Using the Tablet Gestures and Speech of Pairs of Students to Classify Their Collaboration

by

Sree Aurovindh Viswanathan

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

Approved July 2014 by the
Graduate Supervisory Committee:

Kurt VanLehn, Chair
Michelene T.H. Chi
Erin Walker

ARIZONA STATE UNIVERSITY

August 2014

ABSTRACT

This thesis is an initial test of the hypothesis that superficial measures suffice for measuring collaboration among pairs of students solving complex math problems, where the degree of collaboration is categorized at a high level. Data were collected in the form of logs from students' tablets and the vocal interaction between pairs of students. Thousands of different features were defined, and then extracted computationally from the audio and log data. Human coders used richer data (several video streams) and a thorough understand of the tasks to code episodes as collaborative, cooperative or asymmetric contribution. Machine learning was used to induce a detector, based on random forests, that outputs one of these three codes for an episode given only a characterization of the episode in terms of superficial features. An overall accuracy of 92.00% (kappa = 0.82) was obtained when comparing the detector's codes to the humans' codes. However, due irregularities in running the study (e.g., the tablet software kept crashing), these results should be viewed as preliminary.

Dedicated to my affectionate dad,

Dr. Viswanathan Palanivel

ACKNOWLEDGEMENTS

I completely realize that this thesis work would not have been possible without sharing of thoughts and ideas from so many smart and kind people. My advisor, Dr. Kurt VanLehn, is one of the most knowledgeable and supportive person I have ever known, and I thank him for his never –ending passionate support and constant encouragement throughout the course of two years. He was willing to clarify my doubts and was able to provide me with motivation at all times. A simple word of thanks would not substitute for all the help and time he has given me through these years and I am indebted to him for the same.

Besides my advisor, I would like to thank the rest of my committee: Dr. Michelene T.H. Chi, Dr. Erin Walker and faculty member Dr. Heather Pon-barry for their insightful comments and constructive feedback for the improvement of my thesis.

Many thanks to Sujith and all the members of the FACT team who were instrumental in the development of the FACT software system, which was used in this thesis. I would like to thank each one of them for their ideas and for taking part in many, sometimes exhaustive coding discussions. I personally would like to thank Sujith, Aniket Patil, Arun Reddy, Kenil Bhatt, Leo Chen, Lishan Zhang, Matt Schmidt, Megha Tiwari, Refika Kosler, Pradeep Chaudhari, Srinivasaragavan A. Rajamohan, Karthick Jayaraman, Vallabh Patade and Vignesh Kathireshan. Their friendship and assistance helped me a lot and my research would not have been possible without their immense support. I am also thankful to Arun Reddy and Nichola Lubold for helping me in designing and conducting these research studies.

Finally, I would like to express my gratitude towards my family. Even though there were thousands of miles apart, I knew in the bottom of my heart that they are always there to help and support me.

I would like to thank my father, Dr. Viswanathan Palanivel who had been a constant source of motivation and had expressed love and affection at all times. His prayers and blessings have made me to come up to this stage.

The mentoring provided by my sister Dr. Priyaa Viswanathan, is beyond words. She has always been there to hear all my rant and was there to guide and motivate me every day. They were my best well-wishers and the most affectionate people I could ever ask and I thank god for providing them in my life.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Learning to collaborate with student peers is an important skill to practice. Teachers often create small student groups so that they learn by sharing their ideas and working together towards a common goal of solving the problem at hand. This common practice has been shown to exhibit many advantages such as increased learning gains (Chi, Roy, & Hausmann, 2008), spontaneous generation of ideas and higher levels of cognitive reasoning (Hausmann, Chi, & Roy, 2004), group awareness and the ability to work well as a team. However, it is also shown that collaboration is not spontaneous and grouping people together does not always entail effective exchange of ideas (Dillenbourg, 1999).

In order to facilitate this process, intervention from the facilitator becomes necessary. However, the number of parallel groups that coordinate with each other in a typical class room environment can become high and may too overwhelm the facilitator unless the facilitator can skip some groups and only help those that need help. Hence, a system that can provide real time feedback about the quality of collaboration and perhaps the progress of the task becomes indispensable for the facilitator.

Measurement of the collaboration process generally is complex as it involves "observing, capturing and summarizing complex tasks performed by the individual and their group contributions of the subject to the group" (Gress, Fior, Hadwin, & Winne, 2010). Although measurement of collaboration by human coders requires a thorough understanding of the tasks and the individuals' contributions, it may be that superficial measures, such as the dimensions described in section 2, may suffice as well. Once these measurements are obtained, a classical machine learning algorithm could be induced in order to predict the labels which would match with human judgment.

This thesis is an initial test of the hypothesis that superficial measures suffice for measuring collaboration among pairs of students solving complex math problems, where the degree of collaboration is categorized at a high level. Data were collected in the form of logs from students' tablets and the vocal interaction between pairs of students. Thousands of different features were defined, and then extracted computationally from the audio and log data. Human coders used richer data (several video streams) and a thorough understand of the tasks to code episodes as collaborative, cooperative or asymmetric. Machine learning was used to induce a detector, based on random forests, that outputs one of these three codes for an episode given only a characterization of the episode in terms of superficial features. An overall accuracy of 92.00% (kappa = 0.8232) was obtained when comparing the detector's codes to the humans' codes. However, due irregularities in running the study (e.g., the tablet software kept crashing), these results should be viewed as preliminary.

The remainder of the thesis document is organized as follows. Chapter 2 introduces terminology from the field of Computer Supported Collaborative Learning (CSCL) followed by summary of relevant background research. Chapter 3 describes the research problem, its scientific importance and its practical implications. Chapter 4 describes the study's experimental setup, the problem domain and data collection. Chapter 5 describes the method that was used to preprocess the data, feature extraction and the machine learning algorithms. Chapter 6 describes the results that were obtained from the study followed by its conclusion and further steps.

CHAPTER 2

TERMINOLOGY AND BACKGROUND

This chapter reviews the literature related to the measurement of collaborative learning. In particular, it summarizes previous attempts at using machine learning algorithms to measure collaborative learning.

When students work in a group, their behavior only sometimes qualifies as collaborative. Dillenbourg et al. makes a clear distinction between collaborative and cooperative learning (Dillenbourg, Baker, Blaye, O'Malley, & others, 1995). Collaborative learning could be defined as “a situation in which students in the group complete their task together by mutual engagement of the participants in a coordinated effort”. In other words, students work together with a common goal in mind. Cooperative learning could be defined as a situation in which students in a group complete their task by dividing it into subtasks and each subtask is solved with substantial amount of individual effort. In other words, students work on the problems with different immediate goals in mind. Johnson and Johnson (1996) further divide cooperative learning into competitive and individualistic learning. In individualistic learning, each student attempts to solve a subtask on their own while in competitive learning they compete with other students in the class in order to accomplish a learning goal.

A brief account of theories of collaboration would probably need to mention only two major approaches. The Piagetian approach (Socio-constructionist approach) portrays that when students interact with other, the participant improves or masters a particular concept (Doise, 1990). On the other hand, the Vygotskian approach (socio-cultural approach) (Vygotskii) focuses on causal relationships between social interaction and individual cognitive change. In other words, the socio-constructionist theory considers collaboration as a black box and evaluates the final outcome (or impact) on the individual once the learning session is complete. On the other hand, the socio-cultural approach focuses on the entire process and considers the relationship between social interaction and individual learning gains (Dillenbourg et al., 1995). Both Piagetian and Vygotskian theories acknowledge the intertwined aspects of social and individual development.

Quantification of collaboration has been done by Meier, Spada & Rummel (2007). This work distinguishes nine dimensions: task orientation, technical orientation, reciprocal orientation, time management, task division, mutual understanding, dialogue management, consensus and information pooling. Typically, these are rated by human judges working from video recordings or transcripts.

Another approach to measuring collaboration is to measure characteristics exhibited in students' speech and action when they collaborate effectively with each other. These processes are described in more detail in the section below, one per paragraph.

2.1 COMMON GROUND

Grounding among participants describes the mutual awareness of shared knowledge among participants of the group (H. H. Clark & Brennan, 1991; H. H. Clark & Schaefer, 1989). When a concept is grounded, the participants make sure that others have the same understanding of a particular concept as they do. They often clarify with questions that further lead to the explanation of the concept. Thus, the participants make sure that they understand each other properly.

2.2 KNOWLEDGE CONVERGENCE

Whereas grounding is often used for all kinds of meaning in conversation, including transitory ones (e.g., which week are we referring to here?), *knowledge convergence* refers only to information that could be called knowledge in that it has some permanent or durable value (Brown & Campione, 1994; Vygotskii; Webb & Palincsar, 1996). Knowledge convergence often happens as a by-product of collaboration. When this occurs, there students tend to learn from each other.

2.3 TRANSACTIVE CONTRIBUTIONS

Transactive contributions (Gweon, Jain, McDonough, Raj, & Rosé, 2013; Weinberger & Fischer, 2006) could be concisely defined as "reasoning that builds on prior reasoning statements". When a pair of students engage in a productive conversation, they not only produce reasoning statements pertaining to the problem but also add or build on the reasoning statement of the previous person. These types of statements provides an opportunity of the pair of students to learn from each other. Transactivity learners' discourse is positively related to individuals knowledge acquisition (Teasley, 1997).

2.4 SELF-DIRECTED VS OTHER DIRECTED LEARNING

Self-oriented dialogues (Chi, De Leeuw, Chiu, & LaVancher, 1994) occur when a student reasons about a particular concept by thinking aloud. This not only provides an opportunity for him to learn but also provides a chance for his peer to correct any oversight in his understanding. In contrast, Other oriented dialogue exchanges (Shirouzu, Miyake, & Masukawa, 2002) happen when one person explains the concept to the other person while they solve the problem together. This type of exchange help the speaker to understand more about the concepts about the particular task at hand.

2.5 INITIATIVE

Effective collaboration happens when partners take equal initiatives when they solve a problem together. A conversation is defined as bidirectional when there is a two way flow of contextual information governed by transfer of control between two participants. Several studies (Chu-Carroll & Brown, 1997; Walker & Whittaker, 1990), have shown that the dialogue and task initiative exhibited by participants is a good measure of collaboration.

2.6 SCAFFOLDING CONTRIBUTIONS

In group activity, students sometimes try to elicit information from their peers by asking questions. Such scaffolding (Gibbons, 2002; Nussbaum et al., 2009) has been extensively studied in the literature and it has been shown as an important factor which fosters learning. So, presence of scaffolding activity provides evidence that students are contributing to the learning of their peers by questioning their assumptions about various concepts relating to the task domain.

2.7 INDEPENDENT VS JOINT CONSTRUCTION OF KNOWLEDGE

When students are assigned a collaborative problem solving task, they either solve the problem together by building on each other's ideas or they segment the overall problem into small sub problems which are then solved independently. It has been shown that students who co-construct (Hogan, Nastasi, & Pressley, 1999; Tao & Gunstone, 1999) ideas have been more successful in extending the knowledge of their peers than students who work independently on solving a problem. Identification of such processes in a collaborative learning environments would allow us to understand the degree to which these students work with each other.

2.8 CONFLICT RESOLUTION AND TIME MANAGEMENT

When students interact about a concept, differences in cognitive understanding can lead to conflict. For learning to occur, it is essential for the students to explain their side of reasoning and come to a common consensus. If the consensus is not reached, the mere presence of conflict does not entail learning (Chi, 1992).

2.9 REFLECTION

After the collaboration activity is complete, students tend to learn more if they are provided with opportunities to understand their mistakes (Hmelo-Silver, 2003) that have been made during collaborative activity by retrospection on the products of collaboration such as drafts or notes (Baker & O'Neil, 2002).

2.10 PRIOR WORK ON MEASURING COLLABORATION

Although many other observable characteristics of collaborative learning could described, the list above should suffice to give the reader an idea of the complexity of the construct. The process dimensions discussed above have been coded in a conversation by a variety of researchers (Biggs & Collis, 1982; Bloom, College, & Examiners, 1956; CHAMRADA & MANSOUR, 2009; D. B. Clark & Sampson, 2008; Dillenbourg et al., 1995; Garrison, Anderson, & Archer, 2001; Gweon, 2012; Harrer, McLaren, Walker, Bollen, & Sewall, 2006; Hausmann et al., 2004; Hmelo-Silver, 2003; McLaren, Bollen, Walker, Harrer, & Sewall, 2005; McLaren et al., 2007; McLaren, Scheuer, & Mik\vsátko, 2010; Noroozi, Biemans, Busstra, Mulder, & Chizari, 2011; Schrire, 2006; Tao & Gunstone, 1999; Veldhuis-Diermanse, 2002; Wang et al., 2007; Weinberger, Stegmann, & Fischer, 2007). A brief summary of coding categories have been provided in Appendix A.

Given the complexity of collaborative behaviors, as evident in the list above and the elaborate coding schemes of Appendix A, it seems reasonable to use machine learning to detect common patterns of behavior that characterize students who are collaborating.

The following presents a brief summary of the set of techniques that has been used by researchers in this field.

- Soller et al (2002) used hidden Markov model that would identify moments of knowledge sharing interactions that differentiates between effective and ineffective students.
- Kinnebrew and Biswas (2012) presented a differential sequential mining (DAM) technique that looks for sequence of patterns that would differentiate between high and low collaborative students.
- Anaya and Boticario (2011) developed a method that determines the level of collaborative activity by classification and clustering.
- Duque and Bravo (2007) developed a fuzzy model which would classify different forms of collaboration by generation of production rules.
- Roman et al (2012) explored patterns of collaboration by using simple measures of speech presence that distinguishes groups based on the level of collaboration exhibited by them.
- Jain et al (2012) and Gweon et al (2013) deployed unsupervised dynamic Bayesian modeling approach in order to detect speech style accommodations and for estimating other oriented transactivity properties respectively.
- Martinez-Maldonado and his colleagues (Martinez-Maldonado, Kay, & Yacef, 2013; Martinez, Collins, Kay, & Yacef, 2011) developed a classification model, sequence mining and hierarchical clustering that could differentiate between high and low collaboration groups.

Although several machine learning detectors have been built, measuring the process automatically becomes a daunting task due to the nature of the collaboration. Hence, most machine learning detectors, requires human segmentation followed by annotation. In this thesis, a preliminary step towards automation of such process is attempted. Here segmentation of data is performed automatically and then human annotation is performed on this segmentation.

CHAPTER 3

RESEARCH PROBLEM

The long term goal of this research is to automate the detection and measurement of collaboration from various media sources such as audio, video and log files. However, for this current thesis the goal is more modest. Its goal is to automatically identify, at a very high level, the group dynamics of the subjects who work with each other. In-order to achieve this, a classifier was induced from data that were labelled by a human judge.

More specifically, the project involves the following major goals

i. To collect rich data from pairs of students who solve mathematical problems together.
ii. To segment the data algorithmically from the data available in the log file.
iii. To annotate the segmented data (by hand) for three types of group dynamics: collaboration, cooperation and asymmetric contribution.
iv. To computationally extract features from student actions (from log file) and the student conversation (from speech data)
v. To use standard induction algorithms to induce a classifier that inputs the computed features and outputs group dynamic labels that are highly correlated with human judgments.
vi. To come up with a preliminary visualizations that would help the teachers understand how these group dynamics change across time.

The following sections briefly describe the scientific importance of performing research on collaboration and describes the practical usage of these methods.

3.1 SCIENTIFIC IMPORTANCE

When students interact with each other, the actual mental and cognitive process that a person undergoes cannot be visualized as it unfolds in the student's mind. On the other hand, we are only able to monitor the external manifestations of the student such set of actions that student performed, the reactions the student expressed and the set of circumstances that were present when the student was engaged in cognitive reasoning. In order to understand this process better, it becomes important for us to collect micro level data about all the set of actions that the student performs and monitor each action closely in order to understand this process better. Once the data are collected, various pattern recognition and machine learning algorithms could be deployed in order to understand and differentiate between scenarios that lead to high and low levels of collaboration. The patterns thus discovered may help us understand collaboration better.

However, even if studying the patterns induced by machine learning fails to clarify the nature of collaboration, it will be helpful to have a consistent and reliable measure of collaboration. In the current state of affairs, there is no commonly accepted, operational definition of collaboration. Thus, when different researchers apply different definitions of collaboration, they may actually be studying slightly different phenomena, which makes it difficult to accumulate findings and a deeper understanding of collaboration per se. Although it will take considerable work to develop machine-learned measures that are widely accepted and operational, this thesis is a step in that direction.

3.2 PRACTICAL UTILITY

In typical classroom environments, teachers often assign students to work in small groups. However, students don't spontaneously engage in a meaningful interaction and effectively collaborate with each other and yet, working effectively in collaborative environments is increasingly important in education (Dillenbourg, 1999). Hence, it becomes the responsibility of the teacher to motivate the students to collaborate well with each other. However, teacher faces the following problems when they have to orchestrate multiple groups working in a face to face environment.

i. Teachers should be able to choose the right group to attend (Dillenbourg et al., 2011).
ii. Teachers need to know which members, if any, remains passive in the group. This knowledge helps the teachers to motivate the student and to clarify his doubts on the subject.
iii. Teachers need to determine the right time to intervene when multiple parallel discussions are happening in a class (Dillenbourg et al., 2011).
iv. Teachers need to know how much time to spend with each group so that they can divide their time effectively across multiple groups.
v. Teachers often see the final product or end result of the problem solving activity and use it to decide on the correctness of the problem solving process (Race, 2001). However, teachers should judge the acceptability of the group's process based on data about that process.
vi. When there are many groups working simultaneously in the classroom, the teacher may not be in a position to provide individual attention to every group and thus must decide which groups not to visit at all.

These issues suggest that an automatic system that reliably provides collaboration classifications may become indispensable for teachers.

CHAPTER 4

PROJECT DESCRIPTION

This chapter presents the mathematical, physical and software setup that were used to collect data. The mathematical problem used for this study is called the distance time interpretation problem, and it is described in section 4.5.

4.1 HARDWARE DESCRIPTION

Students work on Samsung Galaxy note 10.1, which is a 10 inch tablet with an active digitizer technology. This stylus enables students to write on the tablet as they would do on a normal paper. The server used for this setup is Intel Quad Core I7 processor with 16 GB of RAM and runs windows 7 as the operating system. The tablet and the laptop computer (server) communicate with each other using a local Wi-Fi network that is not connected to the internet.

4.2 SOFTWARE DESCRIPTION

The software that has been used to collect data from the students is called the FACT Media system. This system has two distinct modules namely

1. Fact Tablet Software Application
2. Fact Server Module

The FACT Media system is being developed by a 12-person project team that includes the author of this document

4.2.1 Fact Tablet Software Application. The fact software application is built with the intent of providing students with unconstrained usage as a paper. This would allow students to solve open-ended, exploratory mathematical problems that have been converted to work on this tablet application. These problems are called Classroom Challenges (CC's). These CC's, which were developed by the Mathematics Assessment Project (http://map.mathshell.org/materials/lessons.php) and have been used by hundreds of teachers over several years, are problems that are rich and complex in nature and facilitates learning in the context of that particular problem. This software application enables students to solve these CC's on their tablets and also facilitates teachers to annotate them and return it back to the students for further follow up.

4.2.1.1 Split Screen Setup. The software application has a split screen setup in order to reduce cognitive load on the students. The application's left side is the "Read Only" screen which would allow students to load their previously solved assignments or any reference material that is required to solve the current problem. The application's right side it the "Writable" screen. This one allows students to perform all types of operations detailed in section 4.3

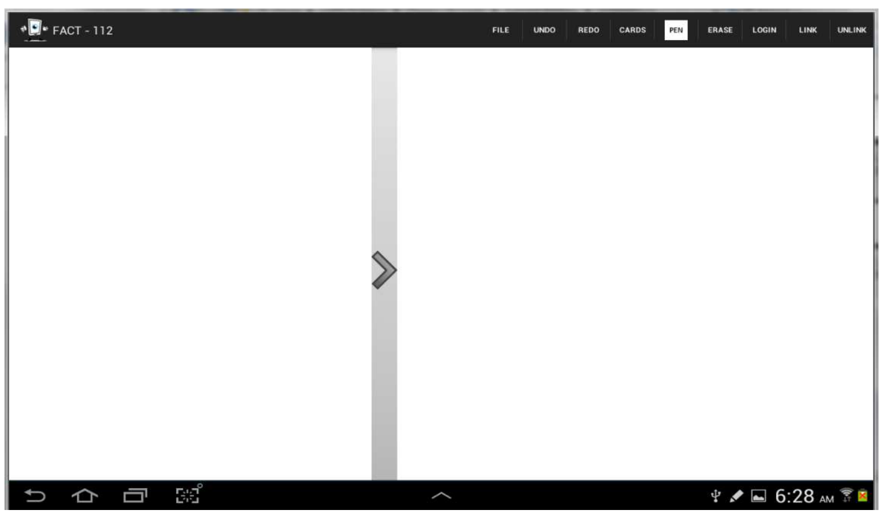


Fig 1. Split Screen Setup Illustration

4.2.1.2 Modes of Operation. There are two major modes of operation that Fact tablet software system can handle. They are the "Solo" mode and the "Group" mode. In "Solo" mode students solve the problem individually without discussing it with their peers and hands over the solution to the teacher. In this mode the actions performed by the student are not transferred to the other students and the worksheet is not shared across various participants. In "group" mode students work together as a group, usually composed of two members, to solve the problem. In this mode, the worksheet is shared by all members of a group. Whenever an action is performed by a participant it is reflected in the all other member's tablets as well.

The teacher can form a set of students into groups and can inform the students to start the group activity. However, student can choose when to enter the "group" mode so that they have their own independence in solving the problem and this would facilitate the student to learn at their own pace.

4.2.2 Components of Tablet software Application. The following are the various components of the tablet software application. These are manipulated by the students in order to facilitate problem solving on tablets.

4.2.2.1 Worksheet and View Port. The application software has a large canvas called the "Work sheet". The worksheet is analogous to a poster that is provided in typical classroom setting. Using their fingers, students can perform traditional scrolling and zooming operations on the worksheet. Zoom operations changes the scale of the worksheet while scroll operation moves the workspace within the window.

At any point in time, the part of the screen that is viewed by the student is called the view port. If the person zooms out considerably then he could see the entire part of the worksheet. However, it is typical that the student only views part of the worksheet while he is solving a mathematical problem.

4.2.2.2 Fact Cards. The application software has small rectangular cards called the "Fact Cards". The cards are analogous to the 2" * 2" index cards that middle school children use. However, Fact cards are size adjustable. The size of the card can be changed using handles located on all four corners of the screen.

When the student drags one of these handles, the size of the card would increase on that side. In addition, these cards also have a control handle on right hand corner of the screen by which the student can resize the card by keeping the aspect ratio of the card unchanged.

Each card also has a set of four controls that provides some extra functionality. The text button (”T”) allows the user to enter text as an input in addition to the available pen/stylus input. The color palette icon allows users to change the color of the card. The cross mark on the corner of the screen is the delete button. One can use it to delete the card.

Fig 2. Fact Card Along with Its Various Controls

4.2.2.3 Fact Image Card. The Fact Image card is capable of holding a graphic image on top of it. This graphic Image can be either a JPEG or SVG or PNG format. These types of cards are capable of performing all the functions of the Fact Card as described above.

4.2.2.4 Fact Graph Card. The Fact Graph card is a card that has a graph grid attached to it. When the sides of the card is modified, the shape of the graph grid is scaled accordingly. This facilitates students drawing simple graphs on the surface of the grid.

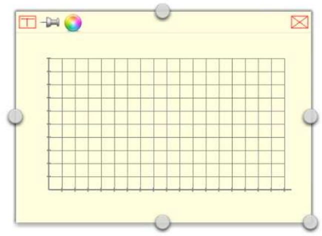


Fig 3. Graph Card

4.2.2.5 Fact Table Card. The Fact table card is card that has a simple table in it. When the sides of the card is modified, the number of rows and columns that are present in the table gets changed dynamically. This would allow students to control the table grid without the need to draw additional lines.

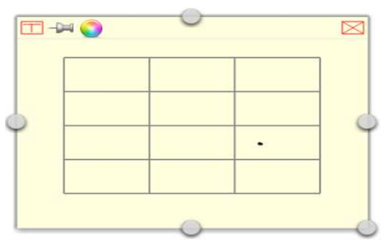


Fig 4. Table Card

4.2.2.6 Strokes. As the sylus moves across a card or the Worksheet, digital ink is deposited, and the internal description is called a stroke. Each stroke is a collection of (x, y) points and are rendered on the screen by connecting all the points with lines. Since these points are captured at a millisecond level a very smooth line is rendered to the user. Although some tablet applications use fingers to lay down digital ink, only the stylus does so in Fact.

4.3 ACTIONS AND EVENTS

Each action that is performed by the student is translated into an event and is sent to the server. The server stores them as a log file. The events present inside a particular command contain relevant information required to update the states of other tablets and also facilitates undo/redo operations on the tablet which initiated the action. Listed below is the comprehensive list of actions that the student can perform on the tablet. These are the raw data from which log features must be extracted.

4.3.1 Scroll and Zoom Action. This action modifies the view port of the screen. Since both the scroll and the zoom is performed using two fingers, these two events are not differentiated and is treated the same for logging and event generation purposes. The command which instantiates this event is called “Scroll Zoom Command”. Event parameters for this action include the four coordinates of the view port namely screen.top, screen.bottom, screen.left and screen.right. It also contains the time stamp denoting the start (start.time) and the end (end.time) of the event.

4.3.2 Card Move Action. This action modifies the position of the card on the screen. The command which instantiates this event is called “Card Move Command". Event parameters for this action include the new position of the card (card.topX, card.topY, card.botX, card.botY), amount of displacement (card.dx ,card.dy) that has been done on the card and the card Id (card.id), which uniquely identifies the card in the application. It also contains the time stamp denoting the start (start.time) and the end (end.time) of the event.

4.3.3 Card Text Add Action. This action modifies the contents of the text on the card. The command which instantiates this event is called “Card Text Command”. Event parameters for this action include the card’s unique id (card.id) and previous text content (prev.text) and the current text content (current.text). It also contains the time stamp denoting the start and the end of the event.

4.3.4 Card Resize Action. This action modifies the size of the card. The command which instantiates this event is called "Card Resize Command". Event parameters for this action include card's unique id (card.id), the side of the card that was changed (card.side) and amount of distance it was moved measured in pixels (card.dx, card.dy). The size of the card is modified according the values of dx and dy. It also contains the time stamp denoting the start (start.time) and the end(end.time) of the event.

4.3.5 Card Add Action. This action adds a card to the application. The command which instantiates this event is called the "Card Move Command". Event parameters for this action include card's unique id (card.id) that got generated and the scale factor (app.scale.factor). It also contains the time stamp denoting the start (start.time) and the end (end.time) of the event.

4.3.6 Stroke Based Actions. There are three different stroke based events. Each stroke event signifies where stoke has to be rendered and a business logic which dictates how this has to be rendered to the screen. The following describes these events.

4.3.6.1 Canvas Drawing Action. This action renders a stroke on the worksheet. The command which instantiates this event is called "Canvas Drawing Command". Event parameters for this action include strokes' unique id (canvas.stroke.id) and the collection of points that got generated. It also contains the time stamp denoting the start (start.time) and the end (end.time) of the event.

4.3.6.2 Card Drawing Action. This action renders a stroke on the fact card. It has to be noted that this card should not contain any graphic associated with the card. The command which instantiates this event is called "Card Drawing Command". Event parameters for this action include card id (card.id) to which this stroke was added, the stroke id (card.stroke.id) and the collection of points that got generated. These points are associated with the card. It also contains the time stamp denoting the start (start.time) and the end (end.time) of the event.

4.3.6.3 Image Drawing Action. This action renders a stroke on the fact image card. It has to be noted that this card should contain a graphic associated with the card. The command which instantiates this event is called "Image Drawing Command". Event parameters for this action include card id to which this stroke was added, the stroke id and the collection of points that got generated. These points are associated with the card. It also contains the time stamp denoting the start and the end of the event.

4.4 FACT SERVER APPLICATION

The Fact Server is primarily responsible for updating its representation of the current state of every tablet, for transferring data across tablets and for recording log files. In addition, it maintains student group preferences that are set by the teacher and is responsible for synchronization of time across various tablet devices.

Whenever an event is generated by the tablet application, it is transferred to the server along with its user information.The server updates its representation of that tablet's state and sends updates to the other tablets of the same group, if any. When a tablet receives these updates, it modifies its current state and redisplays it. Since these updates are sent continuously throughout the problem solving session, the actions of one student are instantaneously reflected in the other tablets of his group members.

4.5 DOMAIN DESCRIPTION

The mathematical problem that is provided to subjects for this thesis study is the distance time interpretation problem. This section briefly describes the goals and problem statement from the Mathematics Assessment Resource Services (MARS) lesson plan. It further describes about how layout has been made in the tablet and provides an illustrative example to explain the domain

4.5.1 Goal. The goal of the problem is to "Interpret distance time graphs abstractly and quantitatively". Students must be able to interpret slopes of these graphs and should be able to make arguments about their hypothesis and should be able to critically reason out the arguments made by their peers.

4.5.2 Cards and Layout. The distance time interpretation problem is a card matching problem. The layout of the problem is a table consisting of 9*3 grid in which cards should be placed. There are three types of cards that are present in the distance time problem. Those are described below.

a. A graph card: A graph card has a graph with X and Y axis. The X axis has variations of time while the Y axis denotes the total amount of distance from home.

b. A table Card: A table card has small n*2 grid, where n denotes the total number of rows that is present in the table. The columns have X (time) and Y (distance) values which corresponds to the variation of the values in the graph card. The distance and time values of the card are pre-filled with values that corresponds to the variation expressed in the problem.

c. A Story Card: A Story card has a concise descriptive story which could potentially be matched to one of the graph cards. The story generally depicts a person moving away from home, who varies his speed along with time and eventually returning back to where he has started.

4.5.3 Correct Matching Triplet. A correct matching triplet is one in which the row of the problem grid contains a triplet (G,T,S) such that graph(G) captures the variations present in the story(S) and the tablet (T) captures the variations that is present in the graph(G) and story (S).

4.5.4 Descriptive Example. Let us consider a simple example to illustrate this. Consider the graph show in fig 4.5.1. The graph has four inflection points namely A, B, C, D.

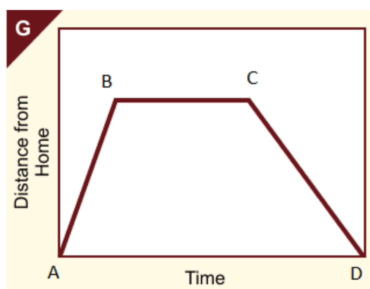


Fig 5. Graph Depicting the Distance From Home

The student should be able to infer the following from the graph at these points

- Point A: Tom (the person) started moving away from home.
- Line segment A-B: Tom moved at a faster pace. Hence the graph depicts a steep increase in slope
- Point B: Tom stopped.
- Line segment B-C: Tom waited in the same location.
- Point C: Tom started moving again.

- Line Segment C-D: Tom returned home at a slower pace than in segment A-B.
- Point D: Tom reached his home.

4.5.5 The Story (S). The story that is to be matched should essentially capture all these inflection points given in the graph. The following is the story that perfectly matches the above graph for this situation.

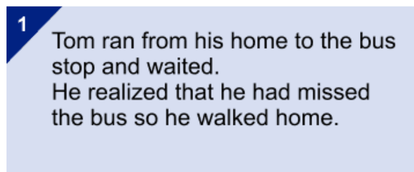
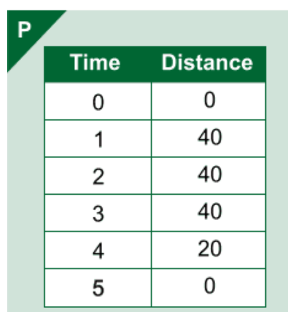


Fig 6. A Story card describing the problem

4.5.6 The Table (T). The table that is to be matched should essentially capture the variation that is present in the graph. The following is a table that perfectly matches the above graph and story


P

Time	Distance
0	0
1	40
2	40
3	40
4	20
5	0

Fig 7. Table Card Showing Variations of the Problem.

4.6 STUDY SETUP

This section presents the study setup required to record various actions made by the subjects when they solve the distance time problem together.

In addition it also explains how the study was conducted and various types of tasks that subjects were required to perform during their study.

4.6.1 Study Procedure. A total of 28 participants participated in the study. The participants in the study were a mix of undergraduate and graduate students of diverse backgrounds from Arizona state university. The participants had a basic understanding of algebra and geometry. No prior mathematical training was provided to them before taking up these sessions. Students were compensated at $10 per hour for their time. The study was conducted in a lab setting. The experimenter served as the teacher for each study session. However, no interruptions from the teacher side were performed. The teacher/experimenter was only responsible to change the tablets configuration from "solo" mode to "group" mode appropriately.

The details of each session are summarized in Table 1. There were no strict time deadlines enforced for this study. However, the overall session lasted an average of 60-110 minutes. Initially participants were briefed about the activities that they are expected to perform during this study and a voluntary consent was obtained before the start of the study. A pretest was provided to gauge their mathematical ability before the actual session. On completion of the pretest, students were given a discovery sheet which contained the set of all the user interface features that students can perform on the FACT Media application. The tablets were then put in "Solo" mode by the teacher/experimenter. Subjects were given unlimited time to explore the application. The average duration for this discovery took about 7-10 minutes. Any doubts (if any) regarding the application were clarified by the end of this session by the experimenter.

The problem solving session started with the experimenter briefly describing the distance time interpretation problem to the student. Then teacher/experimenter put the tablets in “Group” mode so that both students shared the same worksheet. Then subjects begin to solve the distance time problem. During the problem solving session, they discussed with each other and solved the problem together and/or the worked individually. They worked until the entire grid is filled up with cards. There were no restrictions made on how they could work on solving the problem. The overall duration of this phase was between 30-40 minutes.

If the students completed the distance time interpretation problems much earlier than the anticipated time, then they were given additional problems to work with. However, it has to be noted that solving the distance time problem took considerably more time when compared to other problems.

Once the problem solving session was complete, they were given a posttest which was identical to that of the pretest. Usually subjects completed their posttests much faster than the pretest as they were both identical. Then a post survey was conducted after the post test was administered.

Step	**Description**	**Time Duration**
Step 1	Session briefing	5 minutes
Step 2	Pretest	10 - 20 minutes
Step 3	Application Discovery	7 - 10 minutes
Step 4	Session I - Distance time Interpretation	30 - 40 minutes
Step 5	Session II	15 - 20 minutes
Step 6	Post test	5- 10 minutes
Step 7	Post Survey	5 minutes

Table 1. Study Procedure

4.6.2 Recording procedure. There were four different types of input streams that were recorded from the collaborative session. They were

1. Audio stream of the collaborative session. It contained speech of both of the participants. Unidirectional microphones were used to capture this audio signal.
2. Tablet screen contents from each of these participants were streamed to a desktop computer using a HDMI Cable. The video stream was captured at full HD resolution.
3. Each action that subjects performed on the tablets was captured in a log file. This log file has a XML format that is machine readable. The log file generated by these tablets could be found in APPENDIX B.
4. A video of each participant's face was also recorded using two separate Logitech web cameras. The facial videos were streamed to the desktop computer.

The desktop computer showed all 4 videos on its screen as the session was being conducted. Screen capture software was used to save all the screen as one single input stream synched to the audio inputs. The log files were later synchronized to this composite video/audio stream using Elan.

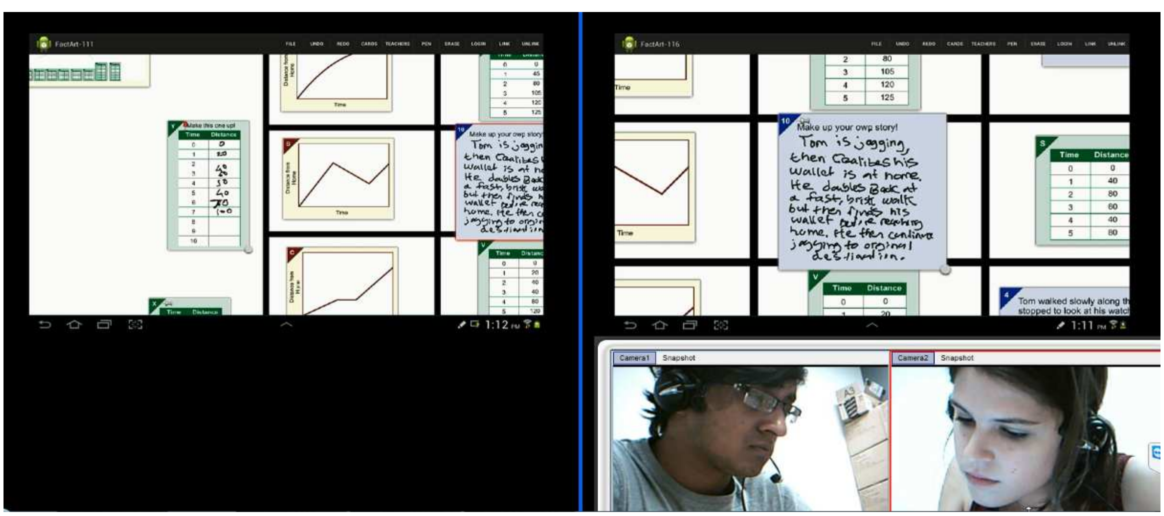


Fig 8. Screenshot of the video during collaboration experiment

CHAPTER 5

METHODOLOGY

This chapter presents the method and the tools that were used to predict group dynamics, when two students are working with each other to solve a mathematical problem. The methods involves the following steps.

I. **Preprocessing:** The log files that were collected from each tablet were merged into a single synchronized stream for the further analysis. Various sub steps were required to achieve this step. This process is detailed in section 5.1. This stream was then fed into R (statistical software) for machine processing and to Elan (video annotation software) for human annotation.

II. **Segmentation:** After the preprocessing step was performed, the data were segmented, that is, partitioned into consecutive segments. Traditional segmentation involves marking segment boundaries by hand. This process is generally tedious and requires lot of effort in order to prepare the data set. For the current problem set, we automated this process. Here the segment time boundaries were algorithmically calculated in the R environment. The details are described in section 5.2.

III. **Human Annotation:** After the segmentation was performed, the boundaries were fed in back to the Elan document. A human coder annotated each segment with collaboration codes by inspecting the audio, video and the facial reaction from the participants. This coding manual is provided in APPENDIX B.

IV. **Feature Extraction:** After human annotation was performed, features were extracted from the log and audio files. Features from the log file are extracted using hand written feature detectors. Then audio features are extracted using hand coded features from Sound-Silence profile extracted from Audacity as well as low level features obtained from OpenSMILE toolkit. The details of feature extraction are provided in section 5.4

V. **Machine Learning Algorithms:** After feature extraction was performed, the subset of features that best predicted the outcome variable were extracted and provided to standard induction algorithms including decision trees, random forests and boosting in order to develop a classifier. The concise description of the induction method is detailed in section 5.5

5.1 TOOLS USED

- **Elan:** Elan is a tool for creation of complex annotations on audio and video data. Each annotation can be a single word or a sentence and could describe any phenomenon. A tier is a set of annotations that are grouped together and is aligned with time. There could be multiple tiers each specifying a particular type of information. An extensive guide on how to use ELAN could be found at (P Wittenburg, 2012)

- **R System:** R system is widely used for statistical computation and for generation of exploratory graphs. It has many idioms and shortcuts for doing complex computational steps in a succinct way.

- **SoX System:** Sound eXchange is a command line tool that provides a number of functions to process audio data. In this project, SoX is used to for segmentation of audio data.

5.2 PREPROCESSING

Preprocessing was done separately for audio and log data. This process is explained in detail in the following sections.

5.2.1 Log Preprocessing: The data that were collected from log files were in the form of XML (Extensible Markup Language). The entire structure of the log file is shown in APPENDIX B. It contains various commands and its respective event parameters in a structured form. The goal of preprocessing is to synchronize the log files to the unified video file and to extract various event parameters from the log file into the R environment for further processing.

Step 1: Extraction of data from Log file: There are 8 actions described in section 4.3. Each of these actions has its respective command and event parameters. An extract from the log file showing the **Card Move Command** is shown below.


```
<replayList              class="edu.asu.fact.events.CardMoveCommand"
cardId="1">
    <endTime>1398729958840</endTime>
    <startTime>1398729957610</startTime>
    <undoCommand>false</undoCommand>
    <botX>658</botX>
    <botY>670</botY>
    <topX>97</topX>
    <topY>69</topY>
    <dx>838</dx>
    <dy>-264</dy>
  </replayList>
```


This command object contains the start and end time stamps of the event along with the end position of the card and the amount of displacement made to the card. These data are extracted programmatically via Java Code.

Step 2: Extraction of Sync time: When the study started, the experimenter marked the start of the experiment by clicking on "Time Sync" menu item present in the users' tablet. The current time of the tablet was marked as a start of the experiment and saved in the logs. This time stamp was programmatically extracted from the log file.

Step 3: Generation of Sync tier and Manual marking using Elan: A new tier was inserted programmatically into Elan. This tier is called the sync tier. A manual inspection of the video was made and the annotator marked the exact position in the time window where the experimenter clicked on the "Time Sync" menu option. This position in time was marked as the total delay from the start of the experiment (exp Delay).

Step 4: Generation of student action tiers: This Elan document was then fed into a Java program which took in the log files of each user as one of its parameters. It retrieved the start of the experiment from the log file and the start of the experiment (exp Delay) from the Elan document and then recalculated each event's start and end time stamp from the start of the experiment. It then fed them into Elan File. The format of Elan File is provided in APPENDIX C with detailed explanation of how this insertion was made.

Step 5: Automatic annotation of student action labels: Each student action was labeled with the information read from the log file. For instance, the "Card Move Command" shown above was transformed into a single string like "CardMoveCommand|1||69|97|658|670|838|-364|" where all the event parameters are appended together with a pipe symbol. This datum was later used by the R script to extract data and its relevant timestamps. Annotations were automatically created with these values and the modified version of the timestamps calculated from step 4.

Step 6: Generation and Visual Inspection: The modified Elan file was then opened using the Elan viewer in order to make sure proper alignment was made. If the alignment was not proper, Steps 2-5 were repeated once again. If proper alignment could not be obtained, then the time stamp present in the log file may have been wrong or the file was corrupt. In this case step 7 is followed.

Step 7: Error Recovery of Logs: This process was a bit tedious and required manual work and careful inspection. The plan was to get a particular event from the log file and find the start of the same event in the video file. Then step 2 was carried out and the start of the event was marked. The time stamp of the start of the event was then used instead of the menu item click. After this, steps 3-6 were performed until proper alignment of the video and the log tiers are made.

Step 8: Mark Study duration: Once step 6 was complete, the synced Elan file was opened in the Elan viewer. A new tier called "duration" was created. In the duration tier, an annotation was marked from start to the end of the distance time interpretation problem. The start of the study was when the experimenter finishes his briefing about the problem and when the students start working on the problem. The end of the study was marked when the students complete their problem solving and calls the experimenter for help.

Step 9: Mark interruption periods: Once step 8 was complete, a new tier called "interruption" was created. In the interruption tier, an annotation was marked whenever a student talked with the experimenter or when there was a problem with the tablet software. These durations were subtracted from the total duration of the study.

Step 10: Importing into R environment: Once Step 9 was complete, the Elan document was fed into the R environment for further processing. Each annotation in the student tiers now contained the synced time stamp, and the text contents of the annotation contained information about various event parameters. These were then processed into a single data frame (analogous to table) using XML library in R.

5.2.1 Audio Preprocessing: The quality of the audio is important for the extraction of features. Hence background noise that is present in the data must be removed. If such noise is present in the data then the following steps are followed

Step 1: Loading the data into Audacity: The audio data were loaded into Audacity which displays them as a waveform. When noise was present, it was indicated by a constant linear band above the base of the wave form. This was manually confirmed by playing the sound segment using the software

Step 2: Detection of Noise Profile: Once the presence of noise was confirmed, the noise profile was extracted by selecting a region in the speech data that contains only noise and no audio data. This was done by using Effect -> Noise Removal and then by selecting "Get noise profile" button.

Step 3: Removal of Noise: Once the noise was obtained by the above step, the next step is to select the whole audio data and then perform Effect -> Repeat Noise Removal. This step eliminated the noise present in the data.

Step 4: Verification of Sound data: Once noise removal was performed, the entire audio set was played in order to check the audio quality. If the quality was not satisfactory, steps 1 to 3 were carried out until satisfactory audio quality was obtained.

5.3 SEGMENTATION

Once the preprocessing steps were complete, the data in the R environment contained all the events that were present in the log file. The card events were extracted in order to perform segmentation.

5.3.1 Goal of segmentation. The goal of the segmentation process was to divide the overall session into separate sub problems. A solved sub problem is one in which a particular card (story or table) was matched with its graph and when the student moved on to the next card. So segment boundaries were placed wherever a sub problem is solved.

A segment boundary was placed whenever a student moved a card into a cell of the table, and someone started to move to a different card. So if there were many consecutive card move events of the same card, they were included inside the same segment.

5.3.2 Details of the segmentation process. Careful segmentation is required in order to capture co-occurrence or individual occurrence of these events. When two students work together on a distance time problem, the events can either overlap with each other or they are separate non overlapping events. These scenarios are explained briefly.

5.3.3 Non Overlapping Move Events. When two events don't overlap with each other, the segment boundary is marked from the end of one card move event to the end of another card movement. These card movements could either be done by the same person or they could be done by two different persons as long as the events involve different cards. Figure 9 depicts how segments are created when two card movements don't overlap with each other. For the sake of brevity, only card move events are depicted in the picture.

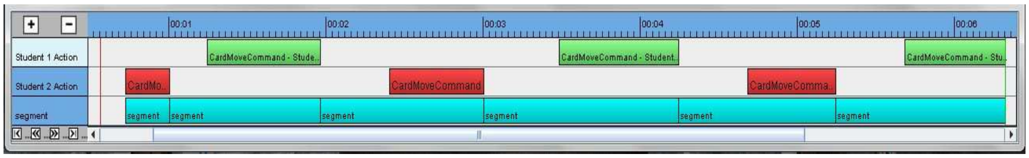


Fig 9. Student Action Tiers for Non Overlapping Events

There are three tiers that are marked to depict this segmentation phenomenon. Student 1 Action tier and Student 2 action tier captures the card move events of student1 and student 2 respectively. Since the events are disjoint, each segment captures the entire problem solving that has happened from one card move to other card move.

5.3.4 Overlapping Move Events. When two events overlap with each other, the segment boundary is marked from end of one card move event to the end of another card movement. Each segment captures some percentage of overlapping activity. Figure 10 depicts how segments are created when two card movements overlap with each other. For the sake of brevity, only card move events are depicted in the picture.

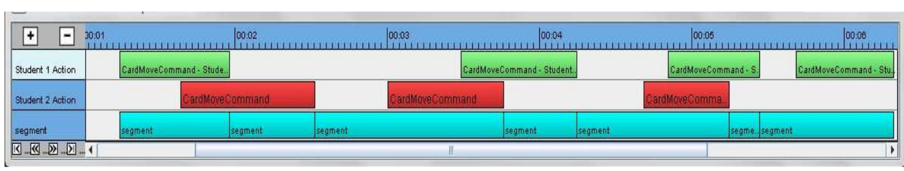


Fig 10. Student Action Tiers for Overlapping Events

There are three tiers visible in Figure 10. Student 1 Action tier and Student 2 action tier captures the card move events of student 1 and student 2 respectively. Since the events co-occur, each segment captures partial to complete activity of one person and partial activity of the other.

The process described above was implemented with R. The output of the segmentation process was fed into the Elan file in order for use by the human coder.

5.4 PROCESS LABELLING

Once the segmentation was performed, the next phase of the process was labeling the segments for their group dynamic type. A human coder visually inspected the actions and conversations exchanged between the subjects and labeled each segment as cooperative, asymmetric contribution or collaboration. A short summarized version of the coding scheme is given below.

5.4.1 Cooperation. The interaction between a dyad was considered co-operation, when subjects have different immediate goals. In other words, they tried to solve different sub problems (card placements) during the segment. Typically, there was little or no conversation between the pair. Mostly the students worked on their own. In some cases, one student idly chattered about the problem they were trying to solve and the other student did not respond back. Since students share different goals, this does not have any characteristic attributes of joint problem solving.

5.4.2 Asymmetric Contribution The interaction between a dyad was considered asymmetric contribution when the subjects shared the same goal (card placement) but one student did most of the work. That is, one person led the conversation and the other person added at most a few reasoning statements. This type of group dynamics is referred to as asymmetric contribution. This definition of asymmetric contribution is quite liberal in how co-operation is defined in literature.

It shares few characteristics of joint problem solving sessions such as common ground, establishing a shared conception of a problem but does not have other properties such as transactivity, scaffolding contributions or argumentative co-construction.

5.4.3 Collaboration. The interaction between the dyad was considered collaboration, when subjects shared the same goal (card placement) and one person often built on other person's reasoning or they engaged in a meaningful knowledge construction process. In few other situations, students engaged in argumentative co-construction process by which they resolved their issues and converged on an understanding of the mathematical content. This definition of collaboration includes various characteristics or attributes of joint problem solving such as common ground, knowledge convergence, co-construction, transactivity, scaffolding contributions and making a shared conception of the problem.

5.5 FEATURE EXTRACTION

The goal of feature extraction was to obtain features that could possibly differentiate between the three types of group dynamic. In order to perform classification, features were extracted from both audio as well as from log files; video data were ignored. This section briefly explains how hand coding was done for log extraction and how OpenSMILE toolkit was used to extract features from audio files.

5.5.1 Feature extraction from Logs. Features extracted from log files primarily described how students worked with each other at every segmented interval. These features were extracted using R. There were two different types of features that were been extracted from the log files. They are

1) Within segment features

2) Across segment features

5.5.2 Within segment features. Within segment features described the events that happened after the beginning of the segment and before the end of the segment. These captured the recent behavior of the student. The features are listed below.

1. **Scroll - Zoom Features:**

Whenever students collaborate with each other, they tended to look at the same card when they solved a sub-problem. In order to perform this action, one student decided on which sub problem to solve and prompted the other person to look at the same sub problem. Typically, the first person had his screen stable and the other person scrolled his screen until he too visualized the same part of the screen. For this process to be captured as features, each zoom event was categorized as "read" or "search" depending on the total amount of time elapsed between each scroll zoom command. For each of these features, the following seven values are calculated

1) Number of events in the segment
2) Mean of the values
3) Median of the values
4) Maximum of values
5) Minimum of values
6) Inter-Quartile Range of the values
7) Standard deviation of the values.

The following are the set of features that are captured from the scroll segments.

Time duration related features:

1) Pause duration between read scroll events of each student.
2) Duration between read scroll events of each student.

3) Total duration (pause + duration) between read scroll events of each student.
4) Pause duration between search scroll events of each student.
5) Duration between search scroll events of each student.
6) Total duration (pause+ duration) between search scroll events of each student.
7) Pause duration between read scroll events of both students.
8) Duration between read scroll events of both students.
9) Total duration (pause+ duration) between read scroll events of both students.
10) Pause duration between search scroll events of both students.
11) Duration between search scroll events of both students.
12) Total duration (pause+ duration) between search scroll events of both students.

Time Overlap Related Features:

1) Time overlap between read and scrolls events of both students.
2) Sequence of consecutive overlap between read scroll events
3) Sequence of non – consecutive overlap between read scroll events
4) Time overlap between search scroll event of both students.
5) Sequence of consecutive overlap between search scroll events.
6) Sequence of non – consecutive overlap between search scroll events.
7) Time overlap between read and scroll of both students.
8) Sequence of consecutive overlap between read and scroll events.
9) Sequence of non – consecutive overlap between read and scroll events.
10) Time overlap between read scroll event of one student and search scroll event of other students.

11) Sequence of consecutive overlap between read scroll event of one student and search scroll event of other students.

12) Sequence of non – consecutive overlap read scroll event of one student and search scroll event of other students.

View Overlap Related Events:

1) Total distance between centers of view port when there is an overlap in time interval between read scroll events of both students.
2) Total area of overlap between view ports when there is an overlap in time interval between read scroll events of both students.
3) Total distance between centers of view port when there is an overlap in time interval between search scroll events of both students.
4) Total area of overlap between view ports when there is an overlap in time interval between search scroll events of both students.
5) Total distance between centers of view port when there is an overlap in time interval between read and scroll events of both students.
6) Total area of overlap between view ports when there is an overlap in time interval between read and scroll of both student events.
7) Total distance between centers of view port when there is an overlap in time interval between read scroll event of one student and search scroll event of other student.
8) Total area of overlap between view ports when there is an overlap in time interval between read scroll event of one student and search scroll event of other student.

2. **Card Related features**

    1. Total amount of time both students looked at the same set of cards.

2. Total amount of time both students looked at the same card that was moved to the solution grid.
3. Total number of cards viewed by that student during each read operation.
4. Total number of cards visualized by that student during a scroll operation.
5. Total number of card visualized by both students during each read operation.
6. Total amount of time it took to move a card to the solution grid.
7. Overlap between two different card move events.
8. Total duration of pause between two card movements by a single person.
9. Total duration of pause between two card movements contributed by both students.
10. Total duration of each segment.
11. The person who moved the card to a position in the solution grid later changes to another positing after some talk.
12. Time they took to write on the card.

5.5.3 Across segment features. Across segment features describes the nature of events that happened up to the beginning of the current segment. These features capture the past behavior of the student. Cumulative sum of each of the features listed above were captured as well. In addition four new features were added to the list.

1. The same card was moved to a different place by the same student.
2. The same card was moved to a different place by different student.
3. Total number of sub problems solved until that point in time.

4. The type of cards student work on during consecutive move operations

5.5.4 Features extracted from audio: The audio file was segmented according to the segment boundaries obtained from the Elan file. There was a two-step process involved in extraction of audio features.

Step 1: Extraction of features from Speech – Silence Profile: The entire audio stream for each session was fed into Audacity and its sound finder feature (Analyze -> Sound Finder) was used to extract the durations of segments where a sound was present. The audio level below -26 dB were treated as silence and the minimum duration of silence between two sounds was considered to be one second. The interval between two sound segments were taken considered as silences. Each of these sound and silence data was further segmented into various smaller chunks based on the segment boundaries obtained from the log files. Then the following features were extracted from speech silence profile

1. Speech Time: Total amount of duration when there was a presence of talk.
2. Silence Time: Total amount of duration when there was no talk.
3. Mean, Median, standard deviation of speech time.
4. Minimum and Maximum duration of speech time.
5. Mean, Median, standard deviation of silence time.
6. Minimum and Maximum duration of silence time.

Step 2: Extraction of Low level features from Speech data: The following low level audio features are extracted from the speech data. The total amount of features provided by OpenSMILE toolkit for each of the categories is given in parenthesis.

1. Features extracted from Mel-frequency Cepstrum coefficients (630).

2. Features extracted from Mel-frequency band (336).
3. Features extracted from linear spectral coefficient (336).
4. Features extracted from loudness (42).
5. Features extracted from voicing (42).
6. Features extracted from fundamental frequency envelope.(38)
7. Features extracted from pitch.(38)
8. Features extracted from jitter(DP).(38)
9. Features extracted from shimmer. (38)
10. Features extracted from pitch onsets.(38)

5.6 MACHINE LEARNING ALGORITHMS

The overall goal of the thesis is to predict the output variables from the input features provided to the system. In order to perform machine learning, the following methods were employed.

Step 1: Centering and Scaling of Variables: The first step was to normalize (i.e., center and scale; z-transform) all the features that were present in our dataset. This step was performed so that all the variables had a mean of zero and a standard deviation of 1. This step was performed so that units of the regression coefficient would be the same across all variables.

Step 2: Application of Tree based learning methods: Random Forests performed the best when compared to other ensemble methods such as bagging and boosting. Each of these approaches involved building multiple trees which were then combined to give a single prediction.

5.6.1 Bagging. In order to increase the prediction accuracy of the model, bootstrap sampling was done on the training dataset and the models were built based on these input samples. Then the average of these models were used to make predictions.

5.6.2 Random Forests. Random forests grows many single classification trees based on bootstrapping samples. When an input sample has to be predicted, each tree gives a class and majority voting is performed in order to select the best class. Random forests are a special case of bagging.

5.6.3 Boosting. The boosting process is similar to bagging except that the trees are grown sequentially. In addition, boosting does not involve bootstrapping and each tree is fit on the modified version of the original dataset.

CHAPTER 6

RESULTS AND CONCLUSION

This section presents a few observations about how the data were collected and various distribution of labels present in the data set. It presents a brief summary of how labels were merged together to create different groups and presents the reliability metrics of the induced classifier.

Nature of data: As expected, we had some technical difficulties with our FACT software system as it was in its initial stages of development. Also, we had some problems with our audio setup, which rendered most of the observations unusable for our machine learning classifiers. Hence, the total amount of dataset and the number of samples were significantly less than what had been estimated during the initial course of study plan.

Quality of Audio data: The audio data were collected using the input merged from two different microphones. The final output obtained had a single stream with both of these input mixed together. In addition, the data had nearly equal distribution of native and non-native English speaker sessions. Table 2 indicates the distribution of sessions between native and non-native English speakers.

Type of Speakers	Number of sessions
Native speakers	6
Non Native speakers	8

Table 2: Audio Data Distribution

Data Annotation: Human annotators performed annotation on four different levels: collaboration(C), low asymmetric collaboration (A-1), high asymmetric collaboration (A-2) and cooperation (P). The operational definitions of the same are provided in the section 5.4

Observations of data: As mentioned earlier, every segment was judged by a human annotator as being an instance of collaboration, asymmetric contribution or cooperation. The category of Asymmetric contribution was future divided into High or Low based on the amount of dialogue between the participants. Table 3 shows the distribution of codes over the whole dataset.

Description	Number of segments
Collaboration	94
Asymmetric Contribution (High)	50
Asymmetric Contribution (Low)	63
Cooperation	118
Total samples	327

Table 3: Distribution of Segment Labels across the Whole Dataset

6.1 STUDY RESULTS

The overall goal of the study is to induce a classifier which can detect collaboration. However, there is some ambiguity about how to treat the asymmetric contribution category, so three levels of granularity were defined for creating and evaluating detectors.

1) Can a binary classifier reliably discriminate between the Cooperative category and all the others? That is, when collaboration is given a liberal definition that includes the asymmetric contribution, can it be discriminated from Cooperation?
2) Can a ternary classifier reliably discriminate between Collaboration, Asymmetric contribution (both High and Low lumped together) and Cooperation?

3) Can a four-way classifier reliably discriminate between all four labels?

The next few subsections report the results for each of the three binary classifications from features extracted from log and audio data. Finally, features extracted from log and audio data were combined together and a classifier was induced.

6.2 RESULTS FROM AUDIO DATA

The following subsection reports the results obtained from classifier by using features from the audio data.

Classification between Cooperative and all other work: The following were the steps that were performed in order to classify collaborative and cooperative work.

Step 1: For the above question to be answered, the data has to be divided into two groups: A liberal definition of Collaborative (denoted H) vs. Collaborative (P). That is, the labels A-1 & A-2 & C were made into a single group called "H" that denoted a liberal definition of collaborative work.

Step 2: Random Forests, bagging and boosting were applied on the dataset. The Random forest classifier performed better in comparison to bagging and boosting.

Step 3: Tenfold cross validation was performed on the model built above.

The Cohen's kappa for this classification is 0.8096. The confusion matrix and detailed class accuracy per class are reported below. Accuracy of the class classifier was around **91.4%**

Actual Class		Predicted Class
H	P	
199	8	H
20	98	P

Table 4: Confusion Matrix for Classification between Collaboration and Cooperation (audio)

TP Rate	FP Rate	Precision	Recall	F- Measure	ROC Area	Class
0.961	0.169	0.909	0.961	0.934	0.975	H
0.831	0.039	0.925	0.831	0.875	0.975	P

Table 5: Detailed Class Level Accuracy for High and Low Collaboration (Audio)

Interpretation: The audio features were able to classify these two classes with reasonably accuracy and the value of Cohen's Kappa indicates that the agreement between the human annotation and the machine learning annotation is excellent.

Classification between collaboration, asymmetric contribution and collaboration: The following were the steps that were performed in order to classify collaboration and asymmetric contribution

Step 1: For the above question to be answered, the data has to be divided into three groups: Collaboration, Asymmetric Contribution and Cooperation. So, the labels A-1 & A-2 were made into a single group called “A” that denoted Asymmetric collaboration. No modifications were done to cooperative work group (P) and collaborative work group(C).

Step 2: Random Forests, bagging and boosting were applied on the dataset. The Random forest classifier performed better in comparison to bagging and boosting.

Step 3: Tenfold cross validation was performed on the model built above. The Cohen's kappa for this classification is 0.6203. The confusion matrix and detailed class accuracy per class are reported below. The overall accuracy of the class classifier: **75.76%**

Actual Class		**Predicted Class**	
	A	P	C
A	81	7	25
P	17	96	5
C	25	3	66

Table 6: Confusion Matrix for Classification between Collaboration & Asymmetric Contribution (Audio)

TP Rate	**FP Rate**	**Precision**	**Recall**	**F- Measure**	**ROC Area**	**Class**
0.717	0.198	0.659	0.717	0.686	0.837	A
0.814	0.048	0.906	0.814	0.857	0.956	P
0.702	0.13	0.688	0.702	0.695	0.896	C

Table 7: Detailed Class Level Accuracy for Collaboration and Asymmetric Contribution (Audio)

Interpretation: It could be visualized from the confusion matrix that, the number of samples misclassified between Collaboration and Cooperation is low. However, there is a thin line of separation between the Asymmetric Contribution and Collaboration as well as Asymmetric contribution and Co-operation, which the detector fails to recognize. The Cohen's kappa indicates that the agreement between the human annotation and the machine learning annotation is moderate.

Classification between all four labels: The following were the steps that were performed in order to classify low and high asymmetric contribution.

Step 1: For the above question to be answered, the data has to be divided into four groups: Hence no modifications were done to any of the class labels.

Step 2: Random Forests, bagging and boosting were applied on the dataset. The Random forest classifier performed better in comparison to bagging and boosting.

Step 3: Tenfold cross validation was performed on the model built above. The Cohen's kappa for this classification is 0.5782. The confusion matrix and detailed class accuracy per class are reported below. The overall accuracy of the class classifier is 69.84**%**

Actual Class		**Predicted Class**		
	A-2	P	C	A-1
A-2	15	8	19	8
P	5	105	4	4
C	17	5	68	4
A-1	3	8	13	39

Table 8: Confusion Matrix for Four-way Classification (audio)

TP Rate	FP Rate	Precision	Recall	F- Measure	ROC Area	Class
0.3	0.091	0.375	0.3	0.333	0.753	A-2
0.89	0.101	0.833	0.89	0.861	0.96	P
0.723	0.156	0.654	0.723	0.687	0.893	C
0.619	0.061	0.709	0.619	0.661	0.888	A-1

Table 9: Detailed Class Level Accuracy for Symmetric and Asymmetric contribution (audio)

Interpretation: It could be visualized from the confusion matrix that the built model was not able to accurately classify different levels of asymmetric contribution. Out of the four classes, A-2 (High level) of asymmetric collaboration is the label that performed poorly. Most of these class labels were easily confused with Collaboration. The Cohen's kappa indicates that the agreement between the human annotation and the machine learning annotation is low.

6.2 RESULTS FROM LOG DATA

The following subsection reports the results obtained from classifier by using features from the log data.

Classification between Cooperative and all other work: The following were the steps that were performed in order to classify collaborative and cooperative work.

Step 1: For the above question to be answered, the data has to be divided into two groups: A liberal definition of Collaborative (denoted H) vs. Collaborative (P). That is, the labels A-1 & A-2 & C were made into a single group called "H" that denoted a liberal definition of collaborative work.

Step 2: Random Forests, bagging, additive logistic regression, J48 graft and boosting were applied on the dataset. The additive logistic regression performed better in comparison to other models.

Step 3: Tenfold cross validation was performed on the model built above.

The Cohen's kappa for this classification is **0.7396.** The confusion matrix and detailed class accuracy per class are reported below. Accuracy of the class classifier was around **88.03%**

Actual Class		**Predicted Class**
H	P	
190	18	H
21	97	P

Table 10: Confusion Matrix for Classification between Collaboration and Cooperation (Log)

TP Rate	**FP Rate**	**Precision**	**Recall**	**F- Measure**	**ROC Area**	**Class**
0.913	0.178	0.9	0.913	0.907	0.927	H
0.822	0.087	0.843	0.822	0.833	0.927	P

Table 11: Detailed Class Level Accuracy for High and Low Collaboration (Log)

Interpretation: The log features were able to classify these two classes with reasonably accuracy and the value of Cohen's Kappa indicates that the agreement between the human annotation and the machine learning annotation is good.

Classification between collaboration, asymmetric contribution and collaboration: The following were the steps that were performed in order to classify collaboration and asymmetric contribution

Step 1: For the above question to be answered, the data has to be divided into three groups: Collaboration, Asymmetric Contribution and Cooperation. So, the labels A-1 & A-2 were made into a single group called "A" that denoted Asymmetric collaboration. No modifications were done to cooperative work group (P) and collaborative work group(C).

Step 2: Random Forests, bagging, additive logistic regression, J48 graft and boosting were applied on the dataset. The additive logistic regression performed better in comparison to other models.

Step 3: Tenfold cross validation was performed on the model built above. The Cohen's kappa for this classification is 0.583. The confusion matrix and detailed class accuracy per class are reported below. The overall accuracy of the class classifier: **72.3926%**

Actual Class		**Predicted Class**	
	A	P	C
A	78	13	23
P	8	102	8
C	31	7	56

Table 12: Confusion Matrix for Classification between Collaboration & Asymmetric Contribution (Log)

TP Rate	FP Rate	Precision	Recall	F- Measure	ROC Area	Class
0.684	0.184	0.667	0.684	0.675	0.817	A
0.864	0.096	0.836	0.864	0.85	0.938	P
0.596	0.134	0.644	0.596	0.619	0.851	C

Table 13: Detailed Class Level Accuracy for Collaboration and Asymmetric contribution (Log)

Interpretation: It could be visualized from the confusion matrix that, the number of samples misclassified between Collaboration and Cooperation is low. However, there is a thin line of separation between the Asymmetric Contribution and Collaboration as well as Asymmetric contribution and Co-operation, which the detector fails to recognize. The Cohen's kappa indicates that the agreement between the human annotation and the machine learning annotation is moderate.

Classification between all four labels: The following were the steps that were performed in order to classify low and high asymmetric contribution.

Step 1: For the above question to be answered, the data has to be divided into four groups: Hence no modifications were done to any of the class labels.

Step 2: Random Forests, bagging, additive logistic regression, J48 graft and boosting were applied on the dataset. The additive logistic regression performed better in comparison to other models.

Step 3: Tenfold cross validation was performed on the model built above. The Cohen's kappa for this classification is 0.505. The confusion matrix and detailed class accuracy per class are reported below. The overall accuracy of the class classifier: **69.84%**

Actual Class		Predicted Class		
	A-2	P	C	A-1
A-2	16	7	20	8
P	5	100	9	4
C	10	9	60	15
A-1	11	6	12	34

Table 14: Confusion Matrix for Four-way Classification (Log)

TP Rate	FP Rate	Precision	Recall	F- Measure	ROC Area	Class
0.314	0.095	0.381	0.314	0.344	0.678	A-2
0.84	0.106	0.82	0.847	0.833	0.93	P
0.638	0.177	0.594	0.638	0.615	0.844	C
0.54	0.103	0.557	0.54	0.548	0.836	A-1

Table 15: Detailed Class Level Accuracy for Symmetric and Asymmetric Contribution (Log)

Interpretation: It could be visualized from the confusion matrix that the built model was not able to accurately classify different levels of asymmetric contribution. Out of the four classes, A-2 (High level) of asymmetric collaboration is the label that performed poorly. Most of these class labels were easily confused with Collaboration. The Cohen's kappa indicates that the agreement between the human annotation and the machine learning annotation is low.

6.3 RESULTS FROM COMBINED FEATURE SET

The following subsection reports the results obtained from classifier by using features from both audio and log data.

Classification between Cooperative and all other work: The following were the steps that were performed in order to classify collaborative and cooperative work.

Step 1: For the above question to be answered, the data has to be divided into two groups: A liberal definition of Collaborative (denoted H) vs. Collaborative (P). That is, the labels A-1 & A-2 & C were made into a single group called "H" that denoted a liberal definition of collaborative work.

Step 2: Attribute feature selection was performed using Best First Search. It selects the subset of features based on its individual predictive ability along with the degree of redundancy between them.

Step 3: Random Forests, bagging, additive logistic regression, J48 graft and boosting were applied on the dataset. The random forest performed better in comparison to other models.

Step 4: Tenfold cross validation was performed on the model built above.

The Cohen's kappa for this classification is **0.8232.** The confusion matrix and detailed class accuracy per class are reported below. Accuracy of the class classifier was around **92.00%**

Actual Class		Predicted Class
H	P	
200	7	H
19	99	P

Table 16: Confusion Matrix for Classification between Collaboration and Cooperation (Combined)

TP Rate	FP Rate	Precision	Recall	F- Measure	ROC Area	Class
0.966	0.161	0.913	0.966	0.939	0.978	H
0.839	0.034	0.934	0.839	0.884	0.978	P

Table 17: Detailed Class Level Accuracy for High and Low Collaboration (combined)

Interpretation: The log features were able to classify these two classes with reasonably accuracy and the value of Cohen's Kappa indicates that the agreement between the human annotation and the machine learning annotation is excellent.

Classification between collaboration, asymmetric contribution and collaboration: The following were the steps that were performed in order to classify collaboration and asymmetric contribution

Step 1: For the above question to be answered, the data has to be divided into three groups: Collaboration, Asymmetric Contribution and Cooperation. So, the labels A-1 & A-2 were made into a single group called "A" that denoted Asymmetric collaboration. No modifications were done to cooperative work group (P) and collaborative work group(C).

Step 2: Attribute feature selection was performed using Best First Search. It selects the subset of features based on its individual predictive ability along with the degree of redundancy between them.

Step 3: Random Forests, bagging, additive logistic regression, J48 graft and boosting were applied on the dataset. The additive logistic regression performed better in comparison to other models.

Step 4: Tenfold cross validation was performed on the model built above. The Cohen's kappa for this classification is 0.6527. The confusion matrix and detailed class accuracy per class are reported below. The overall accuracy of the class classifier: **76.9231%**

	Actual Class	Predicted Class	
	A	P	C
A	77	10	26
P	11	104	3
C	24	1	69

Table 18: Confusion Matrix for Classification between Collaboration & Asymmetric Contribution (Combined)

TP Rate	**FP Rate**	**Precision**	**Recall**	**F- Measure**	**ROC Area**	**Class**
0.681	0.165	0.688	0.681	0.684	0.846	A
0.881	0.053	0.904	0.881	0.893	0.957	P
0.734	0.126	0.704	0.734	0.719	0.905	C

Table 19: Detailed Class Level Accuracy for Collaboration and Asymmetric Contribution (Combined).

Interpretation: It could be visualized from the confusion matrix that, the number of samples misclassified between Collaboration and Cooperation is low. However, there is a thin line of separation between the Asymmetric Contribution and Collaboration as well as Asymmetric contribution and Co-operation, which the detector fails to recognize. The Cohen's kappa indicates that the agreement between the human annotation and the machine learning annotation is moderate.

Classification between all four labels: The following were the steps that were performed in order to classify low and high asymmetric contribution.

Step 1: For the above question to be answered, the data has to be divided into four groups: Hence no modifications were done to any of the class labels.

Step 2: Attribute feature selection was performed using Best First Search. It selects the subset of features based on its individual predictive ability along with the degree of redundancy between them.

Step 3: Random Forests, bagging, additive logistic regression, J48 graft and boosting were applied on the dataset. The random forest performed better in comparison to other models.

Step 4: Tenfold cross validation was performed on the model built above. The Cohen's kappa for this classification is 0.6002. The confusion matrix and detailed class accuracy per class are reported below. The overall accuracy of the class classifier was **72.00%**

Actual Class	**Predicted Class**			
	A-2	P	C	A-1
A-2	13	13	20	4
P	1	111	6	0
C	7	7	76	4
A-1	3	13	13	34

Table 20: Confusion Matrix for Four-way Classification (Combined)

TP Rate	FP Rate	Precision	Recall	F- Measure	ROC Area	Class
0.26	0.04	0.542	0.26	0.351	0.766	A-2
0.941	0.159	0.771	0.941	0.847	0.963	P
0.809	0.169	0.661	0.809	0.727	0.901	C
0.54	0.031	0.119	0.711	0.72	0.698	A-1

Table 21: Detailed Class Level Accuracy for Symmetric and Asymmetric Contribution (Combined)

Interpretation: It could be visualized from the confusion matrix that the built model was not able to accurately classify different levels of asymmetric contribution. Out of the four classes, A-2 (High level) of asymmetric collaboration is the label that performed poorly. Most of these class labels were easily confused with Collaboration. The Cohen's kappa indicates that the agreement between the human annotation and the machine learning annotation is low.

6.4 COMPARISION OF RESULTS FROM DIFFERENT SOURCES

The below table gives the kappa and accuracy of the classifiers induced using audio, log and combined data sets. It could be seen that the combined feature set along with feature selection performed better than log and audio features considered individually. However it has to be noted that the features from audio and logs were comparable to each other with slight increase in accuracy in terms of the audio features.

	Binary Classifier		Ternary Classifier		Four way Classifier	
Features	**Kappa**	**Accuracy**	**Kappa**	**Accuracy**	**Kappa**	**Accuracy**
Audio	0.8096	91.4	0.6203	75.76	0.5782	69.84
Log	0.7396	88.03	0.583	72.39	0.5050	69.84
Combined	0.8232	92.00	0.6527	76.92	0.6002	72.00

Table 22. Comparison of Kappa and Accuracy across Different Classifiers and Feature Sets

CONCLUSION AND FUTURE WORK

The long term goal of the project is to induce automated measures for the quality of collaboration between student pairs using tablet software. However, there are many theories and definitions of collaborative processes. It is not yet clear which of these definitions and theories are amendable to induction of automated detectors. Moreover, it is also not clear which of these definitions will benefit teachers and students. Thus, the ultimate goal is to find a definition of collaboration that both allows automated measurement and is useful for guiding instruction.

In order to find answers the above questions, a primitive pilot study was conducted with the intent to developing a high level classifier that can differentiate between different levels of collaboration. There were four major levels of collaboration that were defined: Collaboration, Asymmetric contribution (High), Asymmetric contribution (Low) and cooperation.

These levels were defined on an ordinal scale in the order listed above and each level captured subsets of process dimensions defined in the literature section. To be more specific, the collaboration and asymmetric categories share quality attributes such as common ground and shared cognition, while transactivity, argumentative and dialogic co-construction were exhibited only in collaborative level of conversation. On the other hand, cooperation was characterized by division of labor among participants where each person completed some subset of tasks in order to solve the overall problem.

Previous research in the domain of automatic identification of collaborative processes, concentrated more on specific dimensions of collaborative processes. Features that could potentially differentiate between various definitions of collaboration were extracted from actions students' performed on the task domain. A similar process was attempted in this study with a difference that the levels of collaboration defined in this study, encompassed a broad spectrum of definitions. The overall intent of this study is to paint an overall picture to the teacher on how each student performed in the classroom.

Features extracted for card sorting task were obtained from streams of time stamped data extracted from the audio and logs of tablet usage. Almost all of the audio features should be generalizable to all the tasks that students perform in real world classrooms. A partial subset of scroll and log features can be utilized for classification of levels of collaboration depending on the task at hand.

When designing features for a task, care must be taken on how each of such features are modelled. Each feature should characterize where the focus of attention is channeled and how patterns of interaction occur when students solve a particular task.

Although the amount of data available for analysis was limited, the initial results obtained from this classification process were encouraging. Given the initial results, I am planning to collect more data and make an attempt to create a real time collaborative system. In addition, my future directions would include comparison of various definitions and processes that are automatable and those which would be helpful in real-time classrooms.

REFERENCES

Anaya, Antonio R, & Boticario, Jesús G. (2011). Application of machine learning techniques to analyse student interactions and improve the collaboration process. Expert Systems with Applications, 38(2), 1171-1181.

Baker, Eva L, & O'Neil, Harold F. (2002). Measuring problem solving in computer environments: current and future states. Computers in Human Behavior, 18(6), 609-622.

Biggs, Jone B, & Collis, Kevin F. (1982). Evaluating the quality of learning. *New York*.

Bloom, Benjamin Samuel, College, Committee of, & Examiners, University. (1956).*Taxonomy of educational objectives* (Vol. 1): David McKay New York.

Brown, Ann L, & Campione, Joseph C. (1994). Guided discovery in a community of learners.

CHAMRADA, Marian, & MANSOUR, Nasser. (2009). Are your students working creatively together? automatically recognizing creative turns in student e-discussions. *Artificial Intelligence in Education: Building Learning Systems that Care: from Knowledge Representation to Affective Modelling, 200*, 317.

Chi, Michelene TH. (1992). Conceptual change within and across ontological categories: Examples from learning and discovery in science. *Cognitive models of science: Minnesota studies in the philosophy of science*, 129-160.

Chi, Michelene TH, De Leeuw, Nicholas, Chiu, Mei-Hung, & LaVancher, Christian. (1994). Eliciting self-explanations improves understanding. *Cognitive science, 18*(3), 439-477.

Chi, Michelene TH, Roy, Marguerite, & Hausmann, Robert GM. (2008). Observing tutorial dialogues collaboratively: Insights about human tutoring effectiveness from vicarious learning. *Cognitive Science, 32*(2), 301-341.

Chu-Carroll, Jennifer, & Brown, Michael K. (1997). Tracking initiative in collaborative dialogue interactions *Proceedings of the eighth conference on*

European chapter of the Association for Computational Linguistics (pp. 262-270).

Clark, Douglas B, & Sampson, Victor. (2008). Assessing dialogic argumentation in online environments to relate structure, grounds, and conceptual quality. *Journal of Research in Science Teaching, 45*(3), 293-321.

Clark, Herbert H, & Brennan, Susan E. (1991). Grounding in communication. *Perspectives on socially shared cognition, 13*(1991), 127-149.

Clark, Herbert H, & Schaefer, Edward F. (1989). Contributing to discourse. *Cognitive science, 13*(2), 259-294.

Dillenbourg, Pierre. (1999). What do you mean by collaborative learning? *Collaborative-learning: Cognitive and computational approaches.*, 1-19.

Dillenbourg, Pierre, Baker, Michael J, Blaye, Agnes, O'Malley, Claire, & others. (1995). The evolution of research on collaborative learning. *Learning in Humans and Machine: Towards an interdisciplinary learning science.*, 189-211.

Dillenbourg, Pierre, Zufferey, Guillaume, Alavi, Hamed, Jermann, Patrick, Do-Lenh, Son, Bonnard, Quentin, . . . Kaplan, Frédéric. (2011). Classroom orchestration: The third circle of usability. *CSCL2011 Proceedings, 1*, 510-517.

Doise, W. (1990). The development of individual competencies through social interaction. *Children helping children*, 43-64.

Duque, Rafael, & Bravo, Crescencio. (2007). A method to classify collaboration in CSCL systems *Adaptive and Natural Computing Algorithms* (pp. 649-656): Springer.

Garrison, D Randy, Anderson, Terry, & Archer, Walter. (2001). Critical thinking, cognitive presence, and computer conferencing in distance education. *American Journal of distance education, 15*(1), 7-23.

Gibbons, Pauline. (2002). Scaffolding language, scaffolding learning: Teaching second language learners in the mainstream classroom.

Gress, Carmen LZ, Fior, Meghann, Hadwin, Allyson F, & Winne, Philip H. (2010). Measurement and assessment in computer-supported collaborative learning. *Computers in Human Behavior, 26*(5), 806-814.

Gweon, Gahgene. (2012). *Assessment and support of the idea co-construction process that influences collaboration.* Duquesne University.

Gweon, Gahgene, Jain, Mahaveer, McDonough, John, Raj, Bhiksha, & Rosé, Carolyn P. (2013). Measuring prevalence of other-oriented transactive contributions using an automated measure of speech style accommodation. *International Journal of Computer-Supported Collaborative Learning*, 1-21.

Harrer, Andreas, McLaren, Bruce M, Walker, Erin, Bollen, Lars, & Sewall, Jonathan. (2006). Creating cognitive tutors for collaborative learning: steps toward realization. *User Modeling and User-Adapted Interaction, 16*(3-4), 175-209.

Hausmann, RG, Chi, Michelene TH, & Roy, Marguerite. (2004). *Learning from collaborative problem solving: An analysis of three hypothesized mechanisms*. Paper presented at the 26nd annual conference of the Cognitive Science society.

Hmelo-Silver, Cindy E. (2003). Analyzing collaborative knowledge construction: multiple methods for integrated understanding. *Computers & Education, 41*(4), 397-420.

Hogan, Kathleen, Nastasi, Bonnie K, & Pressley, Michael. (1999). Discourse patterns and collaborative scientific reasoning in peer and teacher-guided discussions. *Cognition and instruction, 17*(4), 379-432.

Jain, Mahaveer, McDonough, John, Gweon, Gahgene, Raj, Bhiksha, & Rosé, Carolyn Penstein. (2012). An unsupervised dynamic bayesian network approach to measuring speech style accommodation *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 787-797).

Johnson, David W, & Johnson, Roger T. (1996). Cooperation and the use of technology. *Handbook of research for educational communications and technology: A project of the Association for Educational Communications and Technology*, 1017-1044.

Kinnebrew, John S, & Biswas, Gautam. (2012). *Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution.* Paper presented at the EDM.

Martinez-Maldonado, Roberto, Kay, Judy, & Yacef, Kalina. (2013). *An automatic approach for mining patterns of collaboration around an interactive tabletop*. Paper presented at the Artificial Intelligence in Education.

Martinez, Roberto, Collins, Anthony, Kay, Judy, & Yacef, Kalina. (2011). Who did what? Who said that?: Collaid: an environment for capturing traces of collaborative learning at the tabletop *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (pp. 172-181).

McLaren, Bruce M, Bollen, Lars, Walker, Erin, Harrer, Andreas, & Sewall, Jonathan. (2005). *Cognitive tutoring of collaboration: developmental and empirical steps towards realization*. Paper presented at the Proceedings of th 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!

McLaren, Bruce M, Scheuer, Oliver, De Laat, Maarten, Hever, Rakheli, De Groot, Reuma, & Rosé, Carolyn Penstein. (2007). Using machine learning techniques to analyze and support mediation of student e-discussions. *Frontiers in Artificial Intelligence and Applications, 158*, 331.

McLaren, Bruce M, Scheuer, Oliver, & Mikvsátko, Jan. (2010). Supporting collaborative learning and e-Discussions using artificial intelligence techniques. *International Journal of Artificial Intelligence in Education, 20*(1), 1-46.

Meier, Anne, Spada, Hans, & Rummel, Nikol. (2007). A rating scheme for assessing the quality of computer-supported collaboration processes. *International Journal of Computer-Supported Collaborative Learning, 2*(1), 63-86.

Noroozi, Omid, Biemans, Harm JA, Busstra, Maria C, Mulder, Martin, & Chizari, Mohammad. (2011). Differences in learning processes between successful and less successful students in computer-supported collaborative learning in the field of human nutrition and health. *Computers in Human Behavior, 27*(1), 309-318.

Nussbaum, Miguel, Alvarez, Claudio, McFarlane, Angela, Gomez, Florencia, Claro, Susana, & Radovic, Darinka. (2009). Technology as small group face-to-face Collaborative Scaffolding. *Computers & Education, 52*(1), 147-153.

P Wittenburg, H Brugman, A Russel. (2012). ELAN Annotation Format.

Race, Phil. (2001). *A briefing on self, peer and group assessment* (Vol. 9): Learning and Teaching Support Network York.

Roman, Flaviu, Mastrogiacomo, Stefano, Mlotkowski, Dyna, Kaplan, Frédéric, & Dillenbourg, Pierre. (2012). Can a table regulate participation in top level managers' meetings? *Proceedings of the 17th ACM international conference on Supporting group work* (pp. 1-10).

Schrire, Sarah. (2006). Knowledge building in asynchronous discussion groups: Going beyond quantitative analysis. *Computers & Education, 46*(1), 49-70.

Shirouzu, Hajime, Miyake, Naomi, & Masukawa, Hiroyuki. (2002). Cognitively active externalization for situated reflection. *Cognitive science, 26*(4), 469-501.

Soller, Amy, Wiebe, Janyce, & Lesgold, Alan. (2002). A machine learning approach to assessing knowledge sharing during collaborative learning activities *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community* (pp. 128-137).

Tao, Ping-Kee, & Gunstone, Richard F. (1999). Conceptual change in science through collaborative learning at the computer. *International Journal of Science Education, 21*(1), 39-57.

Teasley, StephanieD. (1997). Talking About Reasoning: How Important Is the Peer in Peer Collaboration? In L. Resnick, R. Säljö, C. Pontecorvo & B. Burge (Eds.), *Discourse, Tools and Reasoning* (Vol. 160, pp. 361-384): Springer Berlin Heidelberg.

Veldhuis-Diermanse, Anna Elske. (2002). *CSCLearning?: Participation, learning activities and knowledge construction in Computer-Supported Collaborative Learning in higher education*: Wageningen Universiteit.

Vygotskii. Mind in society: The development of higher psychological processes.

Walker, Marilyn, & Whittaker, Steve. (1990). Mixed initiative in dialogue: An investigation into discourse segmentation *Proceedings of the 28th annual meeting on Association for Computational Linguistics* (pp. 70-78).

Wang, Hao-Chuan, Rosé, Carolyn P, Cui, Yue, Chang, Chun-Yen, Huang, Chun-Chieh, & Li, Tsai-Yen. (2007). *Thinking hard together: the long and short of collaborative idea generation in scientific inquiry*. Paper presented at the Proceedings of the 8th iternational conference on Computer supported collaborative learning.

Webb, Noreen M, & Palincsar, Annemarie Sullivan. (1996). Group processes in the classroom.

Weinberger, Armin, & Fischer, Frank. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & education, 46*(1), 71-95.

Weinberger, Armin, Stegmann, Karsten, & Fischer, Frank. (2007). Knowledge convergence in collaborative learning: Concepts and assessment. *Learning and Instruction, 17*(4), 416-426.

APPENDIX A

SUMMARY OF ANNOTATION CODES

Authors	Team Attributes	Reasoning(Dialogic)	Reasoning (Argumentative)	Cognition/Meta Cognition	On Task	Miscellaneous
(McLaren et al., 2005),(Harrer et al., 2006)	Task Coordination			Task Selection		
(McLaren et al., 2010),(McLaren et al., 2007),(CHAMRADA & MANSOUR, 2009)	Task Manage-ment	1)Request for clarification(I) 2)Summary(I) 3) Critical Evaluation(I) 4)Reasoned Claim + Backing(I) 5)Contribution + Question(P) 6) Clarification followed by feedback. 7) Widening 8)Deepening	1)Contribution+ Counter(P) 2)Qualifier / Compromise (Partial Support) 3) Argument + Evaluation 4) Chain of Opposition		Topic Focus	1)Keywords In Context 2)Key Actions in Context 3)Link and Shape(D) 4)Combined text length 5)Difference in text length 6)Number and Length 7)Branching of Sequences
(Schrire, 2006)		1 Moves in discourse 1a. Initiate 1b. Response 1c. Follow Up				1) Length of argumentation

(Bloom et al., 1956)	1)Analysis (High Cognition) 2) Evaluation (High-Cognition) 3) Synthesis (High-Cognition) 4)Analysis (Low -Cognition) 5)Evaluation (Low-Cognition) 6)Synthesis (Low -Cognition)
SOLO Taxonomy (Biggs & Collis, 1982)	1)Pre Structural (Low-Cognition) 2)Uni Structural (Low-Cognition) 3)Multi Structural (Low-Cognition) 4)Relational (High Cognition) 5) Extended Abstract (High-Cognition)

Practical EnQuiry Model(2001)(Garrison et al., 2001)			1)Triggering Event 2)Exploration (Low –Cognition) 3) Phase of Integration (High-Cognition) 4) Resolution (High-Cognition)		
(Tao & Gunstone, 1999)	1) Equality 2) Mutuality 3) Joint on task 4)Number of tasks Completed 5)Number of tasks Completed Jointly				
(Noroozi et al., 2011)		1)Width 2)Depth 3)Justification/Reasoning		1)Relevance 2)Number of meaningful units	2) Correctness

(Wang et al., 2007)		1)elaboration 2)Positive Evaluation 3)Negative Evaluation 4)Question 5)Suggestion 6)Comment			(Off task Social) 1)Encouragement 2)Greeting 3)Acknowledgement 4)Meaningless Utterance (Typed)
(Hmelo-Silver, 2003)		1.Passive agreement 2Active Agreement 3)Seeking clarification 4)Brief answer without Reasoning 5)Explanation - With reasoning 6)Elaborate explanation	1)Conceptual Conflict 2)Task-Specific conflict	Monitoring(Meta) Reflection(Meta)	
(Hausmann et al., 2004)		Self-directed Other Directed Elaborative Co-Construction	Critical Co construction		
(Weinberger et al., 2007)	Knowledge Contribution Equivalence (Number of turns learners contribute in the discourse)	Transactivity, Externalization, Elicitation, Quick Consensus, Integration oriented Consensus, Problem related Construction, Concept Related Construction, Concept with Backing, Concept without Backing	Conflict oriented Consensus		Learners were On/Off task(Knowledge Contribution and Equivalence)

Knowledge Contribution Equivalence (Cohen 1994)	Number of turns				On/Off task(Knowledge Contribution Equivalence)
(Dillenbourg et al., 1995)		1)Task - “Negotiation” or simple task construction without much communication 2)Communicative - Establish shared understanding by establishing common referents 3) Mutual Adjustment (Co-construction)	Competitive Argumentation (Opposed Alternatives)	Cognitive - reading, searching, exploring , verifying Meta cognitive: Planning and analyzing (The higher the better)	Social Criticism (OffTask)
(Veldhuis-Diermanse, 2002)		Contribution + Illustration Argumentation Repeating information without drawing conclusion Contributing new info found in other infomation sources Referring to new	Contribution+ NO illustration/Argument Contradict + backing or refutation or restriction Contradict + No backing or refutation	Meta Cognition Monitoring Keeping Clarity Planning	Linking facts/Ideas/ Remarks

		information found in other sources Summarizing or evaluating information found in other sources				
(D. B. Clark & Sampson, 2008)	Organization of participants Responses	No Grounds (Level 0) Explanation without Evidence - Level 1 Explanation With Evidence (Grounds) - Level 2 Assessing dialogic Argumentation Evidence + Explanation + Coordination of multiple pieces of Evidence (Level 3)	Counter Claim Rebuttal Against Grounds Rebuttal Against thesis Clarification in response to rebuttal		Off task	
Gweon, Gahgene (2012 Dissertation) (Gweon, 2012)		Compare and Contrast Cause and effect			Relevance	
Holmes (1995)	Orientation	Solution Development Solution Evaluation		Problem Analysis Problem Critique		Non-task

APPENDIX B

FILE STRUCTURE OF FACT MEDIA SYSTEM AND ELAN

This section describes the schema of Fact XML data document and the Elan document format. These files are used by the Fact Media System and the Elan Viewer software's respectively.

Elan Document format: Elan document contains various elements that are used to construct the time line. The log data obtained from the fact media system is fed into this file for visual inspection and to gain insight about the dataset. The following are the various XML tags that are required to be modified in order to get the data into Elan document. The entire schema description along with its DTD can be found at (P Wittenburg, 2012)

ANNOTATION_DOCUMENT: The overall Elan document is enclosed within ANNOTATION_DOCUMENT tag. This is the root level tag and contains information such as date and the version number of the document.

HEADER: This tag contains the the media file location Information and also contains the time units used to build the file. Please be advised that our system uses time elapsed in milliseconds.

TIME_ORDER and TIME_SLOT: The most important tags that are present in the Elan document are these tags. The TIME_ORDER tag is the tag that encloses various instances of the TIME_SLOT tag. Each TIME_SLOT tag contains the ID which uniquely identifies that particular time slot along with the time stamp in milliseconds. The sample data is shown below.

```
<TIME_ORDER>
    <TIME_SLOT TIME_SLOT_ID="ts1" TIME_VALUE="541819"/>
    <TIME_SLOT TIME_SLOT_ID="ts2" TIME_VALUE="542060"/>
    <TIME_SLOT TIME_SLOT_ID="ts3" TIME_VALUE="542451"/>
</TIME_ORDER>
```

TIER: This creates a tier in the Elan file format. Each Tier holds a set of annotations. The tier has the TIER_ID attribute which is the name that is provided to the tier. This has to be unique. An example of TIER Tag is shown below.

```
<TIER LINGUISTIC_TYPE_REF="default-lt" TIER_ID="default">
</TIER>
```

ANNOTATION: Each tier encloses various instances of the ANNOTATION tag. Each annotation tag has "ALIGNABLE_ANNOTATION" tag which contains the start and the end TIME_SLOT_REFS tag which points out to the TIME_SLOT_ID shown above. By this way each annotation is marked in the Elan document. An example of the ANNOTATION tag is shown below.

```
<ANNOTATION>
    <ALIGNABLE_ANNOTATION ANNOTATION_ID="a11" TIME_SLOT_REF1="ts33"
TIME_SLOT_REF2="ts34">
        <ANNOTATION_VALUE>ScrollZoomCommand|152|111|1396|851|
        </ANNOTATION_VALUE>
    </ALIGNABLE_ANNOTATION>
</ANNOTATION>
```

Each annotation tag encloses ANNOTATION_VALUE which contains the value that has to be marked for that annotation tag. It also contains the ANNOTATION_ID which uniquely identifies the particular annotation.

CONSTRAINT: The constraint tag describes various constraints that are added to various tiers.

FACT XML DATA STRUCTURE

The data structure supported by Fact media system is the traditional XML format. The overall file contains the state and the sequence of events that has been performed by the student while working on the application. There are numerous tags that are available as a part of the Fact application. Certain variable of interest are only described here briefly.

SaveFactSurface: The SaveFactSurface tag is the root level tag of the XML document. This tag contains the mac address of the tablet, the group Id of the user and the user id of the user.

FactCard: All the type of cards that are present in the application is represented by the FactCard class. The various card variants are denoted by various classes which are parameterized the class attribute. The FactCard tag encloses all the information related to the Card. For instance, it contains all the positional information of various objects that are enclosed by it. These tags are not used for analysis. The cardId uniquely identifies a particular card class.

```
<FactCard class="edu.asu.fact.model.FactCard" cardId="30"></FactCard>
```

CanvasInkPaths: Canvas Ink paths denote the ink marks that are drawn as a part of the canvas. These ink marks have the following structure shown in the below illustration . It has to be noted that it has an internal tag called “path” which intern has two distinct set of events called the “Action Move” and “Action Line”. The Action Move command happens whenever the user starts drawing at a part of the screen. This indicates that the user has lifted his stylus from one part of the screen and places it on an part of the screen. This command also saves the x and y positions of the particular point. All the drawing made by the person are then stored as a series of “Action Line” Commands.

```
<canvasInkPaths strokeid="1030001">
 …
   <path>
     <actions>
       <IPathAction class="edu.asu.fact.model.ActionMove">
         <x>1000.0</x>
         <y>466.0</y>
```

```
      </IPathAction>
      <IPathAction class="edu.asu.fact.model.ActionLine">
        <x>2500.0</x>
        <y>466.0</y>
      </IPathAction>

    </actions>
 ...
</canvasInkPaths>
```

ImageBasedDrawingPaths : Image based drawing paths denote ink marks that are drawn as a part of the image. These ink marks are associated with the card they are drawn on. The following illustration shows the structure of the image based drawing paths.

```
<imageBasedDrawingPath strokeid="1220114">
  <savedStyle>STROKE</savedStyle>
  <savedJoin>ROUND</savedJoin>
  <savedCap>ROUND</savedCap>
  <path>
    <actions>
      <IPathAction class="edu.asu.fact.model.ActionMove">
        <x>155.78717</x>
        <y>62.884766</y>
      </IPathAction>
      <IPathAction class="edu.asu.fact.model.ActionLine">
        <x>155.78717</x>
        <y>62.884766</y>
      </IPathAction>


    </actions>
    <boundsComputed>false</boundsComputed>
    <drawingComplete>true</drawingComplete>
  </resetPath>
 ...
  <idAssigned>true</idAssigned>
   </imageBasedDrawingPath>
```

replayList: There are a sequence of replayList tags under the saveFactSurface tag. These tags denotes any particular command executed on behalf of the student who is working on the card. It has to be noted that, the sequence of events that has occurred are stored in the same order in the XML file. Each replay list contains a class attribute which denotes the kind of command object it denotes. The CardMoveCommand and ImageDrawingCommand are illustrated below along with their replay list objects. Each of these events has a “start Time” which denotes the start time of the event and “end Time” which denotes the end time of the event. Various other parameters that are discussed in Section x.x are also given present in the below object.

```
 <replayList class="edu.asu.fact.events.CardMoveCommand" cardId="30">
     <endTime>1397002599299</endTime>
     <startTime>1397002599213</startTime>
     <undoCommand>false</undoCommand>
     <botX>2387</botX>
     <botY>2504</botY>
     <topX>2151</topX>
     <topY>2103</topY>
     <dx>2</dx>
     <dy>7</dy>
  </replayList>
replayList class="edu.asu.fact.events.ImageDrawingCommand">
     <endTime>1390594263785</endTime>
     <startTime>1390594263434</startTime>
     <undoCommand>false</undoCommand>
     <cardId>3</cardId>
     <cardStrokeId>1220004</cardStrokeId>
  </replayList>
```

It has to be noted that this particular sequence of objects are extracted from the log file for further analysis.