What Predicts Student Comprehension in Language Learning? Augmenting Student Action with Elapsed Time in an Educational Data Mining Approach

by

Matthew Scott Dexheimer

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2017 by the
Graduate Supervisory Committee

Erin Walker, Chair
Arthur Glenberg
Kurt VanLehn

ARIZONA STATE UNIVERSITY

May 2017

ABSTRACT

Reading comprehension is a critical aspect of life in America, but many English language learners struggle with this skill. Enhanced Moved by Reading to Accelerate Comprehension in English (EMBRACE) is a tablet-based interactive learning environment is designed to improve reading comprehension. During use of EMBRACE, all interactions with the system are logged, including correct and incorrect behaviors and help requests. These interactions could potentially be used to predict the child's reading comprehension, providing an online measure of understanding. In addition, time-related features have been used for predicting learning by educational data mining models in mathematics and science, and may be relevant in this context. This project investigated the predictive value of data mining models based on user actions for reading comprehension, with and without timing information. Contradictory results of the investigation were obtained. The KNN and SVM models indicated that elapsed time is an important feature, but the linear regression models indicated that elapsed time is not an important feature. Finally, a new statistical test was performed on the KNN algorithm which indicated that the feature selection process may have caused overfitting, where features were chosen due coincidental alignment with the participants' performance. These results provide important insights which will aid in the development of a reading comprehension predictor that improves the EMBRACE system's ability to better serve ELLs.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

MOTIVATION

English language learners (ELL) may struggle with many aspects of learning the English language, including reading. Reading is a necessary part of a modern life in America (August & Shanahan, 2006), which may affect their performance in school. There is an increasing need to address this issue as the number of students enrolled in ELL programs are increasing (National Center for Education Statistics, 2015).

The Moved by Reading intervention has been shown to improve reading comprehension in native English speakers as well as ELLs (Walker, Adams, Restrepo, Fialko, & Glenberg, 2017; Glenberg, Gutierrez, Levin, Japuntich, & Kaschak, 2004). The Moved by Reading intervention has two phases. In the first phase, the participant read a passage describing a scenario (Glenberg, Willford, Gibson, Goldberg, & Zhu, 2012). Then, the participant acts out the scenario by physically manipulating toys or images. For instance, if the sentence is "The farmer carries the hay to the barn.", the participant recreates a farmer taking a bale of hay to a barn. In the second phase, participants are asked to imagine manipulating toys. Both the physical manipulation and imagine manipulation are designed to have the participants simulate the action (physically or mentally), thereby enhancing the participant's comprehension of the sentence that was read.

The Enhanced Move by Reading to Accelerate Comprehension in English (EMBRACE) application[1] is an interactive tutoring system designed to increase reading

[1] A full description of the EMBRACE application is provided in CHAPTER 3.

comprehension for young Spanish speaking ELLs by using a Moved by Reading intervention. The EMBRACE application has the user read a story and then act out certain sentences by touching and dragging images to simulate the action described in the sentence. After the story is read, the student answers a few multiple-choice questions about the story. The EMBRACE application does not currently have a seamless method of assessing the student's reading comprehension. Currently, the only method of assessing the student's reading comprehension is through an automatically administered and graded test. To improve the adaptability of the application, this paper seeks to answer the following research questions:

1. Can reading comprehension be accurately predicted using action-based log data?
2. Does timing information improve the accuracy of reading comprehension predictions over user actions alone?

The answers to these questions are important to the creation of a student reading comprehension prediction system. With the creation of a student reading comprehension prediction system, the EMBRACE application will be able to predict the student's reading comprehension while the student uses the application. The student will no longer have to stop the learning process in order for the system to gain an estimate of the student's reading comprehension. In addition, the application can use the predicted comprehension to adapt the curriculum to the better fit the student's need.

CHAPTER 2

RELATED WORKS

Many more intelligent tutoring systems (ITS) have been developed for mathematics and science than for language learning. Language learning is a difficult domain in which to develop an ITS (Heilman & Eskenazi, 2006). A given mathematics problem may have only one correct answer, but a sentence may have many correct interpretations. For example: "'I see' said the blind man as he picked up his hammer and saw." One might interpret the word "saw" as a noun, meaning that the man picked up two object. Another interpretation of the word "saw" is that the man was able to see. This ambiguity inherent to language makes it difficult to automatically assess whether an answer is correct or incorrect. However, some work has been done towards automatically assessing reading skills. Also, many ITSs have been developed for mathematics and science which have models predicting student knowledge. The following sections are a survey of the language learning ITSs and the mathematics and science ITS knowledge prediction models. Within the sections are subsections describing individual ITSs.

Intelligent Tutoring Systems

An intelligent tutoring system is an automated computer program which instructs users in a skill and can provide feedback to the user (Anderson, Corbett, Koedinger, & Pelletier, 1995). An ITS can present materials to be learned, provide the user with questions, respond to user questions, prompt the user to stimulate learning, and provide feedback and hints (Ma, Adesope, Nesbit, & Lui, 2014). It maintains a model of the student's knowledge and adjusts the model, as well as the feedback and questions,

according to the user's behavior. ITS can be broken down into outer loops and inner loops (VanLehn, 2006). The outer loop contains tasks or problems for the user to solve. Each of these tasks contain multiple steps to solve. These individual steps make up the inner loop. The inner loop may provide users services such as minimal feedback on a step, error-specific feedback, a hint on the next step, an assessment of knowledge, or a review of the solution. An ITS may give a hint or feedback during either the inner loop or the outer loop. These hints may come immediately after a correct or incorrect step, or there may be a delay in between the step and the hint. As a part of the outer loop, the next task must be selected. This may be done by the user, selecting a predetermined order, determined by difficulty, or adapt to the user's performance.

Project LISTEN

Project LISTEN's Reading Tutor is an ITS designed to improve children's reading abilities (Mostow, 2012). Participants read a text passage and then read the passage aloud. The Reading Tutor uses speech recognition software to "listen" to the participant as they read. The session begins by registering the participant. Then, the Reading Tutor and participant pick a text to read or other activity, such as composing a story. Eventually, the participant ends the session by logging out. During the reading activity, participants read the text aloud. The Reading Tutor may provide hints based on the participant's real time reading performance.

To evaluate the participant's reading performance, the Reading Tutor first checks for words the participant said which were not in the actual text to obtain a miscue detection accuracy (Mostow, 2012). Then, the Reading Tutor compares the transcript

from the speech recognition software to the actual text to obtain a tracking accuracy (Tam, Mostow, Beck, & Banerjee, 2003). The output of the speech recognition software is assumed to be correct. The system then forms a hypothesis based on the probability that the word was read correctly given the features. The features used in determining the probability are the output of the speech recognition software, the alignment of the output text and the actual text, and the participant's previous performance. The hypothesis is then used to create a word alignment description of the passage. The word alignment describes if a participant did not say a word in the passage, mispronounced a word, inserted a word, or correctly said a word. This comparison is used to create the tracking accuracy. Then, the Reading Tutor uses the latency between words to assess the participant's reading ability (Beck, Jia, & Mostow, 2004). In order to calculate the latency of a word, two conditions must be met. First, the word must be correct. Second, the previous word must have been heard, regardless of whether the word was correct or incorrect. The latency is calculated by subtracting the end of the verbalization of the previous word from the beginning of the verbalization of the current word.

In contrast, EMBRACE does not have speech recognition software. Therefore, the log data for EMBRACE does not have the same temporal resolution and fine grain information for participant reading progress. However, both systems record actions performed by the participant (words spoken in the Reading Tutor, screen presses in EMBRACE). A mapping may be created between the actions performed by the participant in EMBRACE's log data and the words spoken by the participant in the Reading Tutor's log data. While the Reading Tutor assesses if the participant spoke the

correct words in the correct order, EMBRACE can assess if the participant moves the correct object to the correct location. Similarly, while the Reading Tutor uses the time between words to help determine reading fluency, EMBRACE could use the time in between actions to help determine reading comprehension.

REAP

REAder-specific Practice (REAP) is an ITS which is designed to help users practice reading comprehension by providing the user with texts and then asking the user questions about the text (Collins-Thompson & Callan, 2004). The REAP system will select a text passage from a database of documents gathered from the Web. Once retrieved, the data base will analyze the passage for linguistic metadata which is used during passage selection and question creation. Selection criteria of the text passage include the user's preferred topic (such as sports), reading level, sentence complexity, and the user's known vocabulary words.

The students are modeled using a Bayesian network with two hidden states (Heilman & Eskenazi, 2008). The hidden states correspond to the student knowing the target word and the student not knowing the target word. These hidden states contain transition probabilities to observed states, such as if the user answers the question correctly. These observations are used to update the probability that a user is in a given hidden state. The observations include multiple choice cloze, synonym, and definition questions as well as a manually graded summary after reading a passage. In addition to the Bayesian network, the REAP maintains a list of vocabulary words that the user knows as well as a list of target vocabulary words.

iSTART

Interactive Strategy Training for Active Reading and Thinking (iSTART) tutors reading comprehension by using self-explanation reading training (SERT) (McNamara, Levinstein, & Boonthum, 2004). SERT uses self-explanation to enhance reading comprehension. SERT strategies include comprehension monitoring, paraphrasing, prediction, elaboration, and making bridging inferences. iSTART maintains the three phases of SERT: introduction, demonstration, and practice. The introduction phase explains SERT strategies to the user. The demonstration phase shows the user examples of the SERT strategies in action by showing the user the interactions between two characters and asking the questions of the user. The questions asked depend on the user's performance. In practice phase, iSTART provides the reader with a text passage and uses a character (Merlin) to guide the user through various SERT exercises. Merlin asks the user to self-explain a sentence from the passage.

The user's response is evaluated in 3 ways (McNamara, Levinstein, & Boonthum, 2004). The initial screening examines the length and relevancy of used words. The overall evaluation provides the user with feedback. If the length and word content are not appropriate to the task, Merlin prompts the user to try again. If the length and word content are satisfactory, Merlin congratulates the user. The last phase of the evaluation asks the user to explain which strategy they used. This requires the user to think about how they use the SERT strategies.

Genetics Cognitive Tutor

The Genetics Cognitive Tutor is designed to help students learn genetics (Corbett, Kauffman, MacLaren, Wagner, & Jones, 2010). The Genetics Cognitive Tutor is broken down into modules. Each module contains multi-step problems, giving feedback after each step. At each step the participant can ask the tutoring system for a hint. The system responds by displaying a hint specific to the problem that the participant is solving. As the participant progresses through the module, the system adapts the content according to the participant's knowledge.

The participant's knowledge is modeled using Model Tracing and Knowledge Tracing (Corbett, Kauffman, MacLaren, Wagner, & Jones, 2010). Model Tracing generates a cognitive model of the student which is updated as the student performs each step. This cognitive model is used for hint generation. Knowledge Tracing is used to estimate the probability that the participant has learned the material and the estimate is updated for the applicable material at the end of each problem-solving step.

ASSISTments

The ASSISTment System is an ITS creation tool designed to expedite the creation of ITSs (Razzaq, et al., 2009). ASSISTment is designed for deployment in classroom settings. As such, it is designed for teachers to create modules, or ASSISTments, with linear progression through a problem. This means that there is only one problem per ASSISTment, but it provides a fine-grained assessment of the student. However, the content of the ASSISTment does not adjust based on the student's performance. ASSISTment is intended to tutor mathematics for students between 4^{th} and 10^{th} grade.

Assistance is provided to the students in the form of scaffolding questions and hints. The system provides hints based on which specific problem the student is currently on and contains different levels of help, ending in the "bottom-out" hint, which gives the answer. Each scaffolding question must be completed to advance to the next scaffolding question. A "buggy" message is provided as feedback if a specific incorrect answer is entered. ASSISTments can be arranged to be performed in a sequence such as random or linear. ASSISTments allow the teacher to map skills to questions. The ASSISTment System generates reports based on student performance. This allows the teacher to see which skills the students performed poorly (or well).

Intelligent Tutoring System Data Mining Models

Within the intelligent tutoring system data mining models section, several models used to predict participant learning are explored. The following subsections are a survey of data mining models created from ITS log data.

Genetics Cognitive Tutor

The log data from the Genetics Cognitive Tutor has been data mined to predict the depth of participant learning and the participant's preparation for future learning. The preparation for future learning (PFL) model attempts to categorize participants based on their ability to apply their new knowledge in new ways or use their new knowledge to more quickly learn another new skill (Baker, Gowda, & Corbett, 2010). A linear regression model was created from a subset of extracted features using a greedy forward selection method with the cutoff criterion being that no new feature added increases the performance of the model. The selected features are the proportion of actions where the

participant makes an error on the first attempt of a poorly known skill without asking for help and proportion of actions where the participant asks for a hint and then waits over 5 seconds. The resulting model was compared to a written test designed to assess the students PFL using the Pearson correlation. The model obtained a correlation of 0.356.

The depth of participant learning models sought to identify students who learned shallow and those who learned deep (Baker, Gowda, Corbett, & Ocumpaugh, 2012). A deep learner will remember the new knowledge in the future and will be able to apply the new knowledge in new ways. A shallow learner will not recall the new knowledge in the future and cannot apply the new knowledge in new ways. The predicting models used to detect the depth of participant learning were linear regression models, using a threshold (0.5) to categorize the output as either shallow (below 0.5) or deep (at or above 0.5). Two models were created from a subset of extracted features. The features were selected in a greedy forward selection method with the cutoff criterion being that no new feature increased the performance of the model. The first model, multiplicative-interactions, multiplied two of the features together (hence multiplicative). It used the following features: the response time after an error message and the average probability that an action was a slip (accidental error) multiplied by the average response time. The second model, no-interactions, used the following features: the response time after an error message and the average unitized response time in standard deviations. Both models were compared to a written test to assess the depth of the participants' learning. This test was used as a baseline to assess the performance of the models using Cohen's Kappa and by measuring the area under the receiver operating characteristic curve (A'). The first model

obtained a Kappa of 0.389 and had an A' of 0.758. The second model obtained a Kappa of 0.346 and had an A' of 0.767.

Similarly to EMBRACE, the Genetics Cognitive Tutor produces a log of system events, actions, and states. While EMBRACE does not adapt based on the user's behavior like the Genetics Cognitive Tutor, it does have user input which is either correct or incorrect. A mapping could be made for help requests and errors across the two systems. Given the success of these models, the following insights can be obtained: the log data can be mined for information, help requests will be an important feature, and errors will be an important feature.

ASSISTments

The ASSISTment System has been used to create several ITSs including used to collect student data from 2010-2011 from 15,931 students. This data set was mined to improve knowledge tracing and to model knowledge retention. The improved knowledge tracing model uses an existing Knowledge Tracing algorithm and the student's first response time as features for a linear regression model (Wang & Heffernan, 2012). The Knowledge Tracing algorithm used was implemented in MatLab. The student's first response time was binned into four classes relative to the performance of other students for the same problem. The improvement of the new model when compared to the original Knowledge Tracing model was small. When the Root Mean Squared Error for both the original Knowledge Tracing and linear regression model was compared using a two tailed paired t-test, with $p = 0.0389$.

The model predicting participant knowledge retention sought to predict if a student would retain information learned using ASSISTment (Wang & Beck, 2012). The participant was asked to use the skill again 5-10 days later. A logistic regression model was built using the correctness of the response 5-10 days later as the dependent variable. User identity and skill identity were used as factors. The features extracted for each skill were the number of correct responses, number of days the student took to learn the skill, the exponential moving mean of the student's performance, the exponential moving mean of the student's response time, the slope of the student's 3 most recent performances, the number of days since the student last saw the skill, and the difficulty of the problem. The resulting model obtained and R^2 of 0.25. The number of days since the material was seen and the exponential mean performance had the strongest B values while the number of correct responses had the lowest B value.

While ASSISTments is designed for mathematics, some of the lessons learned apply across domains to EMBRACE. The first response time appears to augment an existing prediction model. The exponential mean performance is important for predicting the long-term retention of materials.

CHAPTER 3

EMBRACE

EMBRACE is an iPad application designed to teach English reading comprehension skills to the user (Walker, Adams, Restrepo, Fialko, & Glenberg, 2017). EMBRACE allows the user to choose a story. Each story is divided into chapters. Each chapter is divided into sentences. The stories used are written by the project members specifically for use in the EMBRACE application. Each chapter contains image and audio files used when presenting the stories to the user. Two stories are used in the application: The Best Farm and The Circulatory System. These stories have 7 and 5 chapters respectively. The Best Farm is a narrative story about a farmer preparing for a contest. The Circulatory System is an expository story about the human circulatory system. The Best Farm contains an introductory story to familiarize the user with the system, including how to manipulate the objects on screen. After the initial chapter, participants are then instructed to read the stories in a specific order. Each story requires the participant to complete each of the chapters in order. After completing all chapters within a story, the participant may then move on to the next story.

At the beginning of each chapter, a list of vocabulary words is provided. Participants may tap on the vocabulary words to hear the pronunciation and read the definition. When applicable, the corresponding image is also highlighted. Each chapter consists of multiple sentences and image sets. The image sets depict what is being described by the sentences. The sentences are displayed in the upper right corner of the

application while the images are displayed in the background of the application. All sentences are displayed for the duration of the chapter. Each sentence will either be blue

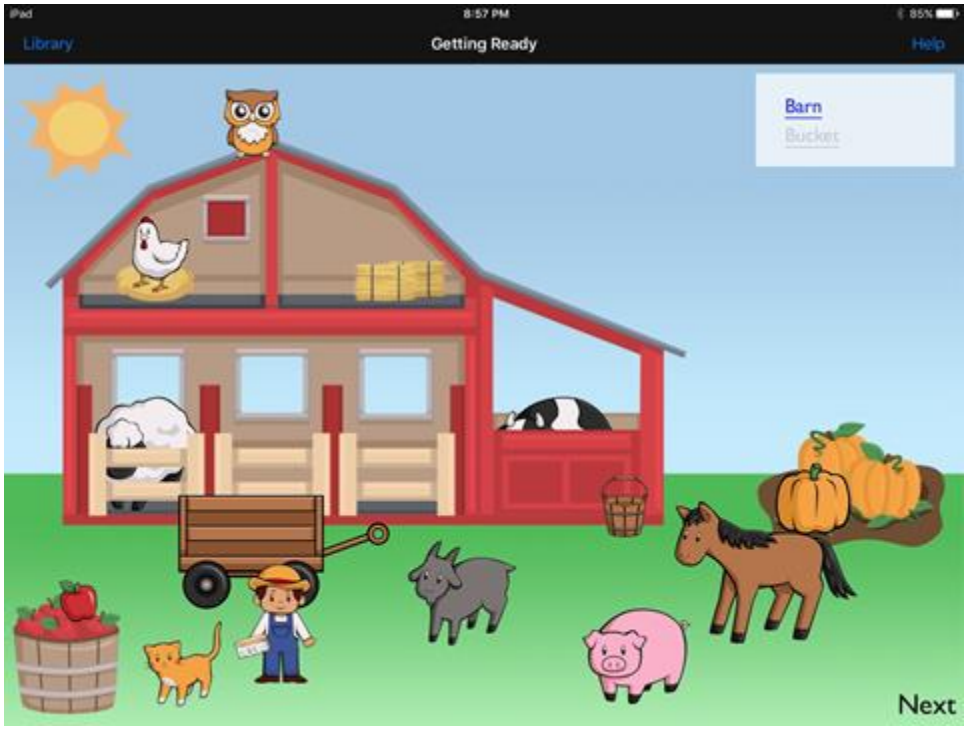

Figure 1. The EMBRACE Initial Vocabulary List

or black. Blue sentences will require the participant to manipulate one or more of the images on screen to act out the action described in the sentence and are called action sentences. Black sentences will not require any manipulation from the participant and are called non-action sentences. The current sentence will have a high opacity in the image whereas the other sentences will have a lower opacity. This difference makes visually locating the current sentence easier.

Both types of sentences contain underlined words. These underlined words, when tapped, will play an audio file of the pronunciation of the word. The underlined words will also highlight the image corresponding to the word if it is applicable (e.g. if "carried"

is underlined, no image is highlighted). The participant advances to the next sentence by tapping the “Next” button in the bottom right corner of the screen.

Figure 2. EMBRACE: Example of Sentences Used in a Story

Because EMBRACE uses a Moved by Reading intervention, before action sentences can advance, the participant must correctly perform all image manipulations. This is done to ensure that the participant has done the physical manipulation correctly. Image manipulations are performed by tapping on an image and dragging that image to another image in order to simulate the action described in the sentence. These simulations may be simple, requiring only one manipulation, or complex, requiring multiple manipulations. For instance, “The farmer brought the hay to the horse” requires the participant to tap on the farmer, drag the farmer to the hay. The application then groups

the two images together. The participant then moves the grouped images to the horse. In addition, the application may prompt the participant to indicate the nature of the relationship between two images. For instance, the participant may have to indicate if the farmer stands on a pig or if the farmer leads a pig. If the participant makes an error, an error sound plays and the image returns to its original position.

The EMBRACE application records the system's state in a log data file. There are 29 data entries recorded per row including a time stamp, the current sentence, and audio file loaded. The EMBRACE log data is similar to ITS log data.

After completing the story, participants were administered a post-test assessment of comprehension. The assessment covered the content of the story. The assessment began by providing a general prompt asking the participant to recall what was covered in the story. This prompt was then followed up with 5-7 questions for each chapter of the story asking for specific details of the story, such as "What is the name of the farmer?" and "How does eating an apple help the horse's teeth?". These responses were recorded and used to assess the participant's comprehension.

The EMBRACE application was used for 3 experimental groups and a control group. The participants were Spanish dual language learners. The 3 experimental groups included: Spanish support only, simulation only, and Spanish support and simulation. The Spanish support was provided in three ways: vocabulary page (at the beginning of the chapter) is provided in Spanish and English, the vocabulary help (while reading the sentences) is provided in Spanish and English, and the first chapter is read by the iPad to the child in Spanish. The Spanish support only group received Spanish support and was

told that the blue sentences were important, but not to manipulate anything. The simulation group was told to manipulate the images to act out the scenario described by the blue sentences. The Spanish support and simulation group had both Spanish support and was told to manipulate the images to act out the scenario described by the blue sentences. The control group received no Spanish support and was told that the blue sentences were important, but not to manipulate anything.

A total of 93 Latino dual language learners were randomly assigned to one of the four groups. Over the course of 5 days, participants used the EMBRACE application and were assessed for reading comprehension using the above procedure. Analysis of the results indicated an effect for simulation on narrative texts. Additionally, there was an effect for simulation and decoding skill. A higher decoding skill increased the effect of simulation. No effect was found for the Spanish support on either narrative or expository texts. This indicates that simulation improved reading comprehension but Spanish support did not.

CHAPTER 4

CORPUS

To address the first research question: (Can reading comprehension be accurately predicted using action-based log data?), the data set analyzed was the log data for participants using the EMBRACE application. The data was collected over 5 days from 96 students in grades 2-5. This data set was previously covered in the EMBRACE description section of this paper. The data was divided by chapters and by participants. For example, participant 1's data was divided into 12 samples, one for each chapter.

Of the 96 students, 48 students were a part of the simulation group. In addition, there was a Spanish support and English only manipulation. However, this division was not considered as the experimenters found no statistically significant difference in the performance of the Spanish support and English only groups. The simulation group was selected to be analyzed because the experimental group performed image manipulations using the application and thus the log data had indicators whether the participant understood the sentence. Explicitly, this is the correct or incorrect action recorded in the log data. The data for these participants were chosen to be analyzed because other models described in the related works section use correct/incorrect successfully.

Each participant read two stories: The Best Farm and The Circulatory System. The Best Farm consists of 7 chapters and proved relatively easy for the participants to comprehend. The Circulatory System consists of 5 chapters and proved relatively difficult for the participants to comprehend. During the experiment, some participants were unable to complete the experiment in the time frame provided to them. This

restriction led to some of the participants being unable to complete some or all the chapters in the application. Similarly, some participants did not complete the post-test assessment. The chapters with either incomplete log data or the corresponding incomplete post-test assessment were removed from consideration. To be explicit, this means that of the chapter log data recorded, only the chapters which met the 3 following requirements were considered: complete log data, completed post-test assessment, and the corresponding participant was in the experimental group. Of the total 576 individual chapter log data (48 participants * 12 chapters), 350 individual chapter log data were considered, see Table 1.

Story	Chapter	Number of Samples
The Best Farm	1	20
The Best Farm	2	36
The Best Farm	3	22
The Best Farm	4	25
The Best Farm	5	33
The Best Farm	6	24
The Best Farm	7	35
The Circulatory System	1	30
The Circulatory System	2	33
The Circulatory System	3	27
The Circulatory System	4	31
The Circulatory System	5	34

Table 1. Number of Samples Per Chapter

For the post-test assessments, the participant was given two chances to answer the question correctly. If the participant answered correctly on the first attempt, the answer was considered correct. Otherwise, the answer was considered incorrect. The participants were given two attempts as a part of the experimental design. However, only the first attempt was considered in the current evaluation. Some of the data mining algorithms

used in this project require a discrete classification. Other data mining algorithms allow for a continuous classification. As such, two classifications of participant reading comprehension were created, one using the percent correct (continuous) and one using the number or incorrect responses (discretized). The number of incorrect responses was chosen as the discretized classification because the number of questions differed between chapters. Zero incorrect answers indicate that the participant gave all correct responses, regardless of how many questions were asked in a chapter.

The number of incorrect responses differed greatly between stories. As such, two different classification methods were used for classifying the participant's knowledge. The boundaries for the discretized classification were decided by minimizing the size of the largest class in all chapters for a story. This was done to avoid a major class imbalance for either story.

Chapter	Class 1	Class 2	Class 3
1	5	8	7
2	12	9	15
3	10	6	6
4	10	8	7
5	9	15	9
6	16	4	4
7	21	11	3
Total	83	61	51

Table 2. Class Distribution by Chapter for the Best Farm

For the first story, The Best Farm, if the participant answered all questions correctly (no incorrect answers), then that participant was classified as belonging to class 1. If the participant answered one question incorrectly, then that participant was classified as belonging to class 2. If the participant answered two or more questions incorrectly,

they were classified as belonging to class 3. See Table 2 for the breakdown of number of participants in each class.

For the second story, The Circulatory System, if the participant answer zero to two questions incorrectly, then that participant was classified as belonging to class 1. If the participant answered three or four question incorrectly, then that participant was classified as belonging to class 2. If the participant answered five or more questions incorrectly, they were classified as belonging to class 3. See Table 3 for the breakdown of number of participants in each class.

Chapter	Class 1	Class 2	Class 3
1	14	13	3
2	13	19	1
3	3	15	9
4	2	12	17
5	7	19	8
Total	39	78	38

Table 3. Class Distribution by Chapter for the Circulatory System

CHAPTER 5

FEATURES

The log data from the Corpus consists of 29 attributes describing the system state. These attributes include four attributes of interest: an attribute indicating if the participant performed a correct or incorrect action, an attribute indicating if the participant started a new sentence, an attribute indicating if the student requested a hint, and an attribute containing a time stamp. To transform the 29 attributes into the features of interest, these state descriptors were broken down into 2 categories: elapsed time and action type. The action type category consists of new sentence (NS), hint (H), correct response (C), and incorrect response (I). The new sentence action type was chosen because it represents the start of a new outer loop. The hint action type was chosen because it is used as a component in features for Baker, Gowda, and Corbett (2010). The correct response and incorrect response action types were chosen because they are used as components in features for Baker, Gowda, Corbett, and Ocumpaugh (2012).

The elapsed time category is the time elapsed between each action recorded. The elapsed time data were grouped according to chapter and participant. The discretization process is shown in Figure 3. For example, the elapsed times for participant 1's first chapter were grouped together and divided into quartiles. The elapsed times for participant 2's first chapter were not considered when discretizing participant 1's elapsed time nor were participant 1's second chapter considered. The elapsed time was discretized into quartiles: 0%-25% or short (S), 25%-50% or slightly short (SS), 50%-

75% or slightly long (SL), and 75-100% or long (L). Quartiles were chosen to account for the fact that participants may read at different rates. Using standard deviations to

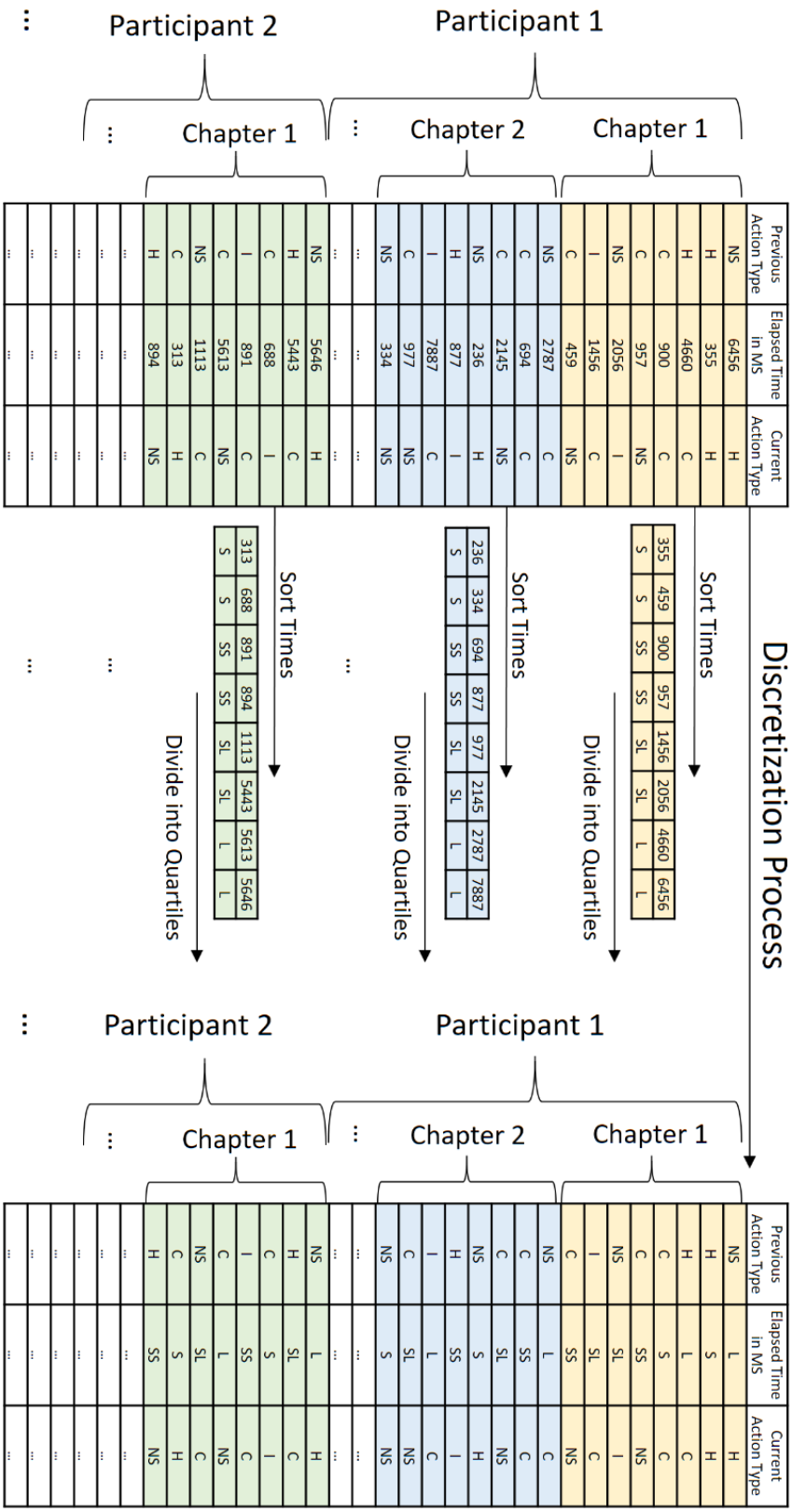

Figure 3. Time Discretization Process

discretize the elapsed time was considered as a possible alternative, however the elapsed times did not form a Gaussian distribution.

The 2 categories, elapsed time and action type, were used to create 3 intermediate attributes: type of action before current action, type of current action, and time in between the two actions. The 3 intermediate attributes were then used to extract the features used for the model creation. The extracted features are a count of the number of times that the following sequence of previous action type, elapsed time, current action type occurs. For the sake of simplicity, the extracted features shall be referred to using the notation x-y-z, where x is the previous action type, y is the elapsed time, and z is the current action type. For example, NS-L-H represents the sequence previous action NS (new sentence), elapsed time L (long), and the current action H (hint). In this case, NS is x, L is y, and H is z. For this notation, a new action type is introduced. The symbol α represents any action type. Similarly, the symbol β represents any elapsed time.

An example of how the features were extracted is provided in Table 4. One of the features extracted would be NS-SS-H. The value for this extracted feature would be 2. This is because NS-SS-H appears as a sequence twice.

Sample	Previous Action Type	Elapsed Time	Current Action Type
1	NS	SS	H
2	H	L	NS
3	NS	SS	H
4	H	L	C
5	C	L	NS

Table 4. Example of Intermediate Attributes for a Participant

All possible combinations of previous action type, elapsed time, and current action type were considered except for I-β-NS, incorrect response, any elapsed time, new

sentence, see Table 5. This sequence I-β-NS is excluded because the EMBRACE does not allow a participant to advance to the next sentence after an incorrect response. Thus, this sequence never occurs and will not have any predictive value. These combinations total to 60 extracted features.

Previous Action Type	Elapsed Time	Current Action Type
NS	S	NS
NS	S	H
NS	S	C
NS	S	I
NS	SS	NS
...	…	…
NS	L	I
H	S	NS
…	…	…
C	L	I
I	S	H
…	…	…
I	L	I

Table 5. Enumeration of Extracted With Time Features

In addition to the combinations described above, two more groupings of possible combinations were included. The first additional grouping is the combination of previous action type and elapsed time. This group is generated by counting the number of times that a specific previous action type is followed by a specific elapsed time. Referring back to Table 4, the feature H-L-α would have a count of 2 because it occurs twice. The second additional grouping is the combination of elapsed time and current action type. Similarly to the previous additional group, this set is generated by counting the number of times a specific elapsed time and a specific current action occurs. Referring back to Table 4, the feature α-L-NS would have a count of 2 because it occurs twice in the grouping.

NS-S-NS	NS-SS-NS	NS-SL-NS	NS-L-NS	H-S-NS	H-SS-NS	H-SL-NS	H-L-NS	C-S-NS	C-SS-NS	C-SL-NS	…
0	2	3	0	0	0	0	2	4	1	0	…
0	2	2	1	0	0	1	1	5	0	0	…
0	0	3	3	0	0	0	1	4	1	0	…
0	0	2	4	0	0	0	1	3	1	0	…
0	1	2	3	0	0	0	1	5	0	0	…
0	0	1	2	0	1	2	1	3	0	2	…
0	2	2	2	0	0	0	1	4	1	0	…
0	2	1	3	0	0	1	0	3	2	0	…
0	2	1	3	0	0	0	1	4	1	0	…
0	0	3	2	0	0	0	2	3	2	0	…
0	0	2	2	1	0	0	2	3	1	1	…
0	2	3	1	0	0	0	1	5	0	0	…
0	2	1	2	0	0	0	2	5	0	0	…
1	0	2	3	0	0	0	1	4	1	0	…
0	2	1	3	0	0	1	0	4	0	0	…
0	2	1	3	0	0	0	1	4	1	0	…
…	…	…	…	…	…	…	…	…	…	…	…

Table 6 Sample of Extracted Features

For an examples a subset of the extracted features, see Table 6. These two additional groups were included to investigate whether the elapsed time before or after an event would have an increased predictive value. Each of the additional groups adds 16 more features to the 60 above for a total of 92 features.

In order to answer the second research question (Does timing information improve the accuracy of reading comprehension predictions over user actions alone?), another set of features was extracted. The second set is the same as the first set, except the elapsed time is not considered. Explicitly, this difference means that the combinations of previous action type and current action type are considered for a total of 15 features. Table 7 shows the combinations of attributes for the first grouping of 15 features. The two additional groupings of other features are a count of the number of times a specific action type occurs as a previous action type or as a current action type. There are two reasons for including these two groups of additional features. First, by using the count of the action type, we can examine the predictive value of an action type (Baker, Gowda, & Corbett, 2010). For example, we can examine the predictive strength of the number of hint requests and compare it to the number of hint requests paired with a specific elapsed time. Second, the last action type in a sample will never be listed as a previous action because the chapter ends after the last action. By distinguishing between the previous action type and current action type, we can represent the last action before the chapter ends. The two additional groups each add 4 features to the 15 features for a total of 23 features.

Previous Action Type	Current Action Type
NS	NS
NS	H
NS	C
NS	I
H	NS
…	…
H	I
C	H
…	…
I	I

Table 7. Enumeration of Extracted Without Time Features

The first set of features will be referred to as “features with time”. The second set of features will be referred to as “features without time”. The second set of features is used to compare against the first set. It will help isolate the predictive value of the elapsed time because the two sets are otherwise identical.

CHAPTER 6

METHODOLOGY

Assessment Strategy

Several data mining algorithms were considered in assessing the predictive value of the features. The predictive ability of the resulting model, using the features with time and features without time was assessed in phases. During all phases, the models were assessed using leave-one-out cross validation (loocv). Loocv is a method of cross validating (a method of preventing overfitting of a model) which requires that a sample be removed from the pool of all samples. The remaining samples, called the training samples, are then used to build a model. The model then predicts what class the removed sample, called the test sample, should belong to. The prediction is then recorded and compared to the actual class that the test sample belongs to. The test sample is then added back to the set of training samples and the process is repeated until each sample has been used as a test sample. After each sample has been tested, the results of the predictions are used to assess the performance of the model.

Loocv was used (as opposed to k-fold) because it allows the models to be validated while leaving as many samples in the training set as possible. Given the relatively small number of samples for each chapter, maintaining as large of a training set as possible is important.

In the first phase of model assessment, two models using the same data mining algorithm were created from all features with time and all features without time. For example, one random forest model was created using the 92 features with time and a

second random forest model was created using the 23 features without time. If the algorithm used to generate the models uses randomness, the stability of the models' predictions was examined. The algorithm's stability was examined because if an algorithm uses randomness, it can produce wildly different models from the same data. Thus, if the randomness influences the model too much, it can drown out the predictive value of any model. If the models' accuracy did not vary by more than 10% between iterations of model generation, then the models' stability criterion was considered to be met. If this criterion was met, then the corresponding algorithm was considered for the next phase. Note that at this time, the accuracy of the model using the features with time and the accuracy of the model using the features without time are not being compared against each other.

In the second phase, the algorithms were used to create a model using each feature individually. For example, one model was created using only the first feature, another model was created using only the second feature. The accuracies of the single feature models were then compared against each other. A subset of features was then selected using a wrapper greedy forward selection (Curuana & Freitag, 1994). A subset of features was created starting with the null set. Each feature was added to the set, used to create a model and then removed. The top scoring feature was then added to the subset and used in subsequent iterations. This process was repeated until N features were selected, where $3 \leq N \leq 9$. N has to be greater than or equal to 3 in order to prevent outliers and statistical anomalies from dominating a model. N has to be less than 10 because 10 would represent over half of the distinct features for the without time feature

set[2]. The actual N for each algorithm was chosen using a greedy strategy. The algorithm assessed would be used to select the top scoring 3-9 features. The algorithm would then be used on those N features to create a model for each chapter. The accuracy of the models for each chapter using the N features were averaged. The N which produced the highest average accuracy was chosen.

In the third phase, the accuracy of the model created using the subset of the features with time and the accuracy of the model created using the subset of the features without time were compared using a pair wise Student's t-test. This comparison was done to assess the predictive strength of the elapsed time. For the standard linear regression model, the R^2 for each chapter and feature set was be calculated. The R^2 is defined as $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$. SS_{res} is defined as $SS_{res} = \sum_i (y_i - f_i)^2$ and SS_{tot} is defined as $SS_{tot} = \sum_i (y_i - mean(y))^2$.

The same procedure described above was repeated for one algorithm which combined the data across chapters. For this algorithm, all of the samples for all the chapters for one story are combined for analysis. The features with time samples from chapters 1-7 of The Best Farm were combined and used to form one set of samples. Similarly, the features without time samples from chapters 1-7 of The Best Farm were combined and used to form another set of samples. This process was repeated for The Circulatory System. These sets of samples will be referred to as the samples by story. For phase 3, no Student's t-test was performed as there were only 2 pairs to analyze.

[2] Even though the without time feature set has 23 features, the NS-α, α-NS, C-α, α-C features should have no variation within themselves, bringing the total number of distinct features to 19.

In addition to the Student's t-test, a non-standard approach was taken to analyze the predictive power of elapsed time. Due to the large computation cost of training some of the models, this procedure was only performed for one of the algorithms. For this procedure, the elapsed time for each of the intermediate samples was randomly assigned. This effectively randomizes the time aspect of all the features. This procedure is repeated until there were 100 sets of randomized times. These randomized time sets were then used to create models in the same way that the actual elapsed time set was. The accuracy of the randomized time set models was assessed and then compared to the accuracy of actual elapsed time set. If the elapsed time has predictive value, then the actual elapsed time set should outperform the majority of the 100 randomized time sets. For this paper, we shall call this non-standard test the empirical sampling distribution test (ESDT).

Algorithms Assessed

Five algorithms were chosen for assessment: artificial neural network (ANN), boosted random forest (RF), support vector machine (SVM), k-nearest neighbors (KNN), and linear regression (LR). All the algorithms were implemented using MATLAB.

The ANN algorithm creates a network of nodes (neurons in the analogy to a neural network) with randomly initialized weights in between each node. The model is then fed samples and the output of the network is then compared against the actual class. The weights in between each node are then adjusted for the next sample. This process stops once some criterion is met, for example, number of iterations, or accuracy of the output. The strength of the algorithm is that it can identify important features without any

indication from the user. While training the model is computationally very expensive, using the model to predict the class of a sample is computationally inexpensive. This will be important because the model is intended to be used quickly on a tablet processor with relatively little processing power (compared to a desktop computer). It is important to note however, that because the weights are randomly initialized, the resulting model may differ between iterations of the algorithm. Therefore, during the first Phase, the stability of the prediction across iterations will be considered.

The boosted RF algorithm is a binary classification algorithm. It creates a series of decision stumps. Decision stumps are simple yes/no classifications using one feature to predict if the sample is in a class or not in a class. Decision stumps are created by taking a subset of the training samples and finding an optimal threshold to split the samples as either belonging to the class or not belonging to the class. The subset of samples is randomly chosen. In a boosted RF, each sample has an associated weight. After a decision stump is created, each training sample is classified using the decision stump. If a sample is incorrectly classified in the decision stump, it's weight is increased, thus increasing the odds that the sample will be chosen in the next decision stump. A random forest is created once a specified number of decision stumps has been created for each feature. If there are more than two classes, a RF model is created for each class. The test sample is run on each RF model and the results are used to determine which class the sample belongs to. The strength of this model is that it does not rely on any one sample or any one feature to classify a sample. Instead, it relies on multiple samples and multiple features to classify each test sample. RF models are computationally cheap to classify a

sample, meaning that they are well suited to run on a tablet processor. It is worth noting that because samples are randomly chosen, the resulting model may be different between iterations. Therefore, during the first Phase, the stability of the prediction across iteration will be considered.

The SVM algorithm is a binary classification model. For simplicity, it can be thought of as creating a linear boundary separating samples in the class and samples not in the class[3]. The boundary created is optimized to have the widest possible margin between the "in class" and "out of class" samples. The strength of this model is that it is relatively computationally cheap to classify a test sample, meaning that it is well suited to run on a tablet processor.

The KNN algorithm is a simple algorithm. It requires a training set of samples. The algorithm then finds the k nearest neighbors and uses those samples to predict which class the test sample belongs to, usually through majority voting. Unlike the previous algorithms, this algorithm is more computationally expensive. However, this algorithm requires no training time, meaning that a system would only need to change which training sets are used to make a prediction for a new problem, e.g. a new chapter. Therefore, it will be easy to adapt this algorithm to any new chapters added or any changed chapters.

The LR algorithm takes a set of features as input and calculates a continuous value for the output. It follows the format $y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n$ where each feature input is multiplied by some constant. The products are summed together

[3] It is possible to create non-linear separation using kernel functions.

(including b_0) to arrive at the predicted value, y. The strength of this algorithm is that it is computationally cheap, meaning that it is well suited to run on a tablet processor. Two types of LR algorithms were implemented for examination. First, a standard LR (SLR) algorithm was implemented and evaluated against the percent of correct responses given by the participant using an R^2. Second, a classification LR (CLR) algorithm was implemented. The CLR took the predictions made by the LR model and then rounded them to the nearest possible percent for that chapter. It is important to note that the different chapters had a different number of questions. The resulting rounded number was then compared to the actual percent and evaluated as either a hit or a miss. The number of hits was then used to obtain the accuracy of the model. These two models were chosen as the LR algorithm is typically used in educational data mining for analysis. The CLR is created as a method to compare the SLR, which gives a continuous prediction, against the other data mining models, which give discrete predictions.

CHAPTER 7

RESULTS

The first algorithm attempted was the ANN using 20 hidden nodes. The training function was the Levenberg-Marquardt function. The model produced wildly varying results (over 20% accuracy difference between iterations for chapter 2 of The Best Farm) while using leave-one-out cross validation, see Tables 8 and 9. The ANN algorithm did not pass phase 1 because of the stochastic nature of the accuracy of the models produced.

Iteration 1			
Story	Chapter	With Time	Without Time
The Best Farm	1	45.0%	30.0%
The Best Farm	2	19.4%	8.3%
The Best Farm	3	40.9%	22.7%
The Best Farm	4	48.0%	20.0%
The Best Farm	5	45.5%	51.5%
The Best Farm	6	54.2%	45.8%
The Best Farm	7	37.1%	37.1%
The Circulatory System	1	56.7%	36.7%
The Circulatory System	2	54.5%	60.6%
The Circulatory System	3	48.1%	37.0%
The Circulatory System	4	41.9%	41.9%
The Circulatory System	5	41.2%	38.2%

Table 8. Percent Accuracy of the Resulting Models for the First Iteration of the ANN Algorithm by Chapter

Iteration 2			
Story	Chapter	With Time	Without Time
The Best Farm	1	35.0%	10.0%
The Best Farm	2	41.7%	36.1%
The Best Farm	3	45.5%	27.3%
The Best Farm	4	56.0%	28.0%
The Best Farm	5	45.5%	33.3%
The Best Farm	6	37.5%	45.8%
The Best Farm	7	45.7%	45.7%
The Circulatory System	1	43.3%	16.7%
The Circulatory System	2	57.6%	57.6%
The Circulatory System	3	44.4%	33.3%
The Circulatory System	4	38.7%	38.7%
The Circulatory System	5	44.1%	32.4%

Table 9. Percent Accuracy of the Resulting Models for the Second Iteration of the ANN Algorithm by Chapter

The next algorithm used was the RF. This model also produced wildly varying results (over 20% accuracy difference between iterations for chapter 4 of The Circulatory System) while using leave-one-out cross validation, see Tables 10 and 11. The RF algorithm also did not pass phase 1 because of the stochastic nature of the accuracy of the models produced.

Iteration 1			
Story	Chapter	With Time	Without Time
The Best Farm	1	27.1%	20.8%
The Best Farm	2	36.2%	25.5%
The Best Farm	3	38.1%	14.3%
The Best Farm	4	33.3%	22.2%
The Best Farm	5	29.5%	25.0%
The Best Farm	6	31.8%	38.6%
The Best Farm	7	40.0%	28.9%
The Circulatory System	1	43.2%	15.9%
The Circulatory System	2	26.2%	52.4%
The Circulatory System	3	13.2%	13.2%
The Circulatory System	4	10.0%	27.5%
The Circulatory System	5	33.3%	9.5%

Table 10. Percent Accuracy of the Resulting Models for the First Iteration of the RF Algorithm by Chapter

Iteration 2			
Story	Chapter	With Time	Without Time
The Best Farm	1	31.3%	41.7%
The Best Farm	2	36.2%	40.4%
The Best Farm	3	45.2%	45.2%
The Best Farm	4	31.1%	28.9%
The Best Farm	5	27.3%	31.8%
The Best Farm	6	36.4%	40.9%
The Best Farm	7	31.1%	26.7%
The Circulatory System	1	50.0%	15.9%
The Circulatory System	2	26.2%	26.2%
The Circulatory System	3	23.7%	18.4%
The Circulatory System	4	32.5%	12.5%
The Circulatory System	5	14.3%	19.0%

Table 11. Percent Accuracy of the Resulting Models for the Second Iteration of the RF Algorithm by Chapter

The SVM algorithm does not use randomness and thus was not considered for phase 1. In phase 2, the N chosen was 4, with an average accuracy of 59.9%[4]. The selected features are shown in Tables 13 and 14. In phase 3, the Student's t-test revealed a statistically significant difference, $p < 0.001$. The resulting accuracies are shown in Table 12 and Figures 4 and 5.

Story	Chapter	With Time	Without Time
The Best Farm	1	80.0%	60.0%
The Best Farm	2	69.4%	47.2%
The Best Farm	3	68.2%	9.1%
The Best Farm	4	68.0%	40.0%
The Best Farm	5	72.7%	45.5%
The Best Farm	6	83.3%	12.5%
The Best Farm	7	82.9%	31.4%
The Circulatory System	1	70.0%	43.3%
The Circulatory System	2	81.8%	54.5%
The Circulatory System	3	77.8%	51.9%
The Circulatory System	4	77.4%	71.0%
The Circulatory System	5	73.5%	67.6%

Table 12. Percent Accuracy of the Resulting SVM Models by Chapter

[4] The process for choosing the N is described in the Methodology chapter.

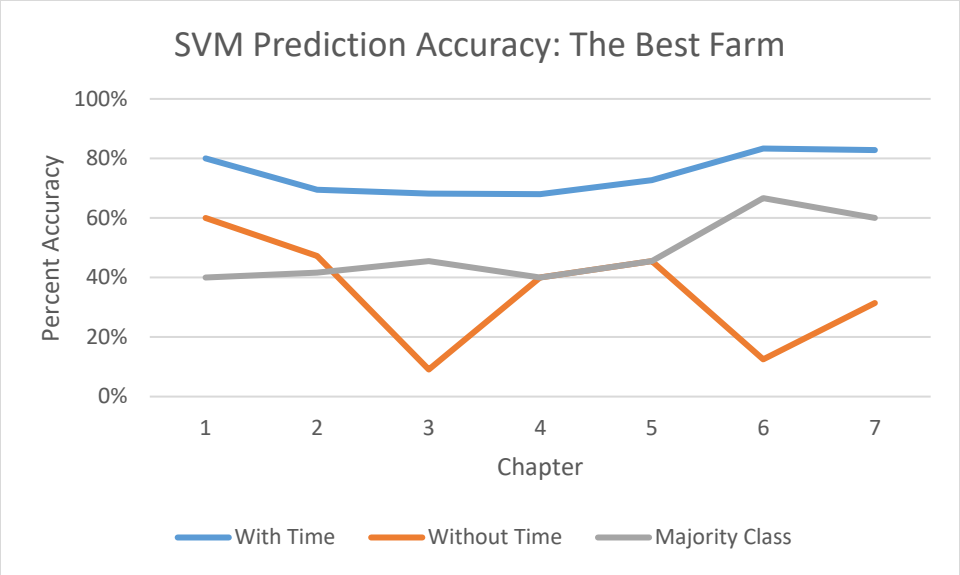

Figure 4. Percent Accuracy of the Resulting Models for the SVM Algorithm for The Best Farm

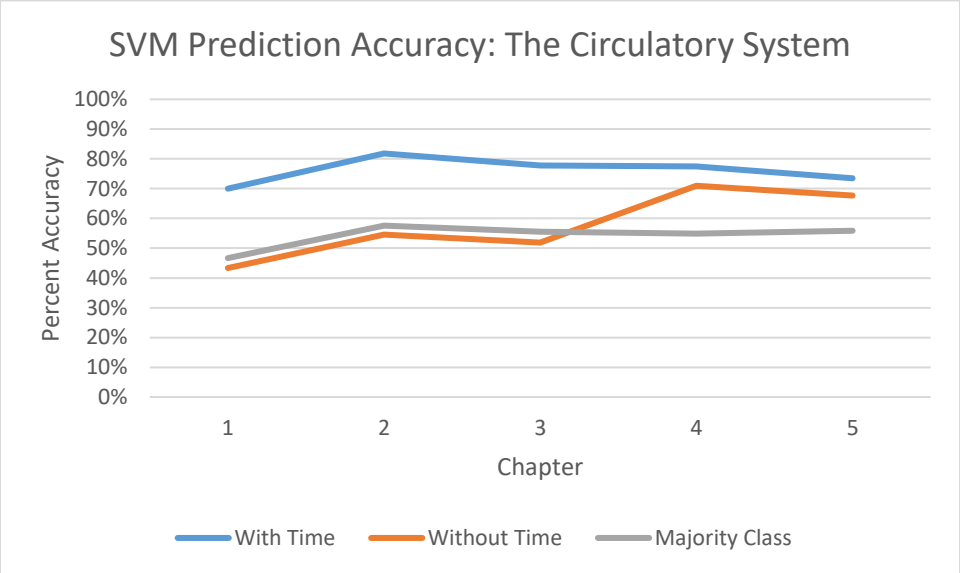

Figure 5. Percent Accuracy of the Resulting Models for the SVM Algorithm for The Circulatory System

Story	Chapter	Rank 1	Rank 2	Rank 3	Rank 4
The Best Farm	1	I-SS-I	X-SS-H	NS-SL-X	H-S-NS
The Best Farm	2	NS-SS-NS	C-SS-H	X-S-H	I-S-H
The Best Farm	3	NS-SS-I	H-S-H	NS-S-C	NS-L-C
The Best Farm	4	NS-SS-X	H-S-C	H-SS-X	NS-SS-NS
The Best Farm	5	H-SL-X	X-SS-C	C-SS-NS	H-SS-C
The Best Farm	6	H-SS-C	C-SS-NS	H-SL-NS	I-SL-C
The Best Farm	7	H-SS-C	C-SL-NS	X-S-NS	X-SS-NS
The Circulatory System	1	H-SS-C	NS-S-X	NS-S-NS	NS-SS-C
The Circulatory System	2	I-S-I	C-SL-H	X-SS-C	C-S-C
The Circulatory System	3	X-SL-C	C-SL-H	C-S-H	NS-S-C
The Circulatory System	4	H-SS-C	NS-S-NS	H-SS-NS	NS-L-I
The Circulatory System	5	C-SS-NS	C-L-C	H-S-H	C-L-NS

Table 13. Top 4 Extracted With Time Features for the SVM Algorithm by Chapter

Story	Chapter	Rank 1	Rank 2	Rank 3	Rank 4
The Best Farm	1	C-H	C-C	NS-NS	I-I
The Best Farm	2	H-NS	H-C	H-H	X-H
The Best Farm	3	H-H	C-H	H-C	NS-NS
The Best Farm	4	NS-NS	H-NS	H-H	C-H
The Best Farm	5	NS-NS	H-NS	C-NS	I-H
The Best Farm	6	H-C	I-C	C-I	NS-I
The Best Farm	7	NS-NS	H-NS	C-H	C-C
The Circulatory System	1	NS-I	H-I	C-NS	I-H
The Circulatory System	2	H-C	C-NS	I-H	H-C
The Circulatory System	3	NS-C	NS-NS	H-NS	H-C
The Circulatory System	4	NS-I	H-NS	C-C	H-H
The Circulatory System	5	H-I	H-H	NS-C	I-I

Table 14. Top 4 Extracted Without Time Features for the SVM Algorithm by Chapter

The KNN algorithm does not use randomness and thus was not considered for phase 1. For phase 2, the N chosen for this algorithm using samples by chapter was 4, with an average accuracy score of 65%. The 4 features chosen are shown in Tables 18 and 19. In phase 3, the Student's t-test revealed a statistically significant difference with $p < 0.001$. The resulting accuracies are shown in Table 15 and Figures 6 and 7. The KNN

algorithm was used to analyze the samples by story. The N chosen was 6. The accuracy of the features with time and without time for The Best Farm and The Circulatory System are shown in Table 16. The KNN algorithm was selected for the ESDT. The KNN model using the actual elapsed time outperformed an average of 48% of the randomized time models. The maximum performance came from chapter 6 of The Best Farm where it outperformed 98% of the randomized time models. The minimum performance came from chapter 3 of The Circulatory System where it outperformed 2% of the randomized time models. These ESDT results are shown in Table 17.

Story	Chapter	With Time	Without Time
The Best Farm	1	80.0%	60.0%
The Best Farm	2	66.7%	58.3%
The Best Farm	3	72.7%	36.4%
The Best Farm	4	64.0%	48.0%
The Best Farm	5	66.7%	45.5%
The Best Farm	6	91.7%	62.5%
The Best Farm	7	80.0%	65.7%
The Circulatory System	1	76.7%	50.0%
The Circulatory System	2	75.8%	63.6%
The Circulatory System	3	81.5%	63.0%
The Circulatory System	4	87.1%	71.0%
The Circulatory System	5	73.5%	61.8%

Table 15. Percent Accuracy of the Resulting KNN Models by Chapter

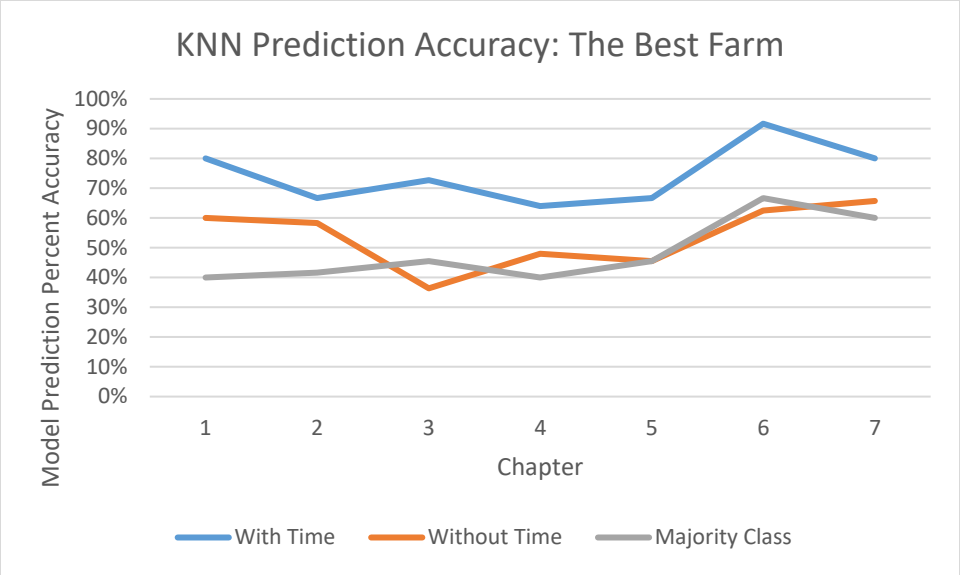

Figure 6. Percent Accuracy of the Resulting Models for the KNN Algorithm for The Best Farm

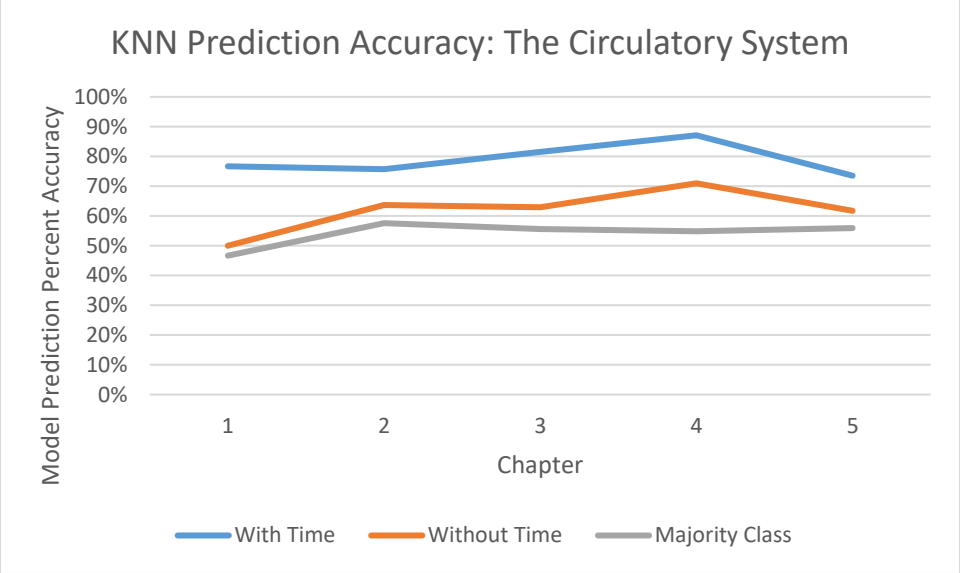

Figure 7. Percent Accuracy of the Resulting Models for the KNN Algorithm for The Circulatory System

Story	With Time	Without Time	Majority Class
The Best Farm	54.9%	51.8%	42.6%
The Circulatory System	67.1%	52.3%	50.3%

Table 16 Percent Accuracy of the Resulting KNN Models Using Samples by Story

Story	Chapter	Percent Above Random Samples
The Best Farm	1	43%
The Best Farm	2	45%
The Best Farm	3	30%
The Best Farm	4	9%
The Best Farm	5	33%
The Best Farm	6	98%
The Best Farm	7	67%
The Circulatory System	1	43%
The Circulatory System	2	2%
The Circulatory System	3	71%
The Circulatory System	4	94%
The Circulatory System	5	45%
Average		48%

Table 17. ESDT: Percent of Random Samples Which KNN Outperforms

Story	Chapter	Rank 1	Rank 2	Rank 3	Rank 4
The Best Farm	1	I-SS-I	X-SS-NS	NS-S-NS	C-L-X
The Best Farm	2	H-L-X	C-L-H	H-S-NS	C-S-H
The Best Farm	3	X-SL-H	I-S-C	X-SS-NS	C-SL-X
The Best Farm	4	NS-L-X	C-SL-NS	H-S-H	H-S-C
The Best Farm	5	X-L-H	NS-SL-NS	C-SL-C	NS-SS-I
The Best Farm	6	C-S-NS	H-SS-C	C-SS-NS	NS-SL-NS
The Best Farm	7	H-SL-C	H-L-NS	C-SL-NS	NS-S-NS
The Circulatory System	1	H-SS-NS	NS-SL-NS	C-L-NS	NS-S-NS
The Circulatory System	2	NS-L-I	C-SS-C	C-L-NS	NS-SS-NS
The Circulatory System	3	X-SL-I	NS-S-I	H-S-NS	H-SS-NS
The Circulatory System	4	H-SS-C	X-L-NS	X-SS-C	H-L-I
The Circulatory System	5	X-S-H	C-L-C	X-S-I	C-SS-NS

Table 18. Top 4 Extracted With Time Features for the KNN Algorithm by Chapter

Story	Chapter	Rank 1	Rank 2	Rank 3	Rank 4
The Best Farm	1	I-I	C-H	C-C	I-C
The Best Farm	2	NS-I	X-H	H-NS	C-NS
The Best Farm	3	NS-I	C-H	H-I	NS-C
The Best Farm	4	NS-NS	H-NS	H-H	C-H
The Best Farm	5	NS-NS	H-NS	C-NS	I-H
The Best Farm	6	I-C	C-I	I-I	H-C
The Best Farm	7	X-I	H-H	NS-NS	H-NS
The Circulatory System	1	H-NS	NS-NS	C-NS	I-H
The Circulatory System	2	C-H	C-C	NS-C	C-NS
The Circulatory System	3	H-H	H-C	NS-C	C-C
The Circulatory System	4	NS-NS	I-I	H-I	H-NS
The Circulatory System	5	H-H	H-I	NS-C	C-C

Table 19. Top 4 Extracted Without Time Features for the KNN Algorithm by Chapter

The CLR algorithm does not use randomness and thus was not considered for phase 1. For phase 2, the N chosen was 3, with an average accuracy score of 29%. In phase 3, the Student's t-test did not reveal a statistically significant difference $p = 0.457$. The resulting accuracies are shown in Table 20.

Story	Chapter	With Time	Without Time
The Best Farm	1	20.0%	30.0%
The Best Farm	2	16.7%	25.0%
The Best Farm	3	36.4%	9.1%
The Best Farm	4	40.0%	32.0%
The Best Farm	5	42.4%	45.5%
The Best Farm	6	12.5%	50.0%
The Best Farm	7	48.6%	48.6%
The Circulatory System	1	26.7%	26.7%
The Circulatory System	2	21.2%	42.4%
The Circulatory System	3	33.3%	22.2%
The Circulatory System	4	19.4%	25.8%
The Circulatory System	5	20.6%	23.5%

Table 20. Percent Accuracy of the Resulting CLR Models by Chapter

The SLR algorithm does not use randomness and thus was not considered for phase 1. For phase 2, the N chosen was 9 with an average R^2 score of -12.1. The resulting R^2 scores are shown in Table 21. For phase 3, the Student's t-test did not reveal a statistically significant difference, $p = 0.380$.

Story	Chapter	With Time	Without Time
The Best Farm	1	-4.101496993	-31.5411846
The Best Farm	2	-1.255183625	-9.393707568
The Best Farm	3	-1.723208563	-28.15613085
The Best Farm	4	-3.433632171	-4.019095111
The Best Farm	5	-64.09981886	-7.352678136
The Best Farm	6	-3.763410436	-8.509814952
The Best Farm	7	-19.02698078	-53.60974555
The Circulatory System	1	-1.969445433	-6.497129097
The Circulatory System	2	-5.732203647	-1.270804528
The Circulatory System	3	-0.036651433	-29.92367632
The Circulatory System	4	0.145165095	-1.443626902
The Circulatory System	5	-1.399222973	-1.172795161

Table 21. R^2 Score of the Resulting SLR Models by Chapter

CHAPTER 8

DISCUSSION

This project investigated the predictive power of elapsed time as a feature. The analyzed data comes from a study of 96 Latino children using the EMBRACE application (Walker, Adams, Restrepo, Fialko, & Glenberg, 2017). A set of 92 features were extracted using elapsed time as a feature and a set of 23 features were extracted without elapsed time as a feature. Several data mining algorithms were used to create models to predict participant reading comprehension. The algorithms were used to create models using the set of features with time to compare against models using the set of features without time. The six models gave contradictory results. The ANN and boosted RF produced sporadic results. The KNN and SVM models using the samples by chapter indicated that the elapsed time between steps is important. The SLR and CLR models and the KNN models using samples by story indicated that the elapsed time between steps is not important. Despite the contradictory results, there is important information regarding creating a reading comprehension prediction model.

Given the success of other models, the contradictory results are surprising. Baker, Gowda, and Corbett (2010) and Baker et al. (2012) both obtained well-fitting LR models using similar features. Even though the data sets were different[5][6] and covered different subject materials, the similarity of the extracted features would lead one to expect that the

[5] Baker, Gowda, and Corbett (2010) and Baker et al. (2012) both had different starting attributes than the attributes used in this project.
[6] Baker, Gowda, and Corbett (2010) and Baker et al. (2012) both had 71 students as opposed to the 20-36 samples used per chapter in this project.

resulting model would be somewhat well-fitted. This expectation is in contrast to the SLR models' low R^2 scores.

Despite the fact that the LR algorithm used in the literature produces continuous predictions, one would expect other algorithms to have similar predictive power given the similarity of the feature set. In this case, the expected results match the KNN and SVM results. The KNN and SVM models using the with time features were consistantly more accurate than the majority class.

The ANN and boosted RF did not pass phase 1. This may be due to the small sample size. ANNs and RFs rely on a large number of training samples, otherwise, the resulting model's behavior will be unpredictable due to insufficient training. Therefore, the ANN's and boosted RF's failure to pass phase 1 may only be due to the small sample size.

The accuracy of the KNN models for features with time performed statistically significantly better than the KNN models for features without time. This supports the explanation that the elapsed time as a feature contains predictive value. Analysis of the selected feature subset for the KNN with time feature set did not reveal consistently chosen features. This may be explained by the fact that the elapsed time was discretized independent of each chapter. Even if a participant took the same amount of actual time in between two steps for one chapter, the elapsed time may have been discretized differently depending on the relative speed of the other actions in a different chapter. However, the most common action type to appear in the features was the New Sentence action type. It

appeared in 26 out of 48 possible features (4 features * 12 chapters) and appeared at least once in the features selected for each chapter.

There are several reasons why the New Sentence action type was the most common action type selected. First, the rate at which the participant advanced, either at the end of a sentence or at the beginning may be reflective of the participant's reading comprehension. A participant who could easily read the sentence needed to spend proportionately less time reading the sentence than participants who could not easily read the sentence. Second, the rate at which the participant advanced, either at the end of a sentence or at the beginning may be reflective of how well the participant stayed on task. For instance, a participant who was engaged in learning the material covered might have spent proportionately more time at the beginning of the sentence in order to read it thoroughly than a participant who was disengaged from the task. Similarly, a participant who was disengaged from the task might have spent proportionately longer advancing to the next sentence, because the participant was distracted, compared to a participant who remained engaged with the task. If we consider the action type New Sentence as the beginning of a task and the other action types as steps in that task, then the importance of the New Sentence action type is consistent with the literature. Wang and Heffernan (2012) use an existing Knowledge Tracing algorithm and the first response time to obtain a linear regression model which improved the existing Knowledge Tracing algorithm.

The accuracy of the SVM models for with time features set performed statistically significantly better than the SVM models for the without time features set. This supports the explanation that the elapsed time as a feature contains predictive value. Analysis of

the selected features subset for the SVM with time features set did not reveal any consistently chosen feature. Again, the New Sentence action type appeared frequently in the selected features for the with time feature set. The New Sentence action type appeared in 23 out of the possible 48 features and appeared at least once in the features selected for each chapter.

The accuracy of the CLR models for the with time features set did not perform statistically significantly different than the CLR models for the without time features set. This lack of difference combined with the very low R^2 scores of the SLR models support the explanation that the elapsed time between steps does not have predictive value for predicting the participant's reading comprehension.

The KNN models using the samples by story for features with time outperformed a model which guesses the majority class. The improvement was 12 percent for The Best Farm and 17 percent for The Circulatory System. This improvement indicates that it is possible to create an accurate reading comprehension model using action-based log data. For The Best Farm, the KNN model using the samples by story for features with time outperformed the model for features without time by 3 percent. This small improvement indicates that elapsed time is not an important feature. However, this statement is contradicted by the 12 percent increase in accuracy for The Circulatory system for the model using features with time over the model using features without time.

In addition, the low average ESDT (less than 50%) and wild variation in percentage between chapters also supports the explanation that the elapsed time between steps has little predictive value for predicting the participant's reading comprehension for

this data set. Because the ESDT uses the same feature selection algorithm for the actual time sets as well as the 100 random time sets, it contradicts the explanation that the selected features for the KNN models are not selected because of features coincidentally aligning, causing overfitting. If the features were chosen due to some intrinsic predictive ability rather than overfitting, then the selected features for the actual data should be consistently above average. In addition, the perturbation in accuracy indicates that the features selected in the actual data are chosen because of overfitting, as opposed to an intrinsic predictive value of the selected features.

The contradictory nature of the results may be due to several causes. First, the log data could have failed to capture an important feature. For instance, the log data does not record the participants' attention to their task. If there is a more strongly correlated feature which is not recorded, then the effects of the extracted features will be diminished. Second, the sample size may be too small[7]. Many data mining algorithms assume an abundance of data. However, once the participants in the experimental group with complete log data and post-test assessment were selected, each chapter had 20-36 samples. The small sample size can especially pose a problem for algorithms such as ANN. Third, the relationship between elapsed time and reading comprehension may not be linear. This would explain the poor performance of the CLR and SLR models. Fourth, time may need to be represented differently. Instead of the elapsed time between each action, perhaps only the elapsed time for the first action should be used. Also, the elapsed

[7] Each chapter had 20-36 samples.

time between actions may be discretized per the time taken in between previous actions or subsequent actions.

Further research on the subject of using time as a feature in predicting participant reading comprehension may come from representing time differently (Wang & Heffernan, 2012). For instance, a potential feature may be the amount of time for the first occurrence of a sequence of action types. Another way in which time could be represented differently might come from representing elapsed time between two actions relative to the adjacent elapsed time. An example of this would be how much longer or shorter the elapsed time is compared to the elapsed time that immediately follows or precedes that elapsed time. Another way time may be used is by considering the amount of time spent on a concept (Wang & Beck, 2012).

Based on the contradictory nature of the results, future projects should take steps to resolve the discrepancy. Any future projects done with this corpus should include more of the samples provided by either extracting the features using a different method or finding a way to include partial data. Ideally, any future work should use a larger data set. This will help resolve the difficulties with the data mining algorithms using small sample sizes.

CHAPTER 9

CONCLUSION

The results of the KNN and SVM models using samples by chapter indicate that the elapsed time between actions is a useful feature for predicting participant reading comprehension. However, the CLR and SLR models using samples by chapter and the KNN models using samples by story indicate that the elapsed time in between actions is not a useful feature for predicting participant reading comprehension. These results indicate that the answer to the second research question (Does timing information improve the accuracy of reading comprehension predictions over user actions alone?) may be yes. If we consider that the LR models may have experienced a floor effect because the classification of the data may not linear, then the KNN and SVM models using samples by chapter indicate that there is a significant improvement in model prediction using timing information over user actions alone. Without being able to confirm why the LR models performed poorly, any conclusion about the second research question will require a degree of caution.

The answer to the first research question (Can reading comprehension be accurately predicted using action-based logging?) may also be yes. If we again consider that the LR models may have experienced a floor effect because the data may not linear, then the KNN and SVM models using samples by chapter and the KNN models using samples by story support that the answer is yes. The KNN and SVM using samples by chapter, using the features with time, outperformed a model which guesses the majority class by 17-40 and 16-40 percent respectively. The KNN models using samples by story,

for the features with time, outperformed a model which guesses the majority class by 12 percent for The Best Farm and 17 percent for The Circulatory System. Again, the poor performance of the LR models requires any conclusion about the first research question to be cautious.

The ESDT indicates that the features selected for the KNN models may have been chosen due to overfitting. Furthermore, use of the ESDT may question the way feature selection is done in educational data mining. This statement comes with several caveats. First, the sample size used for model creation for this project was very small compared to other data sets. Second, the data used for model creation for this project involved a large amount of data processing. Third, there were 92 features used in this project and thus there was an increased chance that a random feature will fit due to coincidence. Fourth, the ESDT is not a proven statistical metric. It was derived specifically for this project and, as such, has not undergone any testing of its statistical rigor.

Limitations

One of the most severe limitations of this project was the small sample size used for the models. Many data mining models require a large sample size in order to function properly. The small sample size used in this project may have caused the contradictory results of the models. An attempt was made to address this issue by using samples by story. This increased the sample size from 20-36 per chapter to 195 for The Best Farm and 155 for The Circulatory System. However, increasing the total number of samples will increase the validity of analysis of models for individual chapters.

Another limitation is the data set used. The log data contains records of many participants who did not complete a chapter or who had to restart a chapter. This may not pose a problem for the original circumstances under which the data was recorded. However, it can lead to unusual results when data mining, as it forces those samples to be removed.

Another limitation is the high degree of data transformation during the feature extraction process. During the feature extraction process, some important information may have been filtered out. In addition, the resulting features sometimes had very little differentiation between samples. For a given chapter, some of the features only varied by a value of 1 for only 1 sample, see Table 6, H-SS-NS feature.

Another limitation is the number of questions used in the post-test assessment. The participants were graded on a binary scale for 5-7 questions. This, combined with the small sample size, makes creating a balanced discrete classification a challenging task. There may be too many participants who obtained the same score for one chapter to create a balanced classification.

Future Work

While the results obtained in this project contradict each other, valuable lessons can be learned. First, different data mining algorithms can produce significantly different results using the same set of features. The KNN and SVM models using samples by chapter KNN models using samples by story had good accuracy while the CLR models had poor accuracy and the SLR models were poorly-fitted. Second, the results contradict what was expected given the previous literature. The literature indicated that the LR

models should have been well-fitted, but the created models were poorly-fitted. Third, the results of the ESDT suggest that the feature selection used by some previous published works may require more validation. The discrepancy between the high accuracy of the KNN and the low ESDT score imply that the feature selection algorithm may be choosing coincidental features. These three lessons indicate that while timing appears to be an important feature for reading comprehension prediction, more investigation is required before it can be widely used.

Future projects developing a language comprehension prediction model should explore how time may be represented as a feature. Potentially, future projects should examine the first response time, comparing elapsed time to adjacent elapsed times, or consider the amount of time spent on a concept. Future research should also consider the interactions between features.

Future language comprehension prediction models should consider other aspects as features, for instance, the proportion of times that the incorrect image was dragged to the correct image. As further research is done, the predictive ability of reading comprehension models will increase. This will allow EMBRACE to predict the needs of the users. This will allow EMBRACE to better serve as an English language tutor, which will improve the quality of life for English language learners.

REFERENCES

Beck, J. E., Jia, P., & Mostow, J. (2004). Automatically Assessing Oral Reading Fluency in a Computer Tutor that Listens. *Technology, Instruction, Cognition and Learning*, 61-81.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences*, 167-207.

August, D., & Shanahan, T. (2006). Executive Summary. *Developing Literacy in Second-Language Learners: Report of the National Literacy Panel on Language-Minority Children and Youth*, 1-9.

Baker, R. S., Gowda, S. M., & Corbett, A. T. (2010). Automatically Detecting a Student's Preparation for Future Learning: Help Use is Key. *Educational Data Mining 2011*.

Baker, R. S., Gowda, S. M., Corbett, A. T., & Ocumpaugh, J. (2012). Towards Automatically Detecting Wheter Student Learning is Shallow. *International Conference on Intelligent Tutoring Sytstems*, (pp. 444-453). Springer Berlin Heidleberg.

Collins-Thompson, K., & Callan, J. (2004). Information Retrieval for Language Tutoring: An Overview of the REAP Project. *In Procedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 554-545).

Corbett, A., Kauffman, L., MacLaren, B., Wagner, A., & Jones, E. (2010). A Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. *Journal of Educational Computing Research*, 219-239.

Curuana, R., & Freitag, D. (1994). Greedy Atrribute Selection. *In Proceedings of the Eleventh International Conference on Machine Learning*, (pp. 28-36).

Glenberg, A. M., Gutierrez, T., Levin, J. R., Japuntich, S., & Kaschak, M. P. (2004). Activity and Imagined Activity Can Enhance Young Children's Reading Comprehension. *Journal of Educational Psychology*, 424.

Glenberg, A., Willford, J., Gibson, B., Goldberg, A., & Zhu, X. (2012). Improving Reading to Improve Math. *Scientific Studies of Reading*, 316-340.

Heilman, M., & Eskenazi, M. (2006). Language Learning: Challenges for Intelligent Tutoring Systems. *In Proceedings of the Workshop of Intelligent Tutoring Systems for Ill-Defined Tutoring Systems. Eighth International Conference on Intelligent Tutoring Systems*, (pp. 20-28).

Heilman, M., & Eskenazi, M. (2008). Self-Assessment in Vocabulary Tutoring. *In International Conference on Intelligent Tutoring Systems*, (pp. 656-658).

Ma, W., Adesope, O. O., Nesbit, J. C., & Lui, Q. (2014). Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology*, 901-918.

McNamara, D. S., Levinstein, I. B., & Boonthum, C. (2004). iSTART: Interactive Strategy Training for Active Reading and Thinking. *Behavior Research Methods*, 222-233.

Mostow, J. (2012). Why and How Our Automated Reading Tutor Listens. *In Proceedings of the Internation Symposium on Automatic Detection of Errors in Pronunciation Training (ISADEPT)*, (pp. 43-52).

National Center for Education Statistics. (2015). *Table 204.20 Number and Percentage of Public School Students Participating in Programs for English Language Learners, by State.* Retrieved from https://nces.ed.gov/programs/digest/d15/tables/dt15_204.20.asp.

Razzaq, L., Patvarczki, J., Almeida, S. F., Vartak, M., Feng, M., Heffernan, N. T., & Koedinger, K. R. (2009). The ASSISTment Builder: Supporting the Life Cycle of Tutoring System Content Creation. *IEEE Transactions on Learning Technologies*, 157-166.

Tam, Y.-C., Mostow, J., Beck, J., & Banerjee, S. (2003). Training a Confidence Measure for a Reading Tutor that Listens. *INTERSPEECH*.

VanLehn, K. (2006). The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 227-265.

Walker, E., Adams, A., Restrepo, M. A., Fialko, S., & Glenberg, A. M. (2017). When (and How) Interacting With Technology-Enhanced Stroybooks Helps Dual Language Learners. *Translational Issues in Psychological Science*, 66-79.

Wang, Y., & Beck, J. E. (2012). *Incorporating Factors Influencing Knowledge Retention into a Student Model Educational Data Mining*.

Wang, Y., & Heffernan, N. T. (2012). Leveraging First Response Time into the Knowledge Tracing Model. *Educational Data Mining*.