetric* (Elliott, 2003), have been integrated to advance instruction and learning (see a review paper by Strobl *et al.*, 2019).

Human scoring and the AES analytical scores have been used together to measure students’ writing competency. These methods evaluate linguistic and discourse patterns based on submitted essays, which in a sense are *writing product* measures. In recent years, in order to gain greater knowledge about students’ writing development, researchers have developed methods to quantify the *writing process*. This quantification of the writing process may contribute to establishing a writing profile for each student. Such profiles consist of information collected from both the product and process of writing, which together represent a student’s learning status. Quantification thus provides richer information to influence the ongoing or future writing instruction (Medimorec & Risko, 2017).

Within this scope, a log file in a computer platform stores time-stamped action sequences to facilitate later analysis. Keystroke log data can capture a student’s writing process (i.e., typing behavior) as the student responds to a writing assignment prompt (Chukharev-Hudilainen, 2014). A primary challenge in analyzing keystroke log data is extracting meaningful features or variables. Almond *et al.* (2012) classified log data into pause time event observations and fitted the dataset to a mixture of lognormal models. Li (2021) further examined the dataset and found the two-component mixture of lognormal models seemed sufficient to fit the data. The estimated parameters were consistent across two writing genres that are a persuasive writing prompt and a literary writing prompt. In the mixture model, the Expectation-Maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977) was used. The EM algorithm is an iterative method for constructing “local ascent” of the log-likelihood function. It gives parameter constraints automatically without requiring the specification of a step size.

To expand the previous state of knowledge on the topic, in this present research, we set up a hierarchical mixture model in Bayesian updating methodology. Within the context of writing assessment, each student submits an essay online, making the writing process become a nested data structure. Namely, log data are nested for each student (essay); the parameters of individual students are random variables drawn from *higher-level population distributions*, that is, *prior* distributions. The individual students’ estimates are shrunk towards the grand mean. This modeling approach is termed as *partial pooling* (Gelman *et al*., 2013). In contrast, in the previous research (i.e., Li, 2021), students were considered as independent data units, which led to the estimates on each student separately. That modeling approach is often referred to as *no pooling* (Gelman *et al*., 2013). Moreover, the algorithms used in Bayesian updating methodology are Random-walk Monte Carlo (RWMC) methods. The RWMC methods are part of Markov Chain Monte Carlo (MCMC) methods. A key idea is that a sample from a probability distribution, formed by constructing a Markov chain, has the same desired distribution as its equilibrium distribution.

In this present research, we aim to (1) design a computer-based assessment to capture students’ writing process, (2) administer the assessment to students to collect the first-hand data, (3) parse the raw data into formatted data which include pause events, and (3) illustrate that a Bayesian hierarchical mixture model can provide interpretable statistics regarding the writing process. Hierarchical mixture models offer several potential benefits over the non-hierarchical alternatives. Because the standard error of the mixture component parameters is based on the number of events from that mixture in the sample, the non-hierarchical mixture fails to produce desirable estimates when the number of observations assigned to one of the components is small (e.g., see details on Li, 2021). This may happen when the value of one of the mixing probabilities is small, so that the number of observations associated with that component is small. Borrowing strength is a well-known virtue in Bayesian updating methodology (Priebe, 1996; Mesquita *et al.,* 2020). Under the hierarchical model’s assumption, *exchangeability*, essays (students) are treated as similar data units. Shorter essays contain less observations, which are most in need of improved inference. The model borrows strength adaptively from other essays to enhance the efficiency of estimation for shorter essays.

The highlights of our work are twofold. Firstly, we provide the source code for designing the sensor and the parser. In the future, if researchers intend to develop their own keystroke logging system, they can modify our callback functions, instead of creating the code from scratch. In the existing literature, the up-to-date source code for developing keystroke logging systems is very rare. Most studies failed to document the details on the design of the computer-based assessment, or the details on how their programs capture students’ writing behaviors. This is probably the case because their keystroke logging systems were for commercial use developed by a company (e.g., Guo *et al.,* 2018) or other copyright concerns (e.g., Uto *et al*., 2020).

Secondly, to model pause events, we endorse the idea that the cognitive models should be a foundation. Any identified features or unique patterns can thus be tied back to the cognitive model, for the sake of interpretability. Time-related variables (e.g., reaction time, response time, or pause observations) and cognitive processes have been examined together in many research settings, probably because these variables are “believed to be a good indicator of the speed and efficiency of mental processes” (Draheim *et al.*, 2019, p. 508). For example, Zhang *et al*. (2018) examined human operator attention in manual control tasks. They found human reaction delay time could be explained by Gamma distribution, where the shape and scale parameters might be used as extracted statistics. White and Staub (2012) studied eye movement during reading. These authors recorded the fixation time (i.e., fixation duration) when the participants were presented with different types of text. They found participants fixating longer when the entire sentence was faint, compared to when the sentence was presented normally. They also explored whether the fixation duration data followed *exponentially modified Gaussian* distribution.

Rouder *et al.* (2003) examined the response time data on *central process* and *peripheral process*. They found that the three-parameter Weibull distribution, with parameters for shape, scale, and shift (initial value) fit data reasonably well. The highlight of the study is that the assumptions of statistical modeling were informed by the cognitive theory of perception. Specifically, this means that (1) the latency of peripheral processes has little variability; and (2) the latency of central processes is variable and skewed. Likewise, in the brief review below, we also present how the existing literature on cognitive model of writing informs the modeling assumptions made in the present research.

II. RELATED WORK

A*. Keystroke Data Analysis Methods*

Keystroke log data can convey information about where a student has paused and how long (measured by milliseconds) the student has paused during text production. Pause events typically are classified according to the linguistic contexts (i.e., *text boundaries*, Medimorec & Risko, 2017). For example, if a student paused when he/she was spelling a word, this event observation belongs to the “*within a word”* linguistic context. If a student pauses between words, this observation belongs to the “*between words”* linguistic context. A single pause event can only be classified into one linguistic context. A set of studies employed various methods―for example, descriptive statistics, data reduction techniques, and machine learning methods―to analyze pause events. Pause frequencies between words during writing are meaningful across different writing tasks (Conijn *et al*., 2019). The underlying dimensionality of the pause events has been identified using principal component analysis (Baaijen *et al*., 2012; Zhang & Deane, 2015; Bennett *et al*., 2022).

In addition, Guo *et al.* (2019) classified writing process data into three states: text production, long pause, and editing. They used semi-Markov processes to model the sequences of writing states, comparing the state transition time and probability for demographic subgroups (e.g., race, gender, and socioeconomic status). Guo *et al*. found that the subgroups employed different processes in writing. For example, lower socioeconomic status students and the Black students showed smaller magnitudes on the ratio of the number of characters in the final essay to the total number of keystrokes produced in the process.

Uto *et al*. (2020) used a time- and learner-dependent hidden Markov model to analyze the writing process. They identified multiple latent states in the Gaussian hidden Markov model (GHMM) as subtasks, and incorporated parameters that express state appearance probabilities for each time interval for each student. Uto *et al.* also found that students with high writing competency and low writing competency displayed different magnitudes on the ratios of states. Sinharay *et al*. (2019) employed a regression tree model where the writing process features and writing product features were the predictors. The product features were obtained using the *e-rater* technique in AES. The process features included many variables, for example, *between words pause*, that was represented by the mean duration of pauses between words for a given essay. In line with the current research trends in log data analysis, Sinharay *et al*. pointed out the need of a variety of methods to “help explain the complex writing processes and validate the variables extracted from the writing processes for educational purpose” (p. 134).

B*. Cognitive Models of Writing*

Writing requires tremendous cognitive resources compared to other language skills (e.g., reading). The act of writing demands multiple fine-grained mental operations that interact recursively and involves effortful cognitive processes which may occur in any order during text production. Researchers have proposed a series of cognitive models to reveal “the hidden decision-making lying behind what seems like a spontaneous process” (Galbraith, 1999, p. 139). This perspective on writing (“cognitive allocation”) indicates how students allocate their cognitive resources during time spent writing (Graham & Perin, 2007). In the oft-utilized model proposed by Flower and Hayes (1981), for instance, three main cognitive processes may occur in any order: planning, translating, and reviewing. The planning stage involves idea generation; translating is about forming a tentative text by specifying the conceptual structure from the previous stage; and reviewing involves modifying the text and making decisions between the written text and the intended text, in the writer’s mental operations.

Leijten *et al*. (2014) refined the cognitive model by highlighting the role that the “*search for content*” plays. When planning or reviewing during text production, students tend to search for what to write and how to unfold it in a coherent way. In their study, they identified activities where students routinely use external sources (e.g., online dictionary). Abdel Latif (2021) further pointed out it is necessary to explicate *searching for content* and *monitoring* in the cognitive model of writing. “Searching for content” is defined as identifying the ideational content needed for text production, by retrieving it from memory. “Monitoring” refers to the activity of adjusting the information retrieved from memory. Moreover, Deane *et al.* (2011) proposed a multi-layer cognitive model of writing, which specifies a set of layers in the low-to-high cognitive processes. For example, lexical/orthographic skill is part of a lower cognitive process, whereas verbal/textual skill is thought to reflect higher cognitive processes.

Historically, researchers employed a variety of methods to gather empirical evidence. By using retrospective interviews, think-aloud cognitive lab verbal protocol data, or video observations on hand movement, researchers suggested that time-related efficiency can convey information about composing strategies and the speed of lexical retrieval while writing (e.g., Matsuhashi, 1981; Perl, 1979).

As mentioned above, cognitive models have granted possible theoretical rationales to the writing process. Particularly, writing involves multiple cognitive processes that interact recursively. Subsequently, the assumptions made in this present research include: (1) any observed pause event can be drawn from one of the cognitive processes; (2) the mixture components are tied back to cognitive processes employed in writing; and (3) a longer pause comes from higher-level cognitive processes (e.g., search for content), whereas a shorter pause comes from lower-level cognitive processes (e.g., orthographic skill such as word spelling).

The rest of the article is organized as follows. In Section 3, we describe the design principle of a computer-based writing assessment. It provides the details on how keystroke log data have been classified into pause events, and the details of the proposed model in estimating the parameter for each student (data vector) in the Bayesian updating methodology. In Section 4, we report the estimation results. In the last section, we further interpret the results with the connection of the existing literature.

III. METHODS

A. *Design of a Computer-based Assessment*

The computer-based writing assessment mainly contains the frontend and the backend. Fig. 1 (below) maps the design principle for this present research. In the design component for the frontend, we (developers) create the website and the applications inside it, using HTML/CSS and JavaScript. In the backend, we set up a server to receive data captured by the frontend.

The design principle contains three layers. Students’ writing behaviors can trigger the callback functions written in the JavaScript program, after which the program records what a student’s writing behavior and converts the behavior into raw data. Subsequently, some items (e.g., questions, tasks, writing prompts) in the assessment platform are received from the server or raw data captured by the JavaScript program, which are temporarily stored in the buffer area of local computers. Then, the raw data are sent to the server after students finish their writing. To avoid long waiting times or to compress the data that will be sent to the server, we only send raw data and will do computation later. After obtaining raw data, we use a parser to format the raw data and model the data. Finally, for the hardware, we rent a Cloud Desktop that is sufficiently powerful to send items and receive data, while 2000 students may write essays at same time.

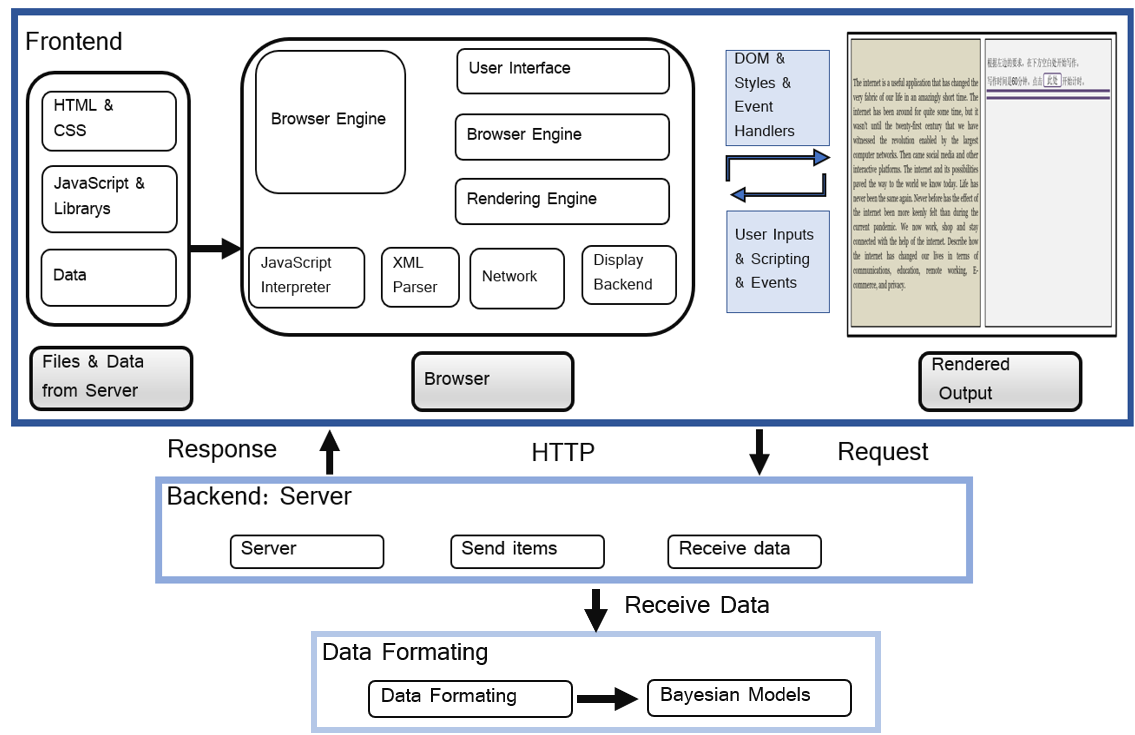


Fig. 1. Design principle of the computer-based assessment [JJ, check this figure]

Place “Check ID” between “Server” and “Send Items”

For the sake of testing formatting algorithms and reducing the server load, we distinguish tasks by processing data. The backend only does the following tasks: 1) working as the server of our platform, so that students can access the website; 2) sending items to students according to their student IDs; and 3) receiving raw data from students. With the raw data, we can structure the raw data to the formatted data, which includes the information by further computations. The purpose of formatting data is to import the data into our Bayesian models for later analysis.

B. *Participants and Data Collection*

The computer-based assessment platform developed in this research presents a writing prompt to students, see Fig. 2 below. The left panel is the writing prompt; the right panel is the typing area. The participants were 87 students who were enrolled in sophomore year at a research university in Mainland China. All participants self-reported that they had tremendous experience in using a keyboard to complete writing tasks. They also self-reported that they have high proficiency in English language skills. All of them speak Chinese as their first language. 47 of them are female; 40 of them are male. The writing prompt, “The Internet and Our Life”, used in this research, was designed by a professor in the department of English at the university. On the left panel, there was a brief introduction to the topic and a prompt question. Students are required to write their opinion on the right panel within 60 minutes. Moreover, students completed the essays voluntarily, that is, the grading of the essay did not affect the student’ final grade.

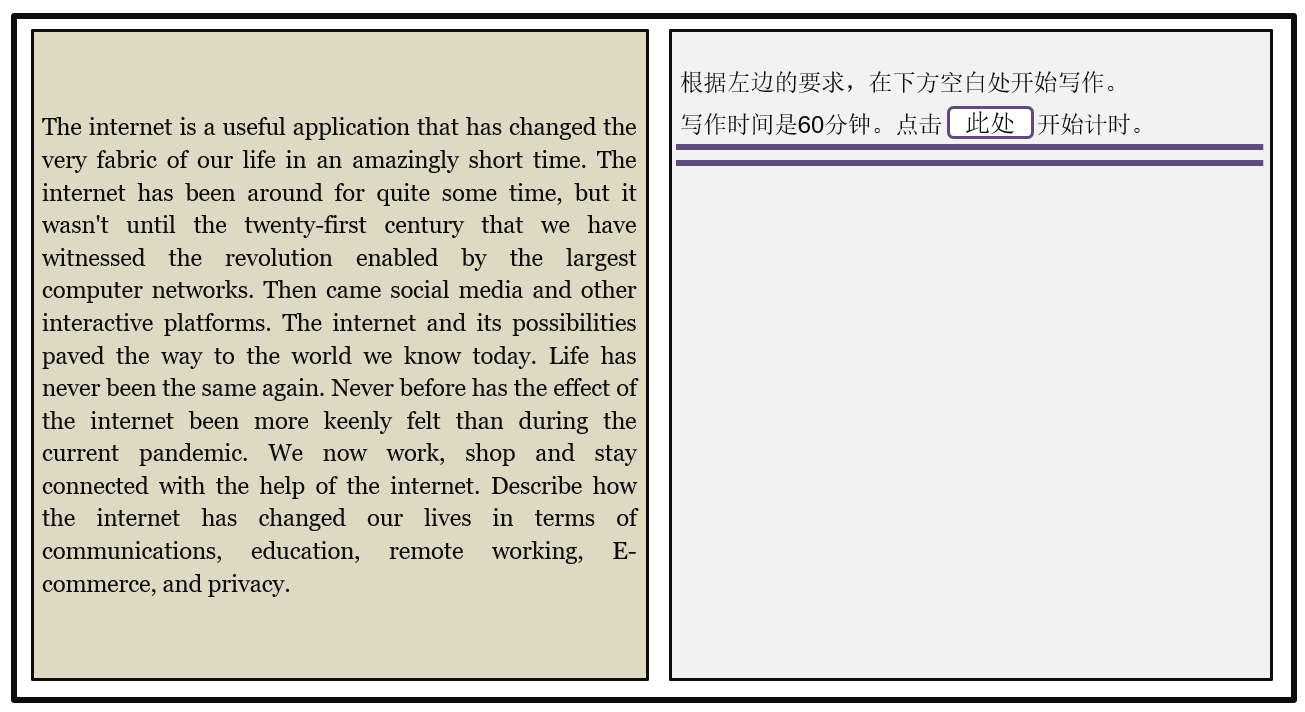


Fig. 2. The writing prompt on the computer-based assessment [JJ, do something on figure 2]

Essay scoring was completed by two English professors in the university. The human raters used a holistic rubric that covered two content categories: 1. adopting a stance with regard to a central topic, and then defending that position via evidence or examples; 2. general coherence and cohesion. Each essay was independently scored by two raters on a 1–6 scale, the interval is increased with 0.5 point. Moreover, we add up the scores given across the two raters, resulting in scores ranging from 2 to 12. The percentage of agreement between the two raters was 97%. Using the Krippendorff’s alpha as an inter-rater reliability index (Hayes & Krippendorff, 2007), the magnitude of the index on this sample was 0.92.

C.  *Pause Event Classification*

The keystroke logging function is implemented in JavaScript and works on the assessment platform, developed in Java. The obtained keystroke log data are in XML format. After the student completes his/her writing task, the frontend application converts these instances to an XML file, in order to store the structured data. The data also contain other information (e.g., student ID). Then, the data are sent to our server that converts the data back, to the instances of the class variable, and reconstructs how the student has typed the essay.

We classify the raw data into various pause events according to the linguistic contexts (e.g., within a word pause, or between words pause). The source code link is shown on the Appendix attached in this paper . Fig. 3 below shows a simple example of a log of keystroke activities when typing a sentence “It is a pen”. The time stamps allow us to compute the length of the pause (in milliseconds), according to the linguistic contexts.

A screenshot of a computer

Description automatically generated with medium confidence

Fig. 3.Typing behavior reflected on a log of keystroke activities

By classifying the operations along the order of their time stamps, we can convert the data from instances of a key-based variable to instances of a word-based variable, which means that we know how the student typed each letter . Taking the word “pen” as an example, the values of keys in the variable are “p”, “e”, and “n”. Each instance of variable has its own time stamp. The application makes them into one, as “pen”, and combines their information, thereby creating attributes “pen”, and “Operation Type” for it. We can reconstruct the whole essay and map the time stamp to each character of the essay. Then, we can compute the pause between words, sentences, and the pause within a word. With the information of “Input Type”, “Key” and “Time”, the formatting script can simulate the process of student’s writing. After reconstructing and mapping, we can compute the parameters that we import into our model. For example, for the *between word* linguistic context, “She studies” as an example. , where is the time stamp of “e” in the word “She” and is the time stamp of “s” in “studies”. Table 1 below is the various events collected in this research.

TABLE 1 [JJ, think about table 1]

EVENTS IN VARIOUS CATEGORIES

|  |  |
| --- | --- |
| Event | Description |
| Within a word | Before the student types a letter in a word. |
| Between word | Before the student types a white space between words in a  sentence. |
| Between sentence | Before the student types punctuation. |
| BEGIN | When the student clicks the “START” button. |
| END | When the student clicks the “SUBMIT” button. |

D*. Proposed Model in Bayesian Updating Methodology*

Mixture models are useful when observations are taken from two or more different populations (Biernacki *et al.,* 2000). Each population is termed as a *mixture component* in the model. In other words, a random variable is drawn from a population which consists of *K* components; meanwhile, *k* is the index for the mixture component, which can take a value between 1 and *K* (McLachlan & Peel, 2000). The probability density function (sampling distribution) of the mixture model with *K* components for a *d*-dimensional random variable is given by:

(1)

Usually, mixture components are assumed to follow the same parametric family (e.g., all normals, all Poissons) with different parameter vectors. The *mixing proportion* parameter is , which is the proportion of the population from the mixture component *k*. The constraint of is . The model parameters are , where = { }. In practice, the component membership typically is unknown.

In the present research, it is unknown from which cognitive process the observed pause event is drawn. The unobserved variable is termed as a *latent mixture indicator*. In this case, let the vector be a categorical random variable/vector, taking on the value *1,…, K*, with probabilities *,…, .* The observed pause events are incomplete data unless the associated cognitive process labels are given. One common approach is to fit the mixture models for small values of *k* (i.e., *k=*2, 3*,…, where k*=1 is not considered because it is not a mixture distribution). In mixture of lognormal distributions, random variables, or the pause events, are represented as the log scale:  = log(. In this research, the proposed model can be reflected on the plate notation (see Fig. 4 below).

Fig. 4 provides a graphical representation of a hierarchical mixture structure in Bayesian updating methodology. In this structure, we denote each essay *i, i* =1, 2, 3,…, *I*. Hence, the observed variable on log scale = is a random vector with observed pause events nested in each essay *i.* Also, is the latent mixture indicator variable that follows the categorical distribution: ~ cat(). If = *k*, then .The event *n* comes from a particular component, indicating that ~ *N* ().

As shown in Fig. 4, the hierarchical model contains two levels. The first level includes pause events, where *n* {1, …, }. The second level is higher level, that is the essay level, where each essay *i* {1, …, *I* }. The parameters of individual essays are random variables drawn from higher-level population distributions. Therefore, all the mixture model parameters are now also indexed by *i:* isthe mean of the *k*th component for essay *i*. The variance of the *k*th component for essay *i* is denoted as . The mixing proportion for essay *i* is  *=* {}. Thus, the parameters of interest for essay *i* are where includes In addition*,* because little information is known about ,, , , **α**, non-informative priors are assigned for these parameters. The conjugate prior distributions are used, that is, is drawn from the normal distribution; is drawn from gamma distribution; is drawn from Dirichlet distribution.

Diagram

Description automatically generated

Fig. 4. The plate notation for the proposed model

The foregoing philosophical approach to the model setup in Bayesian updating methodology contains a series of steps. The first step involves probability specification of the model, as described above. In practice, the models are large (e.g., they include many parameters, high dimensional parameter structure, or complicated probability specifications). In this research, the two-component mixture of lognormal model contains 5 parameters for each essay, that is, the mean of the low component, the mean of the high component, the variance of the low component, the variance of the high component, and the mixing proportion. Thus, for 87 essays, the number of parameters of interests is totally 435. In addition, the mixture model is not identifiable. So, the researchers need to define the mixture component (McNicholas, 2016). In this research, the larger magnitude of the mean is defined as “high component”, because the higher-level cognitive process should take longer time. The smaller magnitude of the mean is thus the “low component”.

The second step involves computing (most often simulating random draws from) the posterior distribution of the parameters of interests, through MCMC algorithms. The third step involves the inferences which are summarized by random draws from the posterior distribution of the model parameters. The objective of the MCMC algorithms is to obtain a representative sample of the posterior distribution. Hence, *convergence* means the MCMC sampler has explored a posterior distribution to obtain the samples (Golden, 2020).

E. *Metropolis-Hastings Algorithm*

The computation on posterior distributions relies on the Metropolis-Hastings algorithms that are part of the RWMC methods. The fundamental underlying idea is the use of an ergodic Markov chain with stationary distribution . For an arbitrary starting value , a chain is generated using a transition kernel with stationary distribution , which leads to the convergence in a distribution of () to a random variable from . The Metropolis-Hastings algorithm implies that the use of a chain () with stationary distribution is similar to the use of an iid sample from where the ergodic theorem ensures the convergence of the empirical average, as shown in (2) below.

(2).

A single Markov chain is sufficient to ensure a proper approximation through estimation of for the functions of interest. The algorithm below shows that the density and the conditional density produces a Markov chain through the transition:

|  |
| --- |
| **Algorithm 1** |
| ~  Take  =  {  with probability  with probability 1-  },  where  = min 1}. |

Furthermore, within this framework, the Gibbs sampler is a special case of the Metropolis-Hastings algorithm. It generates the posterior samples by sweeping through each variable to sample from its conditional distribution with the other variables fixed to their current magnitudes. Specifically, the Gibbs sampler obtains a sample , …, , approximately distributed from without directly simulating from . In this setting, is called the proposal distribution. This algorithm accepts values such that the ratio /is increased, compared with the previous values /( *.* For the random variable **X**, **X** = (, …,), where *i*=1,2,3,…*p*, the simulation can be made from the corresponding univariate conditional densities ,…, ; that is, to simulate:

~ (,).

The associated Gibbs sampling algorithm is given by the following transition from to :

|  |  |
| --- | --- |
| **Algorithm 2** |  |
| Initialize *x*(0) | |
| **for** iteration *p* = 1*,*2*,...* **do** | |
| ~ ( , …, ) | |
| ~ ( , ) | |
| ….. | |
| ~ ( ) | |
| **end for** | |

The densities are called the *full conditionals,* and the advantage of the Gibbs sampler is that these are the only densities used for simulation. Thus, even in a high-dimensional problem, all of the simulations may be univariate.

IV. RESULTS

A. *Descriptive Statistics*

For illustration purpose, we modeled the pause data from the *within word* linguistic context. Other linguistic contexts (e.g., *between words* context) had similar modeling results (see details on Almond *et al*., 2012; Li, 2021). Across 87 essays, the mean of the number of sentences is 31. In addition, for each essay, we computed the number of pause events and the human score. Table 2 below shows how these variables are distributed across 87 essays.

TABLE 2

DESCRIPTIVE STATISTICS

|  |  |  |
| --- | --- | --- |
|  | Number of  Pause Events | Human  Score |
| Minimal | 703 | 6.5 |
| 1st quartile | 817 | 7 |
| Median | 939.5 | 8 |
| Mean | 993.1 | 8.2 |
| 3rd quartile | 1125.8 | 9 |
| Maximal | 1555 | 11.5 |

For the pause data, we put data onto the log scale. Then, we computed the kurtosis and skewness for each essay. Fig. 5 below is the scatterplot for the kurtosis and skewness across 87 essays. The data vectors (essays) possess high kurtosis and positive skewness, even after putting them on the log scale. The normal distribution has kurtosis value of 3 and skewness value of 0. In Fig. 5, most essays have heavy tails (high kurtosis) or bimodals, which are the typical shape of a mixture density. For illustration purposes, we chose four essays to show the distributions using box plot, see below on Fig. 6.

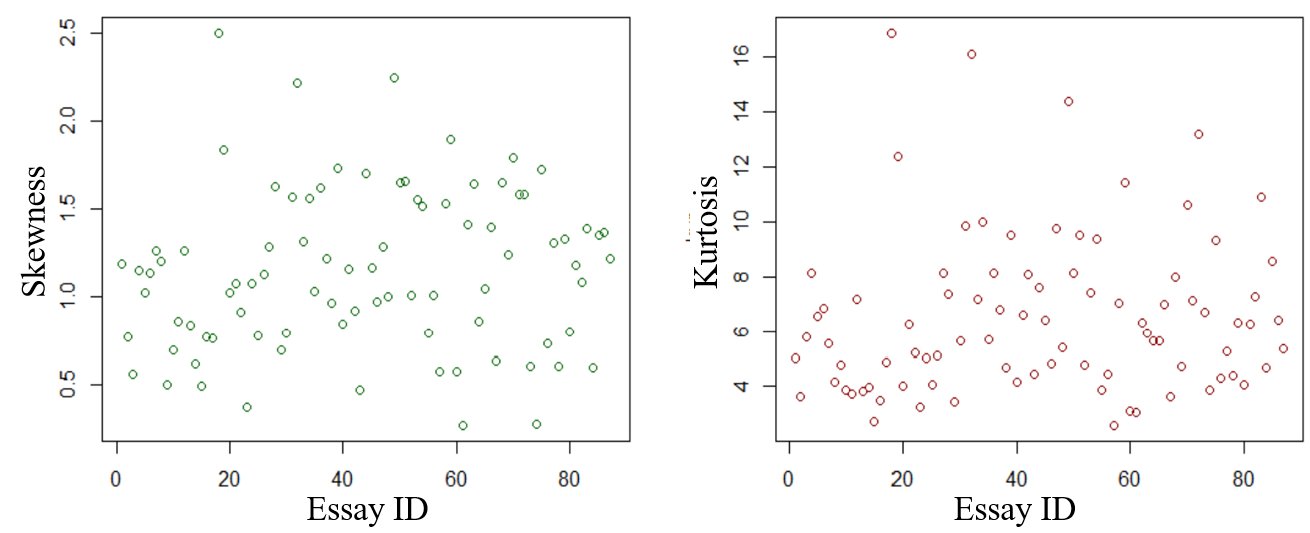


Fig. 5. Kurtosis and skewness across 87 Essays

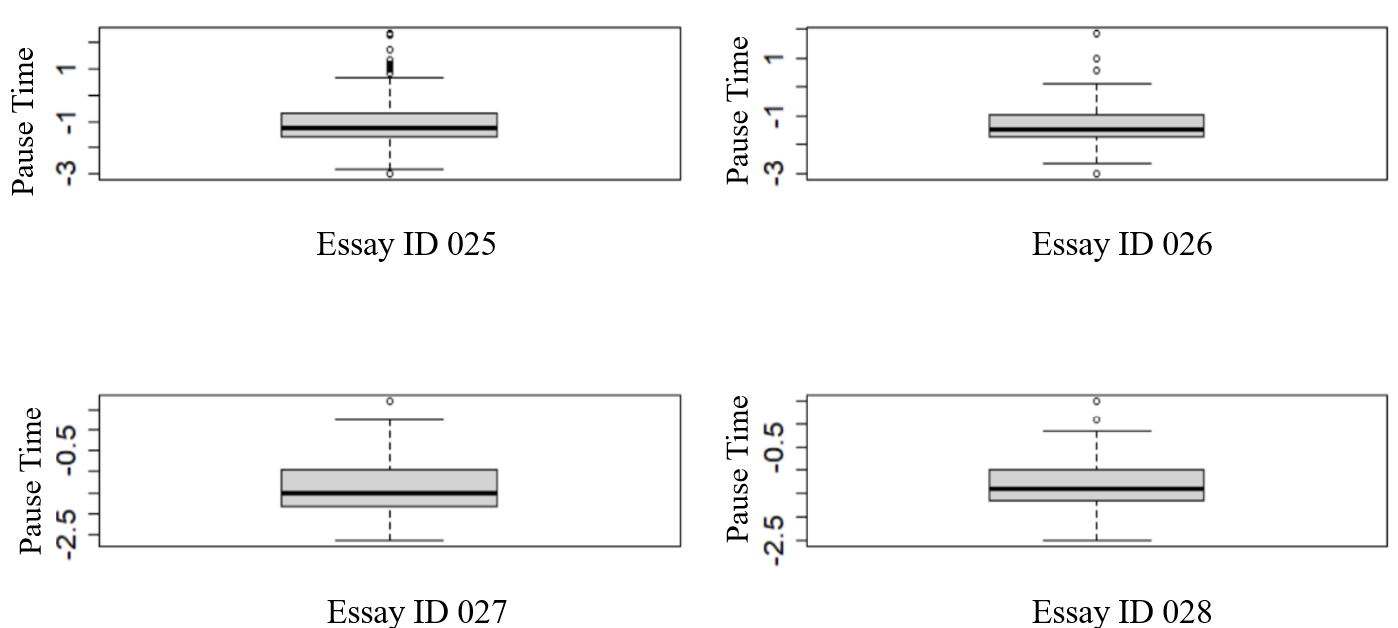


Fig. 6. Pause times for four essays

B*. MCMC Convergence Diagnostics*

For the model estimation, we employ a set of the MCMC diagnostics tools to present the chain convergence results. If the model has converged, the samples from the conditional distributions are used to summarize the posterior distribution of parameters, in this case, the mean, variance, and the mixing proportion for each essay (mentioned above in Fig. 4). Convergence refers to the idea that the Gibbs sampler has reached a stationary distribution. The diagnostics tools show the details of the sampler convergence. We used the R2jags package (Su & Yajima, 2013), to analyze JAGS (Plummer, 2022) programming output in R environment (R Core Team, 2022). The specific code used in this research is shown in the Appendix attached to this article. The Gibbs samplers are implemented in JAGS program in estimating the parameters. Multiple chains are often used to check MCMC convergence. In this research, we specified 2 chains in JAGS.

The R-hat ( ), also known as the potential scale reduction factor (PSRF) was proposed by Gelman & Rubin (1992) as a method of computing the convergence of parameters given 2 or more chains. It is a numerical value to reflect the convergence summary. It is the ratio of within variance to between chain variance. If the chains mixed well, the value should be around 1. In this research, most of values are around 1. Fig. 7 is across 87 essays. It indicates that the Gibbs sampler has reached a stationary distribution. The red horizontal line is the reference line that is equal to 1.

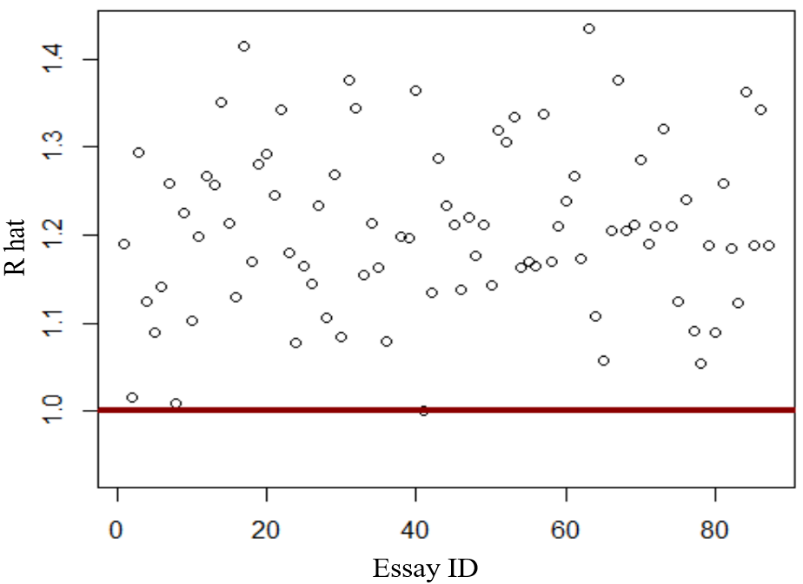


Fig. 7. magnitudes across 87 essays

Because samples from the early iterations are not from the target posterior, it is common to discard them. The discarded samples are termed as the “burn-in” period. We specified the “burn-in” value as 100 in JAGS in this research. Traceplot provides visual inspection for the parameter value at iteration *t* against the iteration number. It reflects sampling behavior and evaluates mixing across chains and convergence. If the model has converged, the traceplot will move around the mode of the stationary distribution (the *y*-axis). Meanwhile, kernel density plots are the empirical density plots of the posterior distributions of the model parameters. If the MCMC sampler is smooth and well-behaved, the kernel density plots should look smooth and unimodal. In this research, the traceplots show that the chains appeared to be around the stationary distribution (*y*-axis) mode with very small fluctuations. The kernel density plots look smooth and unimodal. These indicate that the chains reached stationary distributions. For illustration purpose, Fig. 8 below depicts the traceplots and kernel density plots for the two parameters; that is, the mean of the low component and the mean of the high component for Essay ID 039.

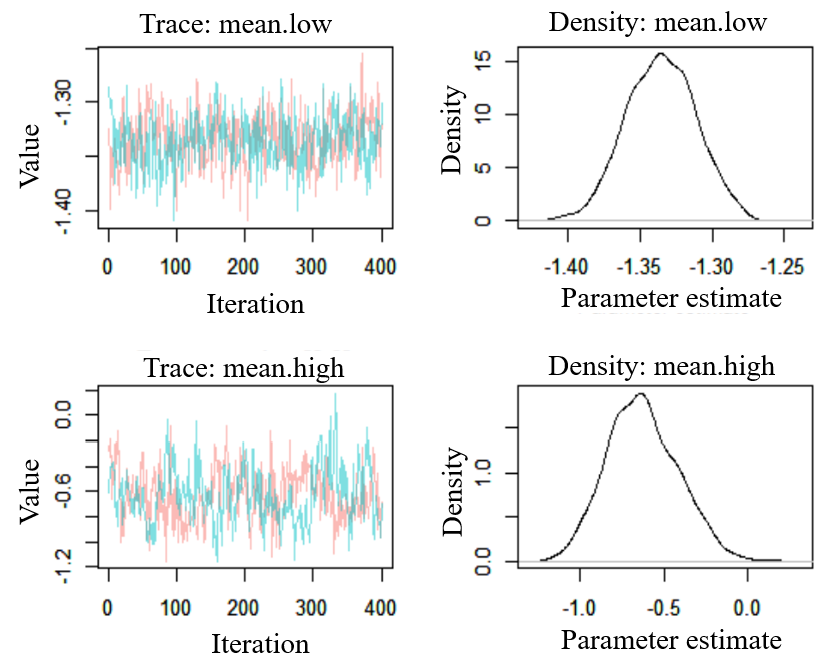


Fig. 8. Traceplots and kernel density for two means in Essay ID 039

Furthermore, the degree of autocorrelation in the samples is also examined as part of MCMC convergence diagnostics. The simulated value of θ at the (*t* + 1)st iteration is dependent on the simulated value at the *t*th iteration. A high correlation between successive iterates prevents the algorithm from exploring the entire region of the parameter space. A standard statistic for measuring the degree of dependence between successive draws in the chain is the autocorrelation. Autocorrelation plot is an important diagnostic tool of the MCMC sample convergence. In this research, no high autocorrelation is found across all parameters. For illustration purpose, Fig. 9 below is an autocorrelation plot for the mean of the high component, for Essay ID 039.

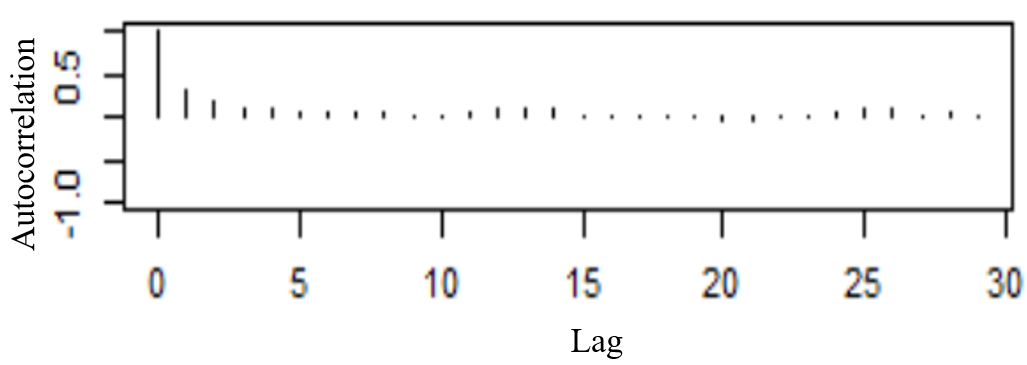


Fig. 9. Autocorrelation plot for the mean of high component for Essay ID 039

C. *Correlation Analysis*

In addition, we conducted a series of correlation analysis for a set of variables: (1) human scores, and (2) the mixture parameters. Human scores reflected the quality of each final writing product. The mixture parameters refer to the estimated mean, variance, and the mixing proportion for each essay. For each parameter, we used the mean of the posterior distribution to represent the estimation. Table 3 below indicates that there is a correlation between the human score and the mixing proportion parameter in the low-cognitive component (). Note that (the mixing proportion parameter in the high-cognitive component) is not reported, because is simply equal to 1- . The confidence interval is presented in parentheses. This implies that students who spend more time on the lower-level cognitive processes tend to have lower human scores. Also, longer pauses at the low component is reflected in the larger mean in that component. Longer pauses may indicate inefficient low-level cognitive processes, which may also impose on working memory resources.

TABLE 3

CORRELATION AMONG VARIABLES

|  |  |
| --- | --- |
| Variable | Correlation to human score |
| mean.high | **-0.34 \*\***  [-0.51, -0.13] |
| mean.low | **-0.15**  [-0.35, 0.06] |
| tau.low | **-0.24**  [-0.43, -0.03] |
| tau.high | **-0.05**  [-0.25, 0.17] |
| pie.low | **-0.35 \*\*\***  [0.15, 0.52] |

*\*\*\*p < 0.01; \*\*p < 0.05.*

Fig. 10shows the correlation plots among variables. A complex relationship between the mixing proportion parameter and the human score is found in the plot. As shown in Fig. 10, the correlation is stronger at the upper bound of the plot. In other words, those students with low human scores have a wider range of values across the mixing proportion parameter; the students with higher human scores do not possess such a wide range of values in the mixing proportion parameter.

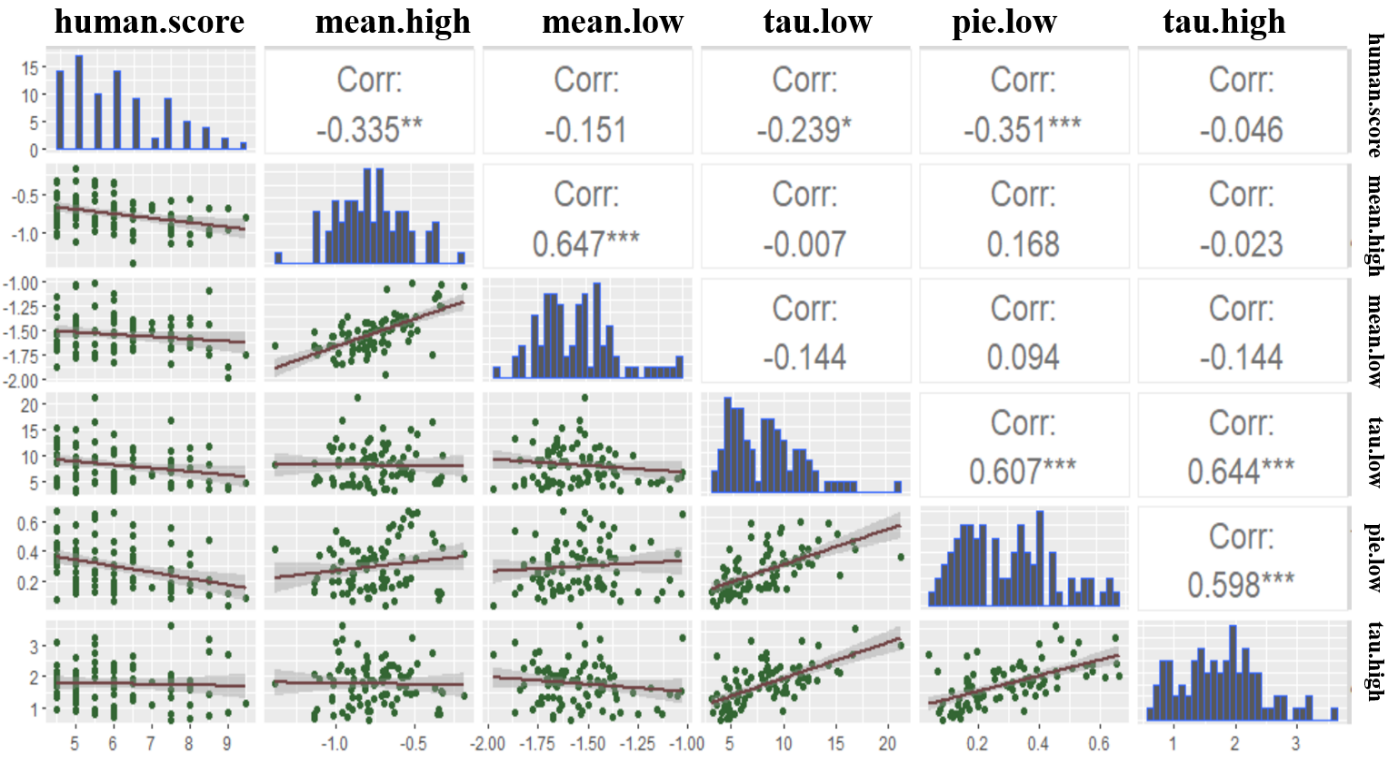


Fig. 10. Correlation plots among variables

V. DISCUSSION

In this section, we discuss our research results in connection to the existing literature. In this article, we reported the design of the assessment in which the keystroke logging system captures students’ writing process. The callback functions, along with the methods of parsing the raw data, are also reported in the source code. In the literature, a series of typing behaviors has been identified. For example, *jump editing* occurs whenever a student moves further away from the current word to a new location for editing purposes. *Between sentences* also names an observed pause category, that reflects how long a student has pause at the location between sentences. But this sort of behaviors was relatively rare, and relatively difficult to model (Guo *et al*., 2020; Almond *et al*., 2012). Similarly, in this research, the mean number of *between sentence* pause events across 87 essays is merely 31. With too few observations, and too many parameters, the estimation results would not possess the desirable statistical stabilities. Thus, we chose to focus on the pause event that occurs more often during writing. In particular, we modeled the *within a word* pause category. In the source code, we also provide the details on how the raw data are classified as the *within a word* category.

The pause event data often show heavy tail or bimodal shapes (e.g., Almond *et al*. 2012). Subsequently, researchers (e.g., Guo *et al*., 2020) have used heavy-tailed distribution (e.g., Stable distribution) to model the data and to extract writing process features. The first-hand data collected in this research also show high kurtosis and skewness (see Fig. 5 and Fig. 6). We used the two-component mixture of lognormal model to fit the data. Moreover, the model took advantage of a Bayesian updating methodology. Shorter essays contain fewer observations, meaning that they are most in need of improved inference. The model we set up thus became the two-component hierarchical mixture of a lognormal model. It allows borrowing strength adaptively from other essays to enhance the efficiency of estimation for shorter essays. This modeling approach reflects the fact that the hierarchy aspect provides “an infrastructure through which one may improve the inference of a parameter of interest by borrowing from information on other related data and parameters that are also part of the model” (Nguyen, 2016, p. 2).

In terms of parameter estimation, researchers who employ MCMC algorithms in Bayesian updating methodology often decide convergence by applying empirical diagnostic tools (Robert, 2007). In this research, we used the MCMC convergence diagnostic tools that are often used in the literature (see details on Roy, 2020). In this research, we used R-hat, autocorrelation plot, traceplot, and kernel density plots. The results indicate each of the chains mixing well; in turn, this means that the quantities can be extracted from the posterior distributions as the estimation results.

The limitation of this research is that the data collection is based on one writing situation. It is unknown how the results might be generalized to other writing sessions under different motivational and cognitive constraints than those observed here. In the existing literature, researchers have explicitly stated that differences in writing conditions and samples should be evaluated. Malekian *et al.* (2019) examined the log data collected from 107 college students’ essays. Each of the essays contained about 1,000 words. The grading of the essay was part of the student’s final course grade. This writing condition may offer more pause events to model the rare mixture components.

In the keystroke log data analysis community, researchers seek the mechanisms to explain and quantify students’ writing process. Currently, studies capturing the dynamics of text production have advanced only far enough to bring explanatory possibilities to light. The findings in the literature tend to examine the association between the extracted features and human scores. Uto et al. (2020) used process data collected from 72 students’ essays to fit a Gaussian hidden Markov model. Using the parameters as the extracted features, they found that students who write fast and spend more time on revisions had a higher human score. Sinharay *et al*. (2019) built a regression tree model. They found that six product features extract from AES’ *e-rater,* along with four process features together, can better predict the human scores. Four process features were time on task, typing speed, burst number, and burst length. Guo *at al.* (2019) found that for some groups such as female students or the low SES students, given the same human score, the extracted writing process features are different from their counter parts.

This research’s findings echo back to the existing literature. The results from this research are interpretable because the mixture model parameters are estimated to map onto the cognitive model of writing. Particularly, five parameters estimated from the hierarchical mixture model were used as writing process features for each essay. The correlation between these features and human scores were examined (as shown in Table 3 and Fig. 10). Students who spend more time on the lower-level cognitive processes tend to have lower human scores. Furthermore, the low-level cognitive process is the least demanding of the measured cognitive processes (Deane *et al*., 2008). Inefficient low-level cognitive processes may impose on working memory resources, which in turn hinders deeper engagement in other cognitive processes such as the *search for content* (McCutchen, 2000). In addition, those students with low human scores have a wider range of values across the mixing proportion parameter; the students with higher human scores do not possess such a wide range of values in the mixing proportion parameter. Future studies may therefore examine whether this indicates a systematic difference between these groups.

APPENDIX

Open source GitHub link:

|  |
| --- |
| ### JAGS Code and R Code  ### JAGS Model File  model{  for (n in 1:N){  compon[n]~dcat(p[ind[n],1:2])  y1[n]~dnorm(mu.low[compon[n],ind[n]], tau.low[compon[n],ind[n]])  }  for(i in 1:I){  p[i,1:2] ~ ddirch(alpha[]+.01)  mu.low[1,i]~dnorm(mu.high[1],tau.high[1])  mu.low[2,i]~dnorm(mu.high[2],tau.high[2])  tau.low[1,i]~dgamma(a.high[1]+.01,b.high[1]+.01)  tau.low[2,i]~dgamma(a.high[2]+.01,b.high[2]+.01)  }  mu.high[1]~dnorm(0,0.00001)  mu.high[2]~dnorm(0,0.00001)  tau.high[1]~dgamma(0.1,0.1)  tau.high[2]~dgamma(0.1,0.1)  a.high[1]~dgamma(1.1,1.1)  a.high[2]~dgamma(1.1,1.1)  b.high[1]~dgamma(1.1,1.1)  b.high[2]~dgamma(1.1,1.1)  alpha[1]~dgamma(1.1,1.1)  alpha[2]~dgamma(1.1,1.1)  }  ### Model setup and Parameter Monitoring  x=log(x)  N=length(x)  compon=rep(NA,N)  I=length(counts)  offset=cumsum(c(1,counts))    ind=rep(NA,N)  for (n in 1:N){ind[n]<-sum((offset<=n))}  compon=rep(NA,length(ind))  compon[which.min(x)] <- 1  compon[which.max(x)] <- 2  mydata=list(x=x,ind=ind, compon=compon, I=I, N=N)  params= c("alpha", "mu.high", "tau.high","a.high","b.high", "mu.low", "tau.low", "p")  pause.inits=function(){  list(alpha=c(0.7,0.3), mu.high=c(6,8), tau.high=1/c(12,14)^2,  a.high=c(1,1.1),b.high=c(1,1.1))  }  result.monitor=jags(data=mydata, inits=pause.inits, params, n.chains=2,  n.iter=9000, n.burnin=1000, n.thin=20,  model.file="C:/Desktop/model.bug") |

ACKNOWLEDGEMENT

This work was supported by the Shanghai Jiao Tong University under project number AF3500015.

REFERENCES

McNicholas, P. D. (2016) Mixture Model-based Classification. Boca Raton: Chapman and Hall–CRC [newly added]

Robert, C. P. (2007). The Bayesian Choice: From Decision-Theoretic Foundations to Computational  
Implementation. New York: Springer.

Guo, H., Zhang, M., Deane, P., & Bennett, R. (2020). Effects of scenario-based assessment on students’ writing processes. *Journal of Educational Data Mining, 12*(1),

19–45. https://doi.org/10.5281/zenodo.3911797

Roy, V. (2020). Convergence diagnostics for Markov chain Monte Carlo. *Annual Review of Statistics and Its Application, 7*(1), 387–412.

Plummer, M. (2003, 2022). Just Another Gibbs Sampler (JAGS). Retrieved from

<https://mcmc-jags.sourceforge.io/>

Su, Y-S, & Yajima, M. (2021). Package ‘R2jags’. Retrieved from

<https://mirrors.sjtug.sjtu.edu.cn/cran/web/packages/R2jags/R2jags.pdf>

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science, 4*(1), 457–472. doi:10.1214/ss/1177011136.

Elliott, S. (2003). IntelliMetric: From here to validity. In M. D. Shermis, & J. Burstein (Eds.), Automated essay scoring: A cross-disciplinary perspective (pp. 71–86). Erlbaum.

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater® V.2.

Journal of Technology, Learning, and Assessment, 4(3).

Strobl, C., Ailhaud, E., Benetos, K., Devitt, A., Kruse, O., Proske, A., & Rapp, C. (2019). Digital support for academic writing: A review of technologies and pedagogies.

Computers & Education, 131, 33–48. <https://doi.org/10.1016/j.compedu.2018.12.005>

Bhatt, R., Patel, M., Srivastava, G., & Mago, V. (2020). A Graph Based Approach to Automate Essay Evaluation," 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 4379-4385, doi: 10.1109/SMC42975.2020.9282902.

McCarthy, K. S., Roscoe, R. D., Allen, L. K., Likens, A. D., & McNamara, D. S. (2022). Automated writing evaluation: Does spelling and grammar feedback support high-quality writing and revision? Assessing Writing, 52(1), 1-21.

Baaijen, V., Galbraith, D., & de Glopper, K. (2012). Keystroke analysis: Reflections on procedures and measures. *Written Communication, 29*(3), 246–277. doi:10.1177/0741088312451108

Chukharev-Hudilainen, E. (2014). Pauses in spontaneous written communication: A keystroke

logging study. *Journal of Writing Research, 6*(1), 61-84.

Almond, R., Deane, P., Quinlan, T., Wagner, M., & Sydorenko, T. (2012). *A preliminary*

*analysis of keystroke log data from a timed writing task.* (ETS Research Report No.

RR-12-23). Princeton, NJ: Educational Testing Service.

Conijn, R., Roeser, J., and van Zaanen, M. (2019). Understanding the keystroke

log: the effect of writing task on keystroke features. Read. Writ. 32, 2353–2374.

doi: 10.1007/s11145-019-09953-8

Li, T. (2021). Identifying mixture components from large-scale keystroke log data. *Frontiers in Psychology, 12*(1), 1-11. DOI: 10.3389/fpsyg.2021.628660

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, Methodological, 39*(1), 1-22.

Golden, R. M. (2020). Statistical Machine Learning: A Unified Framework.

Boca Raton, FL: Chapman and Hall/CRC, 2020.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., & Rubin, D.(2013). Bayesian data analysis. Chapman & Hall/CRC

Priebe, C. E. (1996) Nonhomogeneity Analysis Using Borrowed Strength, Journal of the American Statistical Association, 91:436, 1497-1503, DOI: 10.1080/01621459.1996.10476717

Mesquita, D.P.P., L. A. Freitas, J. P. P. Gomes and C. L. C. Mattos (2020). "LS-SVR as a Bayesian RBF Network," in IEEE Transactions on Neural Networks and Learning Systems, *31*(10), 4389-4393, 2020, doi: 10.1109/TNNLS.2019.2952000.

Guo, H., Deane, P. D., Van Rijn, P. W., Zhang, M., & Bennett, R. E. (2018). Modeling basic  
 writing processes from keystroke logs. *Journal of Educational Measurement, 55*(2), 194–216.

Guo, H., Zhang, M., Deane, P., & Bennett, R. E. (2019). Writing process differences in subgroups reflected in keystroke logs. *Journal of Educational and Behavioral Statistics, 44*(5), 571–596.

Zhang, Y., Martinez-Garcia, M., & Gordon, T. (2018). Human response delay estimation and monitoring using Gamma distribution analysis. In Proceedings of the 2018 IEEE

International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan,

7–10 October 2018; pp. 807–812.

White, S. J., & Staub, A. (2012). The distribution of fixation durations during reading: Effects of stimulus quality. *Journal of Experimental Psychology: Human Perception and Performance*, *38*(3), 603–617. <https://doi.org/10.1037/a0025338>

Rouder, J. N., Sun, D., Speckman, P. L., Lu, J., & Zhou, D. (2003). A hierarchical Bayesian statistical framework for response time distributions. *Psychometrika, 68*(1), 589-606.

Medimorec, S., & Risko, E. F. (2017). Pauses in written composition: On the importance of

where writers pause. *Reading and Writing: An Interdisciplinary Journal*, *30*(1), 1267–

1285.

Bennett, R. E., Zhang, M., Sinharay, S., Guo, H., & Deane, P. (2022). Are There Distinctive Profiles in Examinee Essay-Writing Processes? *Educational Measurement: Issues and Practice, 41*(2), 55–69.

Zhang, M., & Deane, P. (2015). *Process features in writing: Internal structure and incremental*

*value over product features*. (ETS Research Report RR-15-27). Princeton, NJ: ETS Research Report.

Galbraith, D. (1999). Writing as a knowledge-constituting process. In M. Torrance, & D. Galbraith (Eds.), Studies in writing: Vol. 4. Knowing what to write: Conceptual processes in text

production (pp. 139-160). Amsterdam: Amsterdam University Press

Graham, S., & Perin, D. (2007). A meta-analysis of writing instruction for adolescent students. *Journal of Educational Psychology, 99*(3), 445-476.

Flower, L., & Hayes, J. R. (1981). A cognitive process theory of writing. *College Composition and Communication, 32*(4), 365-387.

Leijten, M., Van Waes, L., Schriver, K., & Hayes, J. R. (2014). Writing in the workplace: Constructing documents using multiple digital sources. *Journal of Writing Research, 5*(3), 285–337. <https://doi.org/10.17239/jowr-2014.05.03.3>

Abdel Latif, M. M. M. (2021). Remodeling writers’ composing processes: Implications for writing assessment. *Assessing Writing, 50*(1), 1–16.

Deane, P. (2011). *Writing assessment and cognition.* (ETS Research Report RR-11-14). Princeton, NJ: ETS Research Report.

Matsuhashi, A. (1981). Pausing and planning: The tempo of written discourse production.

*Research in the Teaching of English, 15*(2), 113-134.

Perl, S. (1979). The composing processes of unskilled college writers. *Research in the Teaching of English, 13*(4), 317-336.

Hayes, A., & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. CommunicationMethods and Measures, 1(1), 77-89.

Biernacki, C., Celeux,G., & Govaert, G. (2020). Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(7), 719-725. doi: 10.1109/34.865189.

McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. New York, NY: Wiley.

Uto, M., Miyazawa, Y., Kato, Y., Nakajima, K. (2020). Time- and learner-dependent hidden Markov model for writing process analysis using keystroke log data. *International Journal of Artificial Intelligence in Education, 30*(1), 271–298.

Draheim, C., Mashburn, C. A., Martin, J. D., & Engle, R. W. (2019). Reaction time in differential and developmental research: A review and commentary on the problems and alternatives. Psychological Bulletin, 145(5), 508–535. <https://doi.org/10.1037/bul0000192>