Structural Model Discovery in Temporal Event Data Streams

Chreston Allen Miller

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Applications

Francis Quek, Chair

Christopher L. North

Narendran Ramakrishnan

Denis Gracanin

Louis-Philippe Morency

March 25, 2013

Blacksburg, Virginia

Keywords: Structural Model Learning, Temporal Behavior Models, Model Evolution, Human-Machine Cooperation, Temporal Event Data

Structural Model Discovery in Temporal Event Data Streams

Chreston Allen Miller

ABSTRACT

This dissertation presents a unique approach to human behavior analysis based on expert guidance and intervention through interactive construction and modification of behavior models. Our focus is to introduce the research area of behavior analysis, the challenges faced by this field, current approaches available, and present a new analysis approach: Interactive Relevance Search and Modeling (IRSM).

More intelligent ways of conducting data analysis have been explored in recent years. Machine learning and data mining systems that utilize pattern classification and discovery in non-textual data promise to bring new generations of powerful "crawlers" for knowledge discovery, e.g., face detection and crowd surveillance. Many aspects of data can be captured by such systems, e.g., temporal information, extractable visual information - color, contrast, shape, etc. However, these captured aspects may not uncover all salient information in the data or provide adequate models/patterns of phenomena of interest. This is a challenging problem for social scientists who are trying to identify high-level, conceptual patterns of human behavior from observational data (e.g., media streams).

The presented research addresses how social scientists may derive patterns of human behavior captured in media streams. Currently, media streams are being segmented into sequences of events describing the actions captured in the streams, such as the interactions among humans. This segmentation creates a challenging data space to search characterized by non-numerical, temporal, descriptive data, e.g., Person A walks up to Person B at time T. This dissertation will present an approach that allows one to interactively search, identify, and discover temporal behavior patterns within such a data space.

Therefore, this research addresses supporting exploration and discovery in behavior analysis through a formalized method of assisted exploration. The *model* evolution presented supports the refining of the observer's *behavior models* into representations of their understanding. The benefit of the new approach is shown through experimentation on its identification accuracy and working with fellow researchers to verify the approach's legitimacy in analysis of their data.

GRANT INFORMATION
This research has been partially supported by:
FODAVA grant CCF-0937133, NSF grant IIS-1053039, and NSF IIS-1118018.

Dedication

To my beloved wife, Christa

To my daughter, Hannah

To my parents, Keith and Joyce

To my brother, Justin

To all my friends in Blacksburg

Acknowledgments

This research was partially funded by FODAVA grant CCF-0937133, NSF IIS-1053039, and NSF IIS-1118018. I want to thank Dr. Francis Quek for his guiding support, my committee members for their input, my family for their love and support, and my wife, Christa Hixson Miller, for her never ending support. It is because of her that I was able to finish.

Due to copyright permissions, the video frame in Figure 3.5 was replaced with one owned by me.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The motivation of this dissertation is to aid domain experts (e.g. a psychologists or psycholinguists) in behavior analysis in a meeting room setting. The difficulty of behavior analysis is a domain expert, or just expert, is needed to sift through and explore the data, applying their knowledge in extraction of pertinent information. The data normally consists of surface level information, i.e., observable features of phenomena that can be automatically extracted and processed. Such data is characterized as descriptive, temporal events with associated time points, e.g., Person A walks up to Person B at time T. Sifting through large datasets of this nature to identify and discover behavior(s) of interest is a time consuming and tedious task. The high level meaning of what is extracted is more difficult to automatically classify since such meaning and understanding is found at an abstract level. This classification is more naturally performed by humans, especially experts of the data in question.

Hence, we want to support experts in analyzing their data and identify behavior phenomena that are relevant to them. These experts have background knowledge, experience, and expertise that allow them to identify meaningful *behavior models*. A *behavior model* defines a sequence of behaviors, also called a *pattern* or *model* for short. This knowledge allows them to rank the data as they sift through it, giving structure with which to base model creation. The data space poses a challenge for search and discovery, leading to the need to seek out a human-in-the-loop solution. This dissertation presents an interactive approach that engages the expert as a guiding factor for *behavior model* identification.

In this chapter, we start by discussing the challenges of behavior analysis, an overview of behavior analysis approaches, and then the approach presented in this dissertation. We then conclude with research questions and the dissertation outline.

1.1 Challenges Faced by Behavior Analysis

One goal of behavior analysis is to identifying meaningful phenomena within a large behavior corpus. There are a few challenging nuances of human behavior and its analysis that need to be addressed to approach this goal. First, human behavior is variant. The idea represented by a behavior interaction, e.g., a greeting between two individuals, may be formulated many different ways in the data making modeling difficult. How does one identify situated instances of behavior when the way they exist in the data may be unknown? The second challenge is every observed behavior has the potential to be relevant to an expert depending

on his/her analysis goal(s). Hence, there is no concept of "noise" but rather one of relevance. For example, consider the situation where three students are working together on a math problem when a door slams nearby and draws their attention. One expert may analyze the co-construction of space based on the students' aligned gaze while another may analyze interrupting events. The door slam is "noise" to the first expert but not the second. Lastly, a *pattern's* value to the expert may not be based on frequency or statistical significance but on subjective relevance. We will present some scenarios that exemplify these challenges.

The variance of human behavior can be exemplified by considering the scenario of two individuals (A and B) greeting each other in a social setting. The sequence of events that are relevant are 1. A approaches B, 2. B acknowledges A's presence, 3. A initiates the greeting, 4. B responds. However, each of these four steps can happen in many different ways and possibly some skipped are reversed. For example, step 1 could be A intentionally approaches B or A bumps into B. B may not even notice A and begin to walk off and A needs to catch up to B. Step 2 could be B turns to A or this step does not even occur. Step 3 could be A reaching for a handshake, saying "hello" (among many verbal phrases for greeting), or a pat on the back. B may even initiate the greeting. B may respond in kind or not at all in step 4. Hence, there are many variations of how the greeting can play out in which the idea of the behavior is simple but how it exists in the data can vary widely.

The challenge of variance can also be found in the field of Conversation Analysis (CA) where Harvey Sacks performed an ethnomethodological study with the goal of discovering the means that governed the structure of conversational interaction [129]. This study analyzed

the details of people's conversations based on recordings of real conversations and drew conclusions from these observations. Experts reviewed the data and made observations out of which a set of conclusions was derived. For example, the study showed the many uses of "hello" based on varying contexts. Sometimes "hello" is used for greeting, and others for exclamation. This demonstrates variance of context of meaning.

These scenarios exemplify the contextual variance faced when searching a behavior corpus. Such variance leads to employing an exploratory process where even what specific behavior to look for may not be known. This exploratory process can be seen in an analysis of an interaction session where college seniors employed a high-resolution tabletop [4, 94] for cooperative analysis in a history class. The students were performing a sensemaking task on snippets of historical information on the Jamestown settlement in Virginia. The video and audio recordings of the session were coded for (vocal) referential foci, pointing gestures, and gaze fixation using MacVisSTA [125], a multimodal analysis tool. This coding revealed that the students sectioned off an area of the tabletop to organize the joint understanding of a history timeline. The students spontaneously developed a ratification process by which any proposed change to the timeline was initiated by vocal announcement (e.g., "I think this event belongs here") that results in a convocation of shared gaze and a series of pointing gestures accompanied by utterances discussing timeline accommodations. These utterances were echoed by the wielder of the control device that does actual timeline arrangements to maintain common ground among the group. This sequence of behavioral instance was discovered (not searched for) during the analysis.

The second challenge of no "noise" can be seen in several scenarios. The first being the greeting example described above where different formulations of a greeting may be interesting to different experts. One expert may be focused on identifying hand shake greetings where another may want to identify failed or "awkward" greetings. The instances of the first expert would be "noise" to the second, and vice versa. Plus, for each expert there may be distractors such as interrupting events between any step from other people in the scene. In the case of the CA scenario above, two experts may be interested in analyzing two different conversational contexts of "hello" which are mutually exclusive. In the history sensemaking scenario, it is unknown what would be considered noise since an exploratory analysis is needed to identify what is meaningful.

The third challenge is the value of a *pattern* is based subjectively on each expert's interests and/or analysis goals. The value is based on the judgment and interpretation of experts. This can be seen in the above described scenarios where what is deemed important must be measured by the experts themselves.

The take home message here is that behavior analysis has inherent characteristics that pose as challenging design considerations for supporting behavior analysis. The approach presented in this dissertation was designed taking these challenges into consideration.

1.2 Overview of Behavior Analysis Approaches

We provide an overview in this section to current behavior analysis approaches and how this analysis is conducted. Behavior analysis consists of observing human interactions and providing meaning and understanding for what is observed. There are a number of different settings for this kind of analysis, e.g., human to computer, human to human, small group setting, large crowd, etc. Currently there is much work in supporting behavior analysis in many settings. Our work is focused on the analysis of people in a small group setting, more specifically meeting room analysis. This is the context of this dissertation unless otherwise stated.

From the literature, behavior analysis is conducted with some element of automation and human assistance following two trends of analysis. The first trend focusses on one signal source of behavior data, e.g., speech, to provide a channel of information to analyze and extract relevant behavioral information. The second trend performs analysis across multiple signal sources, e.g., speech, gaze, and gestures, to form a multimodal solution.

Each trend uses varying amounts of data in which to analyze behavior and extract meaning in some form. The data is segmented (automatically and/or manually) into discrete events representing logical categories/labels or continuous signal data on which further processing can be applied. This further processing may include human aid solely or human aid that iteratively feeds back into an automated process.

1.3 Interactive *Model* Evolution

The above examples provide concrete scenarios to exemplify the challenges faced in identifying meaningful *behavior models*. The knowledge of experts is invaluable for identifying relevant phenomena. To support data exploration and identification, it is necessary to explore how the events interact and relate with respect to order and time. Incorporation of experts' knowledge and temporal event reasoning can allow the growth or evolution of a *model* to a desired formulation. This dissertation seeks to provide a greater understanding of the relational sequencing of temporal data and how these relationships can be used for building and identifying *models* representing phenomena of interest.

Model construction is conceived as a two stage process of initial specification and incremental piecewise refinement. This refinement takes the form of small additions or subtractions that evolve the *model* into the desired representation. Such piece-wise construction is performed on the beginning and end atomic units, also known as semi-intervals [39]. The exploration and explication of potential relational possibilities allows for construction of a *model* that is based on relative and temporal ordering; the ordered relationships one semi-interval has with respect to others either by relative order or some timing information.

We approach the problem by looking at *patterns* of events within multimodal channels. As mentioned before, a *pattern* defines a sequence of behaviors, also called a *behavior model* or *model*. Behaviors are encoded as annotated event intervals with relative and temporal order being implicitly or explicitly defined. An example is a greeting among two individuals

with the possible formulation: <A walks up to B>[within 1 second]<A shakes B's hand> and <A says "Hello">. Here one could potentially 'evolve' the *pattern* by successively adding/removing relationships with other events, and/or pruning relational connections. However, evolving this *pattern* without guidance is a large search space even for a small *pattern*. Our solution is founded on creating a formalism of a *pattern* based on structure, timing, and ordered relationships. We operate on a *pattern* at the semi-interval level (start or end of an interval). This representation was first introduced in [39] and later revisited in [101]. Semi-intervals are also known as instant-based models (points) in multimedia authoring and synchronization [14, 15]. Semi-intervals allow a flexible representation where partial or incomplete knowledge can be handled since operations are on parts of an interval and not the whole. *Patterns* are evolved at the semi-interval level, which we call a *1-step* change, representing the smallest change that can occur.

The novelty of our approach is in engaging the expert in an interactive, online process that iteratively uses his/her input as opposed to the expert providing example *models* of interest that a system uses to build a classifier of what may be relevant. The evolution process allows the expert to step through the data during an analysis and converge to an end result. The significance of this is providing a method with which to explore a conceptually rich data-space and let the guiding factor be the knowledge and expertise of the expert, as he/she knows what is and is not interesting to them. Hence, our goal to to support the expert in identifying relevant *models* within their data.

1.4 Research Questions

The challenges highlighted by this dissertation lead to several research questions. This dissertation is about seeking answers to the following questions.

1. *How does one* ***explore and identify relevant models*** *within multimodal data?* This is the overarching question of this dissertation. The motivation of the research is based on the necessity to explore multimodal data and identify *models* that are relevant to an expert's analysis goal. Therefore, we seek how this can be done effectively.

2. *How can one* ***represent*** *multimodal data to support and facilitate exploration and model identification?* A data representation is needed that is understandable for experts to facilitate the manipulation of a model's structure. Some complex representations used hinder such understanding and manipulation. The representation also needs to have the representational powerful to capture the necessary aspects of *behavioral models* that are important to experts.

3. *How does one* ***seed such exploration****?* Where does one start an exploration of this fashion? We need to investigate what a beneficial and informed starting point is for experts and how this can be formulated.

4. *How does one* ***formulate a model*** *that matches relevant instances in the data and can this formulation* ***support insight discovery****?* How can a *model* be formulated so that relevant matches can be found based on this formulation? This ties into the

chosen representation. Does this formulation support the expert in insight discovery of their data and lead to meaningful understanding? We want to see if the created approach actually does support the expert in their analysis task.

1.5 Dissertation Organization

The rest of this dissertation is organized as follows. In Chapter 2 we discuss how behavior analysis has been and is currently conducted. This exploration unearthed trends and approaches taken. This chapter then discusses how the surface level information is extracted for analysis purposes. Here we look at what techniques and approaches are used for supporting behavior analysis. The important aspect from this chapter for our work is the temporal and relational ordering of events which is detailed at the end of the chapter. The organization and modeling of this data is then discussed in Chapter 3. Here the varying *model* construction techniques are detailed along with several data modeling techniques. The story of our research into how to explore a temporal event data-space is told in Chapter 4. Here the four pieces necessary for our solution are presented. Next, Chapter 5 describes the evaluation of our approach. This comprises of four phases in which accuracy of identified *models* is assessed along with feedback from several researchers. Chapter 6 describes the details of the software developed to test and realize our approach. The current state in terms of functionality are discussed as well. After which, in Chapter 7, our contributions are highlighted, future work is discussed, and conclusions close the dissertation.

Chapter 2

Analysis Approaches

In this chapter, an overview of current methods employed for conducting behavior analysis is reviewed. The first section describes what behavior analysis is and the trends for such analysis seen in practice. After which, we discuss how behavior corpora are segmented for analysis. This leads to describing the relational and temporal characteristics of the data, of which our work is grounded. The final section details the relevant literature for temporal logic and relational ordering, from which the chapter concludes.

2.1 Behavior Analysis

For our purposes, behavior analysis is defined as observing human interactions and behavior and reaching understanding and meaning through analysis of these observations. Currently there is much work in supporting behavior analysis in many settings, e.g., human to com-

puter, human to human, small group setting, large crowd, etc. The motivating setting of this dissertation is the analysis of people in a small group setting, or more specifically, meeting room analysis. Later on, we will show that the approach of this dissertation has wider applicability.

From the literature, behavior analysis is conducted with some element of automation and human assistance in which there are two trends of analysis. The first trend focusses on one signal source of a behavior instance, e.g., speech, to provide a channel of information to analyze and extract relevant behavioral information. The second performs analysis across multiple signal sources, e.g., speech, gaze, and gestures to form a multimodal solution.

The first trend, a unimodal approach, is applied in earlier literature. Here the focus is around processing and segmentation of text and speech. Much work has been done in segmenting text into meaningful sections, providing insight into the structure of the text. Segmenting text allows for the assignment of categorical labeling which in turn provides higher-level meaning. Most work in this area takes advantage of language structure and lexical analysis to provide insightful segmentation. Hearst in [52] segmented ideas and topics within text using lexical analysis. In [78] the authors process spoken text in audio format and use human aid to tune automatic results. Flammia in his dissertation [35] presents a tool to provide a user interactive annotation system for discourse segmentation of dialogue with segmentation performed by humans. The work in [40] provides automatic segmentation based on lexical and language information. Automatic extraction of audio features were performed in [73] for aiding in classification. The crux behind these works is employing computer supported

automated processing and presenting the results to the observer. The results, in turn, allow the contents of the text to be presented as abstract ideas.

These works and others [112, 113, 145, 19] provide segmentation and classification support of text which in turn supports a higher-level understanding of the text. However, unimodal approaches are only able to provide a subset of available information since the processing is along one dimension.

The second trend, a multimodal approach, incorporates multiple modes of interaction and explores how segmenting information along multiple dimensions can provide greater insight. There has been a strong trend toward creation and analysis of multimodal corpora. The authors of [123, 120, 122, 124] investigate and report on how human interactions and communications are inherently multimodal. The the authors of [124] argue the value and deeper understanding multimodality adds to analysis of human behavior. The aforementioned research and others [88, 89, 22, 21, 23] embrace this form of analysis and perform an extensive analysis of human behavior within a meeting room setting. They captured many modes of communication (e.g., speech, gesture, head pose), time synchronized them and performed analysis across all these channels of communication. The results showed the added benefit of a multimodal approach as behavior phenomena could more accurately be captured and recognized.

The result from this multimodal approach is a detailed multimodal infrastructure developed for behavior analysis [24], which has been referenced by other work as important guidelines and a rich dataset for analysis in the area of behavior analysis [22, 88, 23, 21, 89, 125]. The

supporting tool of this effort in the reported works above is MacVisSTA [126, 125].

The importance of conducting multimodal analysis has been recognized by other researchers in recent years as a number of research projects have also been exploring this area. The research in [107] applies a multimodal approach to analyzing a haptic device for aiding blind students in learning mathematics. Analysis in a large video corpus of hand gestures was performed by [141] where observers annotated keyframes and allowed the system to perform automatic extraction between the keyframes. Georgeon, et al. explores creating symbolic representations of behavior through the use of activity traces [42]. Their focus is similar to ours in that they view the relations between events as an important key in understanding a higher level meaning of the events. They also address supporting analysts in *modeling* activity of subjects (observed human behavior) through symbolic representations. Our work differs in that our focus is in providing a means of supporting the *model* creation process through interactive exploration and discovery. Since a number of systems have been created to support multimodal analysis in its many forms, there has been a joint effort to aid current multimodal analysis by creating a standard data format that can be used to share datasets and results between multimodal analysis tools [132].

Many multimodal corpora have been created to follow this analysis approach which predominantly consists of sequences of descriptive events (*behavior models*). The VACE/AFIT [24] multimodal meeting corpus is a detailed recording of multiple sessions of Airforce officers partaking in war gaming scenarios in a meeting setting. The Semaine corpus [86] is a collection of emotionally colored conversations. The Rapport and Face-to-Face corpora [46, 103]

are sets of speaker-listener interactions. One of the largest to date is the AMI corpus [18] which contains 100 hours of recorded meetings. Mörchen created a series of datasets of varying degrees of modalities [101]. These mentioned corpora and datasets are highlights of a growing community of such data.

With the increasing number of multimodal datasets, tools were needed to visualize the data for analysis. These tools have been developed to visualize multi-channel annotation information coupled with varying degrees of multi-channel support of audio and video. Well known examples of these tools are MacVisSTA [126] (mentioned above), ANVIL [64], Theme [146], EXMARaLDA [131, 34], ELAN [154], C-BAS[1], Transformer, and VCode [51]. The AMI corpus uses a different approach through use of the Nite XML toolkit. The Nite XML toolkit provides extensive support for complex annotation representation and a supportive interface in which the visualization is centered around annotated transcription text (e.g., dialogue) of a corpus and is linked to supportive media, e.g., audio or video.

In general, manually encoding the data is necessary at some point in annotating the data. Automatic processing is applied for pre-processing and/or for aspects in the data that can be annotated with high accuracy using automation. More researchers are turning to machine learning techniques to aid in classification and labeling of data, such as [103, 7, 140, 110, 149]. The results from automatic and manual encoding/labeling of the data allows the observer (or expert) to view the annotated data, analyze it, and draw conclusions and insight.

A challenge is providing good segmentation of the data. Segmentation results in sequences

[1]Developed at Arizona State, http://www.cmi.arizona. edu/, but the url for C-BAS is broken.

of event information potentially consisting of multiple channels of information. Each channel represents an interaction aspect recorded, e.g., gaze, speech, or trajectory traces of hands. This allows observers to apply their knowledge and expertise to the data without needing to be burdened by the vigilance task of segmenting themselves. In the context of this dissertation and our motivating domain, how events are distinguished and subsequently extracted is known as discourse segmentation. This is discussed in the next section.

2.2 Discourse Segmentation

Understanding how events are extracted and formatted is important for supporting behavior analysis. In this section, we discuss the current approaches and methods used to perform discourse segmentation. At a high level, discourse segmentation is performed through many multimedia processing and segmentation techniques. It is the science of segmenting a signal whether it is text, speech, video, audio, or some other signal data. The significance is the meaning behind the signal, which in our case is human interactions and behavior. Most earlier methods found in the literature focused on discourse segmentation in text and speech. Research, such as [52, 78, 112, 113, 40], use lexical and language rules to aid in segmenting text and speech into topical and categorical sections. Much of this work is based on the idea that human-to-human dialogue can be modeled as a sequence of discourse segments. However, this can be challenging as such segmentation may be open to interpretation. Automated processing based on lexical and language rules are employed coupled with human

tuning and/or machine learning. A commonly utilized machine learning technique has been the C4.5 decision tree, as in [78, 40, 19]. Most of the research reviewed address segmenting discourse of English text, but some exceptions, such as [19], operate on a different language. Each language has its own assumptions of structure, hence, for each language, different assumptions must be made for segmentation. Such research highlight the need to apply the appropriate segmentation rules dependent on the context of the data.

Later research uses other aspects of speech, such as prosodic information [73] and cue phrases [145], to aid in performing segmentation. Such work began to see the importance of utilizing several different aspects of discourse in order to better extract meaningful segments of information. Incorporation of more information to produce better and more accurate analysis results can be seen as leading to the multimodal approach to behavior analysis.

Various methods are employed for conducting segmentation for multimodal behavior analysis. They generally boil down to manual encoding/labeling the data and automated processes that can annotate or label the data within a certain degree of accuracy. Manual encoding, as seen in [107, 89, 88, 124], produces very accurate annotations that are normally validated across several annotators. However, this is tedious and time consuming, hence, application of automated encoding is very beneficial along with some human guidance. Examples can be seen in [141, 42, 21, 23, 24, 22]. This allows a computer to sift through the data, parse out information, and present the results to the user who then decides the next step. This allows some of the processing load to be lifted from the user. Some forms of this refine results through Relevance Feedback [127] which iteratively applies the observer's input to inter-

mediate results, feeding the observer's input back into the automatic processing algorithms [141, 42]. In these approaches, the observer initializes or primes the processing algorithms. The authors of [141] also employ manual annotations to define keyframes in which to perform interpolation, allowing an interactive key-framing pipeline to model hand and head traces in video. Some behavior analysis approaches are fully automated, such as [22, 23, 140], however, the datasets were preprocessed.

Many computer vision techniques have and are being used to extract gaze, speech, and/or gesture information automatically ([148, 123, 120, 122, 121, 103, 104, 5] among many others). This information is automatically extracted and can be presented to the observer as time-aligned sequences. Some of the multimodal work discussed in Section 2.1 performed the segmentation by manually scrubbing through the data [107, 89, 88, 124].

Within the last few years, a number of techniques and technologies have emerged to aid in segmentation. These include using the Microsoft Kinect sensor [62] to segment body pose, position, and depth information [160, 16, 82, 76, 77, 33, 45, 157] and speech recognition [38], crowdsourcing, e.g., mechanical turk, for performing intelligent segmentation by large groups of human workers [141, 111, 79, 61, 90, 20], plus many automated techniques [56, 81, 137, 161]. These are just a few. For a more extensive look, the curious reader is referred to the recent proceedings of ACM Multimedia (MM) and the International Conference on Multimodal Interaction (ICMI).

The information that is automatically extracted and segmented is surface level information. Surface level information is observable features of phenomena that can be automatically ex-

tracted and processed, such as blob detection, head position, face detection, and hand traces. Such extraction is becoming increasingly commonplace as better segmentation algorithms are being developed. Greater detail of such techniques is not the purpose of this dissertation but is a very active area of research as seen above and [41, 96, 156, 49, 57, 139].

The important point of this discussion is that there are many advanced techniques for automatically segmenting aspects of discourse. The above is by no means an exhaustive list representing the state-of-the-art, but representative of the trend. The results of this segmentation are the categorical event data our approach operates on. A very important element of the event information extracted is the temporal aspect, which provides timing and relational ordering with respect to other events. This provides a starting point into how events relate and is one aspect our approach capitalizes upon in supporting *model* construction and exploration. The next section discusses the literature of temporal logic and relational ordering in the context of this dissertation.

2.3 Temporal Reasoning and Relational Ordering

This section discusses the related work of temporal relations with respect to ordered events. We detail work on temporal ordering and *pattern* discovery. Our approach is founded on creating a formalism of a *pattern* based on structure, timing, and ordered relationships. The ordered relationships that are inherent to temporal event data have been an active area of research. Allen in his work [3] formulated thirteen relationship principles that express all

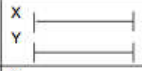
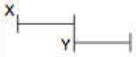
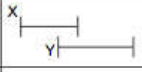
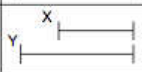
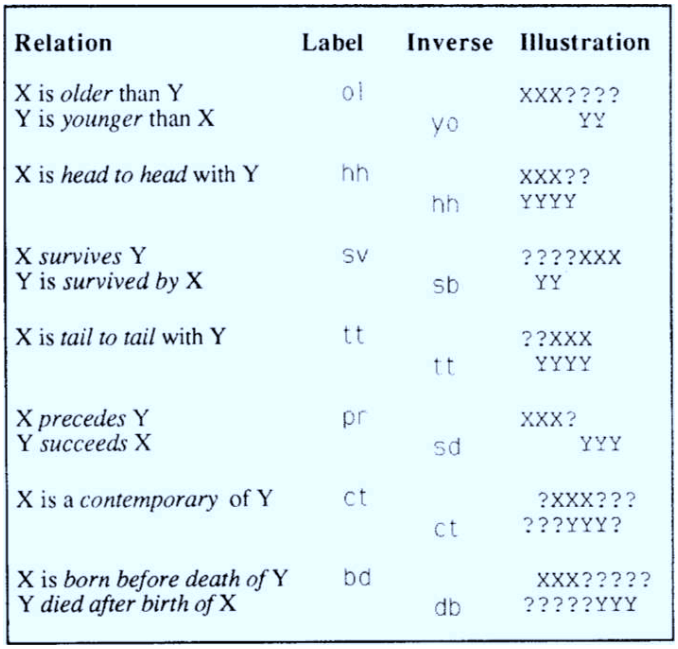
Relation	Relation for Inverse	Pictorial Example	Endpoints Constraint
X before Y	Y after X		$X.t_e < Y.t_s$
X equal Y	Y equal X		$X.t_s = Y.t_s$ $X.t_e = Y.t_e$
X meets Y	Y met-by X		$X.t_e = Y.t_s$
X overlaps Y	Y overlapped-by X		$X.t_s < Y.t_s$ $X.t_e > Y.t_s$ $X.t_e < Y.t_e$
X during Y	Y contains X		$X.t_s > Y.t_s$ $X.t_e < Y.t_e$
X starts Y	Y started-by X		$X.t_s = Y.t_s$ $X.t_e < Y.t_e$
X finishes Y	Y finished-by X		$X.t_s > Y.t_s$ $X.t_e = Y.t_e$

Figure 2.1: Allen's temporal relations. Figure courtesy of P. Kam and A. Fu. Discovering temporal patterns for interval-based events. *Data Warehousing and Knowledge Discovery*, pages 317326, 2000. Used with permission from A. Fu, 2013.

the possible ordering relationships between two events. These can be seen in Figure 2.1. Since then, much work has been conducted to detect, extract, and represent such temporal information. Examples are [28, 50, 58, 97, 98, 135, 147]. These research endeavors represent the trend of discovering events and temporal relationships from temporal data.

After Allen, Freksa revisited interval relationships at the semi-interval level (an interval's start and end) [39]. Semi-intervals allows a flexible representation where partial or incomplete knowledge can be handled since operations are on parts of an interval and not the whole. Freksa's relationship principles can be seen in Figure 2.2. All of Allen's relationships can also be represented by Freksa's formalism. Interestingly enough, very little work has focused on using semi-intervals. The most notable was completed by Mörchen and Fradkin in [101] where they explored semi-intervals for use in unsupervised pattern mining.



Relation	Label	Inverse	Illustration
X is *older* than Y Y is *younger* than X	ol	yo	XXX???? YY
X is *head to head* with Y	hh	hh	XXX?? YYYY
X *survives* Y Y is *survived by* X	sv	sb	????XXX YY
X is *tail to tail* with Y	tt	tt	??XXX YYYY
X *precedes* Y Y *succeeds* X	pr	sd	XXX? YYY
X is a *contemporary* of Y	ct	ct	?XXX??? ???YYY?
X is *born before death of* Y Y *died after birth of* X	bd	db	XXX????? ?????YYY

Figure 2.2: Freksa's temporal relations. Figure courtesy of C. Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1-2):199 227, 1992. Used with permission from C. Freksa, 2013. Original caption: "Eleven semi-interval relationships. Question marks (?) in the pictorial illustration stand for either the symbol denoting the event depicted in the same line(X or Y) for for a blank. The number of question marks reflects the number of qualitatively alternative implementations of the given relation."

Much work has be done to mine, process, and predict temporal knowledge. Mörchen [99] presents Time Series Knowledge Mining (TSKM) which is a method to extract and represent temporal knowledge. He further discusses methods for prediction of events. Such prediction based on temporal patterns is closely related to our work which is more focused on the relationships between events with respect to relational order and history. In [54], prediction is acquired through probabilistic modeling based on detected temporal patterns. The interest of our approach is not predicting *patterns* but understanding the ordered relationships between events, i.e., how do two events relate structurally and contextually.

The authors of [71] report on discovering the relationships between patterns. A number of techniques designed to represent temporal data and discover patterns using unsupervised means are described in [100]. Our data domain can be described as categorically descriptive events (nominal data), which [71] comments that this feature of data renders some methods of modeling inapplicable, such as statistical analysis approaches dependent on numerical data.

Assigning relational structure to nominal data can be difficult since the meaning can be context dependent and open to interpretation. Research conducted has addressed this difficulty in various ways. The authors of [74] use a similarity technique in performing unsupervised learning with mixed numeric and nominal data. Through this they were able to cluster the data to provide classification information. This is in contrast to [25] where the authors use dissimilarity to learn classification information for nominal data. Fuzzy logic [158, 159] has also been employed for discriminating between nominal features. However, fuzzy logic

requires pre-defined rules that govern how the nominal features relate and hence painstaking consideration is needed before processing can begin.

A number of probabilistic techniques have been employed to describe nominal data. Such techniques include stochastic context-free grammars, Hidden Markov models (HMMs), and class-based n-grams [55, 142, 143]. N-gram processing has been successfully employed as a probabilistic model for speech processing [12, 143]. This kind of processing gives a probabilistic model of successive speech features based on previously seen features, hence providing ordered relational sequence information. More specifically, our research [95] explores how to structure nominal data based on semi-interval processing using n-gram techniques from speech processing. Other work [39, 83, 101] has explored the concept of breaking temporal events into core units of beginnings and endings through semi-interval relationship processing.

Situated within the data mining domain, the symbolic temporal pattern mining (STPM) approach focuses on discovering "interesting" patterns among *symbolic* time series data [71, 100]. STPM is related to the need to represent time and timing among events. One such approach is T-patterns developed by Magnusson [83] in which a sequence of events will occur within certain time windows of each other, e.g., $A_1 \xrightarrow{T_1} A_2 \xrightarrow{T_2} A_3$ for time intervals T_1 and T_2. T-patterns are used as the basis of pattern representation and identification in Theme [146] where each T_i, for $i \geq 1$, is set through various statistical methods. Several other research endevors have pursued T-Patterns [10, 150]. Related are frequent episode mining (FEM) algorithms [115, 130] that operate on interval and point data in order to identify event

sequences of varying lengths (episodes) given certain timing restrictions between events. FEM algorithms identify these episodes given a threshold (frequency or statistically based). They are a very efficient methods to identify trends in data.

Temporal reasoning and relational ordering of event data has been extensively researched. However, not as much research has be focused on processing using semi-intervals. It is these event boundaries, how they interact and relate in time and order that is crucial to behavior analysis. Hence, how can we support an expert in analyzing data of this nature? To answer this, we need to assess the data modeling schemes in practice and choose one for our approach appropriately. The next chapter details techniques for temporal data modeling and describes the representation used in our approach.

Chapter 3

Temporal Data Modeling

This chapter summarizes the different data models and representations for temporal event data. The discussion begins with the two paradigms of modeling: parametric and structural. We then highlight the different representations commonly employed for temporal data. The chapter concludes with a discussion on the representation chosen for our approach.

3.1 Parametric vs Structural Learning

In the process of supporting an expert in identifying a *model*, understanding the different *model* learning methods is essential. By learning we mean identifying *models* in data based on some specified criteria. This section discusses the two major classes of learning: parametric and structural. Each class learns a data *model* differently. Understanding each class better will aid in choosing the best class to apply to our problem.

For learning *models*, the most common approach is parametric where one explicates a training data set and constructs a representative *model* based on the data. The weights or parameters of the *model* are learned based on training data. The *model* structure is set and the proper values are assigned to the respective weights/parameters that reflect the trends seen in the training data. Classic examples are Hidden Markov Models (HMMs), Neural Nets (NNs), Decision Trees [128], and Support Vector Machines (SVMs) [30]. Systems built on Relevance Feedback [127] refine a *model* by adjusting weights according to what the user chooses. Weights and parameters are normally hidden from the user as the underlying structure is complex.

In structural *modeling*, the *model* structure itself is modified. Such changes help mold the *model* to an optimal or desired formulation. An example of such *model* construction is the Evolutionary Computing strategy known as Genetic Programming (GP) [32, 67, 68, 69, 70] where a population of individual solutions (*models*) undergo evolution through iterative generations of change. The idea of structural variation has also been explored in numerical contexts where numerical model modification is used, as in [27, 59, 80]. Structural variation is explored in [27] while [29] discusses structural variation seen in nature with respect to genetics. A more popular paradigm has been model selection. The works of [105, 106] explore the use of multiple models and model tuning based on the current environment. Model updating has also been used as seen in [48] where Adaptive Model Checking (AMC) is used to update a model based on verification of the model's performance. Model selection is also used as in [136] where adaptive model selection is employed based on a penalty scheme

that dictates which model to use based on the current situation. In contrast, exploration of model modification outside of numerical context appears to have received less attention.

Each class of learning, parametric or structural, has its own set of assumptions and implications. Parametric assumes an already defined *model* structure, e.g., setting the number of hidden states in an HMM or layers in a NN. The appropriate weight/parameter values for the *model* are learned with training data. Structural assumes an initial *model* structure which is subject to change. The *model* can be altered to explore different variations in the data. An interesting note is that parameter variation can be applied to structural variation. Once a *model* has been established, learning weights may take place if desired.

The approach in this dissertation uses structural modification principles to support the creation and discovery of the best representative *models*. The structural variation methods described above are applied to a numerical context with which much more literature is available than can be presented here. The numerical context is also in contrast to our target data domain. Hence, the problem of application to our challenging domain has, as yet, not received significant research attention.

3.2 Temporal Data Models

In this section, we present data models commonly used for temporal data. For our purposes, temporal data is defined as events represented by points and/or intervals with an associated order and/or absolute time stamps. There are a number of models used to represent temporal

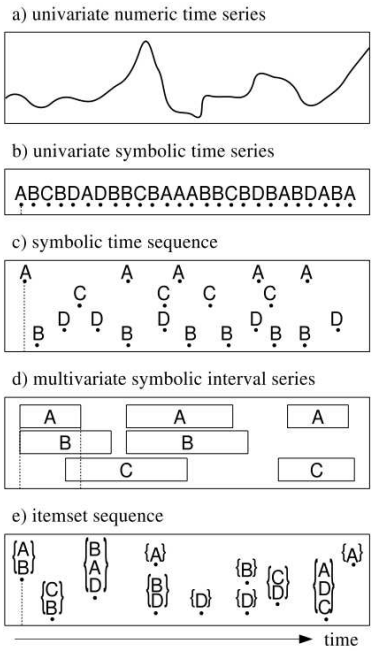


Figure 3.1: Example temporal data models. Figure courtesy of F. Mörchen. Unsupervised pattern mining from symbolic temporal data. *SIGKDD Explor. Newsl.*, 9(1):4155, 2007. Used with permission from F. Mörchen, 2013

data. Mörchen in [100] reports on temporal data models that are typical in practice. These can be seen in Figure 3.1. (A) illustrates the univariate numeric time series where numeric values reside at each time point. This is commonly used in statistics. In (B), we see univariate symbolic time series where a nominal value (symbolic or categorical) resides at each time point. Many data mining techniques for sequence or pattern chains use this representation as in [75]. (C) illustrates symbolic time sequence where nominal values reside at each time point with possibly two or more values sharing the same time point. An example use can be seen for network monitoring [84] where status events follow this representation. Multivariate symbolic interval series (D) represent event intervals in time with potential overlap. Many behavior analysis techniques utilize this representation as will be discussed in the next section. Lastly,

in (E) is the item set sequence where each time point has one or more item sets, i.e., a set of symbols, associated with it. This representation is employed for sequential association rule mining [2].

As noted above, the representation most often seen used to represent discrete events in behavior analysis is multivariate symbolic interval series or "music score" representation. This is the representation we use and is discussed in more detail in the next section.

3.3 "Music Score" Representation

A simple method employed to represent discrete events is time intervals consisting of a beginnings and endings (semi-intervals). This approach is known as multivariate symbolic interval series, also known as "music score". An example can be seen in Figure 3.2A. This is a common representation used for multimodal analysis of behavior [126, 64, 146, 131, 34, 154, 51]. Plus, detailed analysis published by the experts that motivated this dissertation [21, 88] describe their analyses using this representation. A linear sequence of events can easily be represented (Figure 3.2B) as well as more complicated relationships as explained in Allen's work [3] seen in Figure 2.1 and later Freksa's work seen in Figure 2.2. The details of Allen and Freksa's work was discussed in Section 2.3. In the rest of this section, we detail the use of this representation and why it is our chosen representation.

Allen viewed events, as most research does, as an atomic unit. The possible relationships between events explained by Allen represent the overlapping aspects of discrete events. The

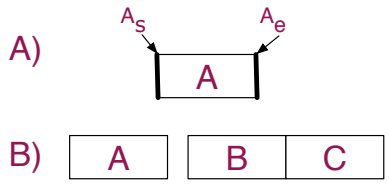


Figure 3.2: A) Graphical rectangle created through connecting semi-intervals. B) Linear event sequence of connected semi-intervals.

graphical representation of overlap is accomplished through using multiple *tracks*, e.g. Figure 3.1D. When events overlap, multiple *tracks* are used. Hence a "music score" representation is used to represent multiple *tracks* of occurring events. it has been called "music score" as the visual representation resembles a score of music with lines for different instruments. An interesting side note related to this is the author of this dissertation is a pianist.

Each *track* can represent a single source of event data, e.g. person A's gaze fixations or person B's speech events. Therefore, each *track* represents a linear source of event information, or *event stream*. Combining multiple *tracks* allows for representation of overlap between multiple event sources allowing the representing of non-linear event sequences or interacting event sources. A collection of tracks is viewed in this dissertation as an *event sequence*. Hence, an *event sequence* is comprised of a set of *event streams*.

Freksa in [39] expanded on Allen's work and redefined the atomic unit of an event as an event's beginning and end, also called semi-intervals. As discussed earlier, semi-intervals allows a flexible representation where partial or incomplete knowledge can be handled since operations are on parts of an interval and not the whole. Finer detail of representation is also possible using semi-intervals as opposed to complete intervals. Plus as demonstrated in

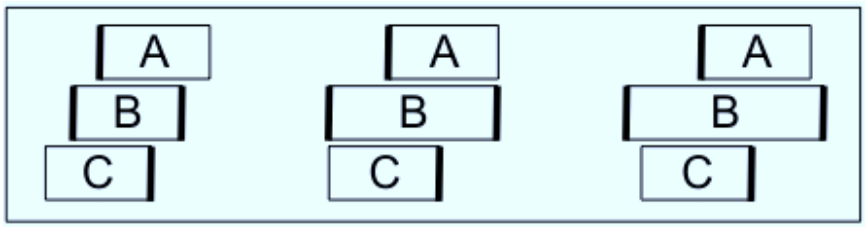


Figure 3.3: Example of one *pattern* defined using semi-intervals (bold lines) is able to capture many cases in the data. Capturing these cases using solely Allen's principles is more complex. Figure courtesy of F. Mörchen and D. Fradkin. Robust mining of time intervals with semi-interval partial order patterns. In *SIAM Conference on Data Mining (SDM)*, 2010. Used with permission from F. Mörchen, 2013.

[101] and seen in Figure 3.3, the use of semi-intervals enables one to capture more cases in the data given a *pattern* comprised of semi-intervals as opposed to complete intervals.

Overall, many works have adopted the music score representation when describing, analyzing, and modeling events. These include [3, 28, 39, 54, 58, 99, 98, 100, 101, 97]. The power in this representation is its simplicity as it can be easily comprehended and modeled.

Such a representation has faired well in expressing inter-related events. A number of multimodal analysis tools used for behavior analysis employ a music score representation for modeling events in time. These tools include [64, 126, 131, 13, 154, 51]. Plus many experts in the field of behavior analysis express behaviors of interest in this fashion [18, 24, 64, 66, 86, 88, 89, 132, 151]. The expression of events in this manner was used successfully in [124] to analyze and discover behavioral phenomena, such as dominance relationships in a meeting room setting. Figure 3.4 and 3.5 are examples of the music score notation being used for event representation in analysis. Here, the analysis consists of an array of *event streams* such as gaze, speech, and gesture. These examples reflect the notable attribute of combining

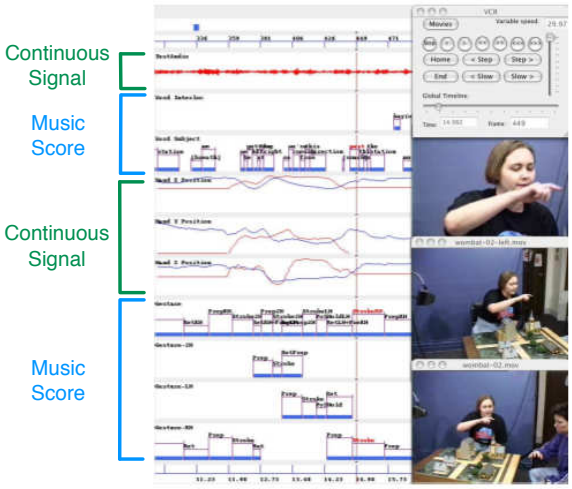


Figure 3.4: Screenshot of MacVisSTA courtesy of R. T. Rose, F. Quek, and Y. Shi. Macvissta: a system for multimodal analysis. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI 04, pages 259264, New York, NY, USA, 2004. ACM. Used with permission from R. T. Rose, 2013.

continuous signal data with that of music score representation of discrete events. Sometimes a continuous signal is needed to aid the analysis and representation of the behavioral data by representing an event sequence in time, as is the case in gesture traces [123, 141]. Continuous signals are sometimes discretized in order to show discrete sections in the signals (episodes), as done in [101].

Due to the simplicity, representational power, and widespread usage of this representation, we have chosen this representation for our approach. It is straightforward to understand, represent, and manipulate. These aspects have allowed a functional foundation of our work to be developed.

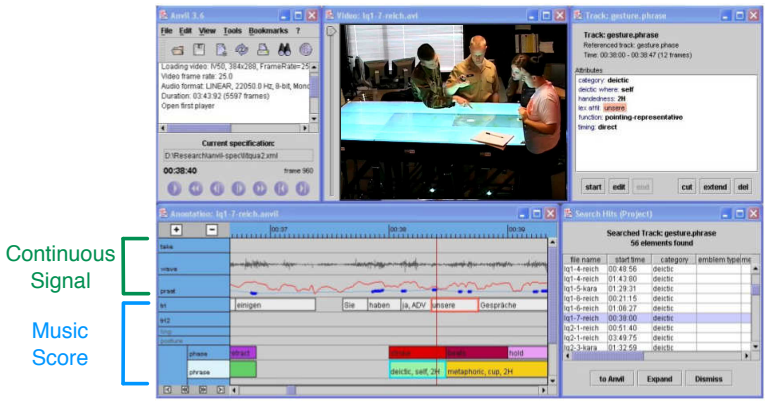


Figure 3.5: Screenshot of ANVIL courtesy of M. Kipp. *Gesture Generation by Imitation From human behavior to computer character animation.* PhD thesis, Saarland University, Saarbruecken, Germany, 2003. Used with permission from M. Kipp, 2013.

Chapter 4

Exploration of a Temporal Event Data-Space

The challenging part of supporting domain experts in behavior analysis is exploration of non-numerical, temporal, descriptive events, e.g., Person A walks up to Person B at time T, and identification and discovery of relevant and meaningful *models* amongst these events. There are a few challenging nuances of human behavior and its analysis to address. First, human behavior is variant. The idea represented by a behavior interaction, e.g., a greeting between two individuals, may be formulated many different ways in the data making *modeling* difficult. How does one identify situated instances of behavior when the way they exist in the data may be unknown? The second challenge is every observed behavior has the potential to be relevant to an expert depending on his/her analysis goal(s). Hence, there is no concept of "noise" but rather one of relevance. For example, consider the situation where three

students are working together on a math problem when a door slams nearby and draws their attention. One expert may analyze the co-construction of space based on the students' aligned gaze while another may analyze interrupting events. The door slam is "noise" to the first expert but not the second. Lastly, a *model's* value to the expert in the analysis of human behavior may not be based on frequency or statistical significance but on subjective relevance. Our approach is designed to address these challenging nuances.

Through our investigations, we have discovered a set of four components necessary to achieve our exploration and supporting analysis goal. *First*, it is necessary when an expert is provided a choice as to which events are meaningful to provide some form of comparison metrics or ranking for events. *Second*, the nature of the behavior interactions being analyzed are situated in time and in context. Hence, support for situated analysis is needed. *Third*, an effective means of searching the events for *models* is required. *Lastly*, an interactive and iterative process incorporating the expert as a guiding factor is key. The following sections present the progression of our work through these four components.

4.1 Event Ranking

The initial investigations of this dissertation began with researching ranking metrics for events. Publication of this piece can be seen here [95] and is included below with minor adjustments. In this early stage of research, we describe the data we were working with as a nominal event data space. The presented work in this section was accomplished using our

prototype system detailed in Section 6.1.

Abstract: This work investigates using n-gram processing and a temporal relation encoding to providing relational information about events extracted from media streams. The event information is temporal and nominal in nature being categorized by a descriptive label or symbolic means and can be difficult to relationally compare and give ranking metrics. Given a parsed sequence of events, relational information pertinent to comparison between events can be obtained through the application of n-grams techniques borrowed from speech processing and temporal relation logic. The procedure is discussed along with results computed using a representative data set characterized by nominal event data.

4.1.1 Introduction

Nominal data is descriptive in nature but difficult to assign any ordering or interval relational structure of one piece of data with respect to another. Typical of such data are those that are conceptually and categorically labeled where any numerical ordering of the data is difficult. For ordinal values, this question is easily solved since the data are numerical in nature or can be relationally ranked or ordered (e.g., high, medium, low). Relational information about nominal data is dependent on the context and the interpreter of the data.

This work focuses on providing a relational understanding between events from media streams. Providing relational insight to events seen in media streams such as audio, video, and motion data is difficult. In the analysis of an interaction session where college seniors

employed a high-resolution tabletop [94] for cooperative analysis in a history class [4], the video and audio recordings of the session were coded for (vocal) referential foci, pointing gestures, and gaze fixation using MacVisSTA [125], a multimodal analysis tool. The analysis of relational information between all the coded events was conducted by humans since the nature of the information of interest is nominal and difficult to process otherwise. However, when only the recorded events are available and a *model* of understanding is desired, building a meaningful *model* may be difficult since ranking one event with respect to another is dependent on a number of factors that are typically dependent on context.

Media streams of data capture a series of these temporal events. A way to approach the construction of such a *model* is to start from a small subset of sequential events and "grow" the *model* one event at a time by incrementally incorporating events of interest. The challenge becomes providing the relational structure of these events that supports addition or removal of events or event substructures to the growing *model*. How do you compare events that are nominal in nature? One possible answer is by probabilistic modeling of the temporal sequence of events. We base our temporal relationship *model* on Allen's temporal relation principles [3]. Many techniques have been created to discover patterns based on Allen's principles in given data sets. This work focuses on providing relational information between events based on these patterns so as to better understand how the events interact.

We propose understanding this interaction better by applying n-gram processing and a temporal relation encoding. N-gram processing is a technique for probabilistic modeling of a sequence [12]. Techniques have been developed to use n-grams for speech processing through

creating probabilistic models based on sequences of phonemes. Temporal event information can be viewed analogously as a sequence of phonemes in which n-gram probabilistic processing may be applied. Section 4.1.2 discusses other approaches relevant to processing temporal event data and nominal event data. Section 4.1.3 describes the proposed approach applied to temporal nominal data. Following, Section 4.1.4 presents an experiment for the provision of how the proposed system works within a data set, after which, conclusions and future work are presented and discussed.

4.1.2 Related Approaches

The ordered relationships that are inherent to temporal event data have been an active area of research. Allen in his work [3] formulated thirteen relationship principles that express all the possible ordering relationships between two events. Since then, much work has been conducted to detect, extract, and represent such temporal information. Examples are [28, 50, 58, 97, 98, 135, 147]. These works represent the trend of discovering events and temporal relationships from temporal data.

Mörchen [99] presents Time Series Knowledge Mining (TSKM) which also is a method to extract and represent temporal knowledge. He further discusses other methods for prediction of events. Such prediction based on temporal patterns is closely related to our proposed work which is more focused on the relationships between events with respect to relational order and history. In [54], prediction is acquired through probabilistic modeling based on

detected temporal patterns. The interest of our proposed work is not predicting *patterns* but understanding the ordered relationships between events, i.e., how do two events relate structurally and contextually.

The authors of [71] report on discovering the relationships between patterns. A number of techniques designed to represent temporal data and discover patterns using unsupervised means are described in [100].

Assigning relational structure to nominal data can be difficult since the meaning can be context dependent and depend on the interpreter and her experiences and expertise. The authors of [74] use a similarity technique in performing unsupervised learning with mixed numeric and nominal data. Through this they were able to cluster the data to provide classification information. This is in contrast to [25] where the authors use dissimilarity to learn classification information for nominal data. Fuzzy logic [158, 159] has also been employed for discriminating between nominal features. However, fuzzy logic requires pre-defined rules that govern how the nominal features relate and hence painstaking consideration is needed before processing can begin.

A number of probabilistic techniques have been employed to describe nominal data. Such techniques include stochastic context-free grammars, Hidden Markov models (HMMs), and class-based n-grams [55, 142, 143]. N-gram processing has been successfully employed as a probabilistic model for speech processing [12, 143]. This kind of processing gives a probabilistic model of successive speech features based on previously seen features, hence providing ordered relational sequence information.

4.1.3 Proposed Approach

The processing of speech phonemes has been successfully done using n-grams for purposes of speech recognition [12]. Such processing defines the probability of one phoneme following others. Then given that k phonemes have occurred, where $k \geq n$, the probability of the next phoneme in the sequence can be given, hence a sequence of phonemes can be built one at a time through probabilistic measures. Initially, phonemes were the basic building block for such processing, but processing has also been conducted with respect to words [12, 55, 109], phrases [43], and word classes [12, 85].

Ordered nominal data within a temporal event stream can be seen as a sequence in which the order gives meaning to the individual parts and the relationships between each piece of data can be defined by this ordering. This is similar to the relational information used in n-gram processing of phonemes. Phonemes within speech have ordered relationships to other phonemes. The n-gram processing of elements of speech provide the relational information necessary for more accurate processing as the previous sequence of phonemes give an idea of what is next. This is possible since n-gram processing is based on conditional probability where $Pr(A|B)$ is the probability of A given B where B is the history and context. For more details on the mathematics of n-grams, the reader is directed to [12].

We would like to use this same processing concept to provide relationship information to temporal event data. Therefore, we would like to answer the question: given a sequence of events, how does one rank events so the next event in the sequence can be wisely chosen?

We propose that given a simple mapping, n-gram processing can be used to give meaningful relational information to aid in answering this question. This mapping can be realized through application of Allen's temporal relation principles [3]. An event can be described by its start time and end time. As done in [58], the relationships between events' start and end times can represent all of Allen's principles hence encompassing all relational ordering possibilities. A simple schema similar to that of [58] can be used to represent a sequence of temporal events in a single linear data stream (Figure 4.1 A).

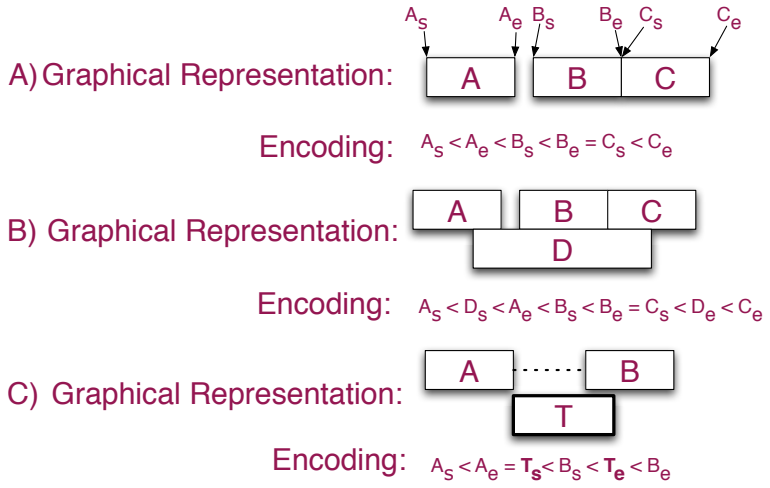


Figure 4.1: A) Example encoding of a sequence of events. A_s and A_e represent the start and end times of event A, respectively. Relational operators are used to indicate the ordered relations between start/end times. B) Example of events with overlap. C) Example of events with a temporal constraint.

It can easily be seen that all of Allen's principles can be represented through such an encoding, including the many forms of overlap (Figure 4.1B). In addition, the relational operators between start/end (S/E) times are able to capture how these times interact. An event A might end before B begins ($<$) or at the same time ($=$). The interpreted meaning of these operators with respect to the involved events depends on the events and their context.

The probabilistic expression of a sequential sequence of events without overlaps is analogous to current n-gram processing of speech. Here events are seen as atomic units where n-gram processing reveals *next/previous* events adjacent to the query. However, with the introduction of overlap, the atomic unit of an event must be broken further down into its S/E times. Hence, the probabilities of *next/previous* events are now with respect to S/E times. For overlap, another level of n-gram processing is needed to catch overlap adjacencies. For example, in Figure 4.1 B, the query for event A, $A_s < A_e$, will give information about B_s as an adjacent S/E time to A. However, the sequence for A is interrupted by D_s. One form of n-gram processing can provide the adjacent events to A and another can provide overlap information, hence capturing D_s and B_s. These different forms of processing are based on the choice of n and performing selective processing on the event stream.

Another relational constraint to consider that is not part of Allen's principles is the relational timing between events. A sequence of events may only be of interest if certain timing constraints between events are held. For example, the sequence of $A_s < A_e < B_s < B_e$ may only be of interest if B starts within a certain time window of A ending. To handle this need, a timing event can be introduced to capture the temporal constraint as seen in Figure 4.1C. A query with a conditional insertion of a timing constraint provides a means to incorporate this constraint information within the context of n-gram processing. Hence, searching for $A < B$ with timing constraint T as in Figure 4.1C would include a query for $A < B$ with a conditional insertion of T after A. The overlap of T and B will signify a satisfied constraint.

With this linear description of temporal events, n-gram processing can be applied similarly

to that of speech processing. The contribution of this approach places emphasis on an applicable ability to give a categorical ranking of data that is normally difficult to do so otherwise. This proposed approach relies on event data extracted from media streams. Earlier mentioned examples are gaze fixations, pointing gestures, and other surface level characteristics of observable behavior. Such examples are within the context of recorded human behavior from various media streams. A description of techniques currently available for said extraction is beyond the scope of this paper but are well studied areas [41, 96, 156].

4.1.4 Experiment

We wrote a library in Python based on the n-gram mathematics described in [12]. The library is initialized with a set of event sequences and queried with a subset of a sequence resulting in probabilistic relationships to potential *previous*/*next* events. This library provides forward and backward probabilistic viewing since a query cannot be assumed to always describe the beginning of a sequence.

To demonstrate this proposed system, a small set of event sequences were extracted from the well understood domain of baseball. A set of play-by-play event sequences were encoded using the above procedure. The play-by-play information was recorded from the publicly available database at `http://www.hokiesports.com/baseball/stats/2009/`.

There are many aspects of a baseball game that can be recorded and represented in a multimedia stream of temporal information: All video and audio of the field and players,

the feedback from the crowd, video and audio of both team dugouts, etc. For simplicity of exemplifying the approach, a set of ten plays consisting of current active players on the field during a play were encoded. These plays represent 83 events, with 48 being unique. The current version implemented provides the probabilities of the *previous*/*next* events with some overlap capabilities, however the support of time constraints is not included at this time. The results also include the associated relational operators with respect to the query.

Two queries representing common events in baseball games were submitted to the database of plays. The results of each query give a probabilistic distribution of *previous*/*next* events where each distribution sums to one. Due to rounding, some of the distributions do not sum to one with an error range of $\pm.02$.

The first query is [B hit ball$_s$ < B hit ball$_e$ = B to 1st$_s$] where B is the batter. This query looks at the beginning of a play and sees what possible events occur after the batter hits the ball and begins to run to first base. The results can be seen in Table 4.1 where R, 1st bm, 2nd bm, and CF correspond to runner(at bat team player on base), first and second basemen, and center fielder, respectively. The results for the *previous* events show this query is only seen at the beginning of a sequence. The probabilistic results for the *next* events show a distribution of event start possibilities consisting of runner behavior on base(s) and opponent players positioning themselves according to the ball behavior. The conditional operator is also given to provide insight as to how the events conditionally relate to the query.

The second query is [SS catch ball$_s$ < SS catch ball$_e$] where SS denotes the short stop. This query is interested in the actions taken by the SS after catching the ball. The results can be

Table 4.1: Results - Batter Hits ball and runs to 1st

	Event	**Probability**	**Operator**
Previous			
	Beginning	1.0	N/A
Next			
	R off $2nd_s$	0.09	=
	R off $1st_s$	0.09	=
	2nd bm to $2nd_s$	0.09	=
	R to $3rd_s$	0.09	=
	R to $2nd_s$	0.09	=
	CF to $ball_s$	0.09	=
	1st bm to $1st_s$	0.18	=
	R to $home_s$	0.18	=
	1st bm catch $ball_s$	0.09	<

Table 4.2: Results - SS catches the ball.

	Event	**Probability**	**Operator**
Previous			
	R off $2nd_s$	0.17	<
	R off $1st_s$	0.17	<
	B to $1st_s$	0.17	<
	B hit $ball_e$	0.17	<
	CF throw to SS_e	0.17	=
	C throw to SS_e	0.17	=
Next			
	End	0.125	N/A
	R to $2nd_s$	0.125	=
	B out_s	0.125	=
	R to $1st_s$	0.125	=
	B to $1st_e$	0.125	=
	R off $1st_e$	0.125	=
	R off $2nd_e$	0.125	=
	SS tags R_s	0.125	=

seen in Table 4.2 where C corresponds to the catcher.

The results of the *previous* events show a probabilistic distribution of events that can explain how the SS potentially came into possession of the ball. The *next* events encompass a batter or runner in the beginning or ending process of acquiring ground, several of which give the SS an opportunity to try and get an opponent out as they are either in transit to a base or have bravely stepped off a base. All the events that began previously to the SS catching the ball appear to end when the SS has possession of the ball.

4.1.5 Conclusion and Future Work

These preliminary results exemplify how combining temporal encodings and n-gram processing results in a strong descriptive measure with which to compare and rank events. Comparison and ranking information is necessary when a *model* is being built from concurrent media streams. The temporal encoding provides ordered relationships between S/E times of events while n-gram processing provides event probabilities based on context and history. Arguably prediction from temporal patterns [54, 99, 100] can be molded for the same probabilistic measures, however doing so essentially results in probabilities based on sequence history which is exactly what n-grams are designed to describe.

Future work includes a further investigation of the relational operator. A *previous*/*next* event might have several different relational occurrences within the database with respect to the query, giving multiple associated relational operators. Inclusion of time constraints

and how n-gram processing behaves for time constraints is also in progress as is how event overlap influences n-gram processing.

Future experiments will include data sets from military exercises and wargame scenarios [24]. The data sets are rich in provided corpus for analysis and have already been partially analyzed providing a known ground truth for comparison.

4.1.6 Acknowledgments

We would like to thank Mark Croushorn for his aid in parsing the baseball plays. This work was partially funded by FODAVA grant CCF-0937133.

4.2 Situated Analysis

Our next investigation was into supporting situated analysis coupled with our comparison metrics described above. The details of what is meant by situated analysis are found in the following subsections. Publication of this piece can be seen here [92] and is included below with minor adjustments. The presented work in this section was accomplished using our prototype system detailed in Section 6.1.

Abstract: Multimodal analysis of human behavior is ultimately situated. The situated context of an instance of a behavior phenomenon informs its analysis. Starting with some initial (user-supplied) descriptive *model* of a phenomenon, accessing and studying instances

in the data that are matches or near matches to the *model* is essential to refine the *model* to account for variations in the phenomenon. This inquiry requires viewing the instances within-context to judge their relevance. In this paper, we propose an automatic processing approach that supports this need for situated analysis in multimodal data. We process events on a semi-interval level to provide detailed temporal ordering of events with respect to instances of a phenomenon. We demonstrate the results of our approach and how it facilitates and allows for situated multimodal analysis.

4.2.1 Introduction

Multimodal analysis of human behavior is ultimately situated. Consider an analysis of several students solving a math problem together with a focus on how the students 'co-construct' space together in the course of their discussion. This co-construction is done through pointing (where spatial reference may be initiated or referenced), gaze (as students co-attend to the shared space that now has meaning), and other gestural forms (that shape the space). The students' gaze, speech, and gesture channels are recorded into a database. During their session, a door slams outside their workspace, at which point all the students turn around. The question is: "Is this very strong, temporally coherent alignment of gaze important?". Looking at the gaze data alone, the strong temporal and spatial alignment may suggest an important event has occurred, but for co-construction of space, it is probably unimportant. The only way to judge relevance is to look at the specific instance.

However, what if the analysis is focused on how an external attention-synchronizing event may alter shared decision making, e.g., clearing out irrelevant distractor concepts when the group returns and picks up where they left off. In this case, the door-slam incident and gaze alignment phenomenon is very relevant. One analyst's data is another analyst's noise. The tension is that we want to have machine help to filter events and focus analysis.

The question is how to solve the problem of filtering and focusing analysis. Such detailed analysis is characterized by focused attention on particular instances of behavior within-context where inter-relations (temporal order) among events in the channels for a particular instance are analyzed. This process is time consuming and very tedious as some form of manual inspection referencing the original media context (e.g., video) is necessary. The situated contexts of individual instances are very important - an instance being a particular occurrence of a phenomenon within the data. We propose a situated analysis approach that supports viewing instances of a phenomenon within-context and automates processing of temporal ordered relationships across channels. Our approach is to allow the analyst to define a *model* of a phenomenon, guided by what is in the data, then, to identify and locate instances of the *model* (and sub-structures) within the data and show the context(s) of the occurrence(s). Typically, the data consists of time intervals describing observed events, e.g. "Student 1 begins a gaze event at Student 2 at time t_i and ends at time t_j", for $i \geq 0$ and $j > i$. Flexible representation of events (and the *model*) are provided through semi-interval processing [39] where a time interval of an event is viewed as comprising of beginning and end atomic units (semi-intervals). The semi-intervals are viewed as base representative units

as opposed to a complete interval. We process these time intervals to extract the necessary information that supports our approach.

Multimodal analysis integrates multiple data channels (e.g., gaze, gesture, and speech) where incorporation of such information provides more accurate analysis results [124]. These data channels are conduits of data streams that contain events describing actions in the data-set. We extract "within-context temporal relations" with respect to the *model* from these data streams. We abstract away from the data streams and focus on the situated events that describe the observed actions. Then for each instance found, the situated events can be viewed, providing the analyst the opportunity to judge the relevance of the instance with respect to their analysis goal and to further investigate other identified instances.

In Section 4.2.2 we provide an overview of our approach. We then review related work in Section 4.2.3. Section 4.2.4 describes multimodal data streams in terms of the events they contain and how we abstract away from streams and focus on viewing the events they contain. Afterwards, Section 4.2.5 describes how our approach extracts temporally ordered relations between events in all data streams and the process of identifying specific instances of a phenomenon. Section 4.2.6 details how we support situated analysis through automated means by combining the components described. Section 4.2.7 follows with a discussion of example analyses using our approach on multimodal human behavioral corpus. Lastly, Section 4.2.8 provides conclusions and discussion of future work.

4.2.2 Approach Overview

Here we describe an overview to our approach that will serve as the foundation to the rest of Section 4.2.

Our approach specifically seeks to incorporate the knowledge and guidance of the domain expert (such as a psycholinguist, or other behavioral expert) in human behavior analysis tasks. Conversely, we want automatic processing to support domain experts as they apply their knowledge in exploring multimodal data. Other attempts at providing automation support for analysis include [141] and [21]. As *models* of phenomena are created and discovered (through exploration and refinement), the identification of other instances of the *models* is needed. Our approach is to provide automatic processing of temporal relations between data stream events and view such relations situated (within-context) within instances of a phenomenon.

We provide support in three ways. *First*, we view the analysis process as beginning with the domain expert advancing a hypothesis (*model*) of a behavior phenomenon and proceeding to explore instance(s) of the phenomenon (point of focused attention). How the phenomenon is present in the data may be unknown at the start of analysis, hence, the expert starts to explore the data with an initial idea. This initial idea is likely to be an incomplete *model* but a general structure of the phenomenon. *Second*, our approach can identify instances of this *model* in the data and present *related* events to each specific instance. *Related* events refer to events found in context to a specific instance and their respective temporal relations to the

instance. The *related* events presented are from all data streams enabling situated analysis as we extract within-context information of a domain expert's *model*. *Third*, all instances of the *model* are identified, allowing a comparison of relations across contexts which may help in formulating new insights. With these three components, our approach allows a situated view of cross-stream relations with respect to a phenomenon.

4.2.3 Background and Related Work

Temporal Event Data: The ordered relations that are inherent to temporal event data is an active area of research. Allen in [3] formulated thirteen relationship principles that express all the possible orderings between two event intervals. These orderings describe how events between data streams relate. Research, such as [50, 58, 97, 98, 135], has focused on processing numerical or descriptive univariate, multivariate, or symbolic temporal data with the goal of discovering events and temporal relationships from temporal data. Others have explored the discovery of temporal knowledge and associated patterns, such as [54, 71, 100].

The stream data of multimodal data-sets is a mixture of ordinal and nominal data. As per the temporal data models reported in [100], gesture traces are a collection of univariate numerical time series (ordinal) while speech and gaze target fixations are multivariate symbolic interval series in which their meaning is categorically descriptive (nominal). Explorations into the relational structures of nominal data can be seen in [25, 74, 158, 159]. More specifically, our prior research [95] explores how to structure nominal event data based on semi-interval

processing using n-gram techniques from speech processing. Other research [39, 101] has explored the concept of breaking temporal events into core units of beginnings and endings through semi-interval temporal ordered processing. Previously mentioned research ([3, 58, 98, 135]) has investigated patterns and relations among interval data irrespective of the data being ordinal or nominal.

Multimodal Analysis: Multimodal analysis of human behavior typically involves multiple time-synchronized data streams which consist of different observable human behavior signals (e.g. different gesture types, gaze fixations, and speech). The streams are then annotated either by hand, automatically, or a mixture of both. Then manual analysis, normally with some machine automated help, of the streams is performed. Example application and results of this process can be seen in [21, 24, 63, 66, 89, 107, 123, 155].

There has been much work in automatically detecting and recognizing certain behavior, a few examples being [44, 102, 151]. However, it is the more detailed analysis prominently exemplified by the math students example that is much more difficult to automate and is the kind of analysis where human guidance is needed. The goal of this detailed process is to explore the data and to identify and understand instances of phenomena. This is done through inspection of the temporal relations of the events within the streams. However, several challenges surface in this process. The identification of instances of a phenomenon that are relevant to the analysis being conducted can be difficult. Plus, coordination between the multiple streams requires careful attention as seen in [21, 124]. Both of these challenges were exemplified in the previously described analysis scenario of math students.

In terms of the analysis focus, we have observed two viewpoints employed for multimodal analysis of human behavior. The first focuses on identifying behavior instances within sections of recorded data (e.g. recorded video/audio sessions). This is a useful and powerful means as it shows the context in which the behavior occurs. However, what if an expert is interested in identifying instances of a particular behavior (e.g., co-construction of space) and viewing all the contexts in which they are situated. The second viewpoint is the reverse of the first by showing the behavior within-context. This is a plausible approach since the focus of some analyses is instances of a specific behavior. Examples of this can be seen in [21] where cues were identified within a group meeting setting that could signal floor control events. Other research [123] has found cues in gaze and gestures that aid in discourse analysis. These cues can be flags in the identification of specific behavior of interest for which the expert would want to view the different instances within their respected situated contexts. This prior research exemplifies that the difficulty is in identifying how phenomena are manifested within the corpus. By taking the viewpoint of behavior within-context, we can begin with a general structure (*model*), or sub-structure, that describes the phenomenon and see how events within the corpus are *related* to the structure.

4.2.4 Multimodal Data to Events

The first component of our approach has the expert start with a hypothesis with which to identify instances. Hence, we need a way to describe the multimodal data to allow for defining a *model* of the hypothesis. We want to abstract away from data streams and view

the descriptive events. In this section we define data streams and the events they produce and how this desired abstraction can be obtained.

Multimodal data typically exist in the form of data streams (time series of data points and signals) and music score data. Music score data are multivariate time series data that describe event intervals in time that possibly overlap. A number of multimodal analysis tools (e.g., [64, 126]) employ this form of multimodal data representation. Our first step is to convert these data streams into a homogeneous stream of events. An event is an interval in time representing when an action begins and ends. It comprises of a beginning, end, and description. Figure 4.2A shows three examples of events, A, B, and C, with their respective beginning (start) and end times. Viewing events with respect to their beginning and end is known as semi-interval processing [39] where a beginning or end is a semi-interval. Sometimes it is difficult to determine what is an event within a data stream (e.g., deciding exactly when an event begins and ends) however, we assume such detection occurs prior to our processing. Our focus in this paper is not on how to extract events from data, but on how to interpret these events after they are extracted.

Multimodal analysis comprises multiple data streams by definition. Hence, there will be overlaps between the multiple streams and cross-modal relations supportive of multimodal analysis. In Figure 4.2B, we see three data streams, Student 1's speech, Student 2's gaze, and Student 3's gestures, with respective events. It can easily be observed how these represented events occur in overlap.

We abstract away from various specific representations in the separate data streams and

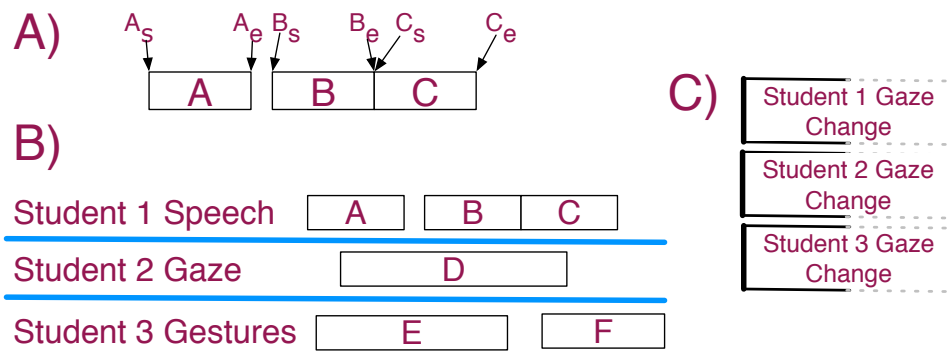


Figure 4.2: A) Example of a sequence of events with their labeled start and end times, respectively. B) Example of event overlap between streams. C) Example of an attention-synchronizing *model* with concurrent change of student's gaze.

focus on the events they contain while keeping the context (data stream) from which they originate. The strict notion of data streams fades into the background as the events become a prevalent part of the analysis. It is the events and what they represent that aid in the analysis. This approach allows the flexibility of processing that is mode independent where the kind and number of modes used does not matter.

Figure 4.2C shows an example *model* of a hypothesis an expert might use to find instances of attention-synchronizing events during the math student session described earlier. Assuming three students are involved, the expert specifies a *model* where the student's gaze change at the same time. This involves specifying the temporal order of start gaze change semi-intervals for each student. In this case, the order represents concurrent action (equality). This will identify instances where the students have a synchronized gaze change. This example shows how we abstract away from the idea of three gaze data streams and focus on the gaze events of the students.

4.2.5 *Model* Aspects

Following from our first component, temporal relations between the *model* and *related* events is needed. The coordination between events within multiple data streams is necessary to understand a specific *model* instance. The temporal order in which events occur is key in describing a specific phenomenon, e.g., the coordinated gaze in response to a door-slam describes an attention-synchronizing phenomenon. This leads to the need of our second component which is identifying an instance of a *model* and extracting within-context temporal information between the *model* and *related* events. First we will address extracting the temporal information, after which, we will describe the process for identifying a model *model*.

Event Temporal Relations

The challenge is temporal relations describing phenomena are structural in nature and not a set of parameters. Such a structure must be learned. Consider the scenario where a group of four History students use a horizontal, high-resolution, interactive tabletop display [94] to perform analysis of information snippets on virtual notecards (displayed on the table) [4]. The students were not given any instructions for making sense of the data, however, over time their actions around the shared space evolved into a ratification process in which changes to the notecard placement and utilization of the display space could be made. This process begins with the wielder of the interaction device announcing a statement or question

around a piece of data (*announcement*) which lead to the coordinated gaze of the other three students (*co-gaze*). The wielder and other student(s) then proceeded to discuss a piece of data and decide (*ratify*) what to do with it.

The students' interaction resulted in a sequence of proposal, echo, and act with enunciation that produced a ratification process which advanced the joint project and common ground. This temporal sequence exemplifies the structural nature of temporal relations in multimodal data. Other approaches, such as HMMs [128] and n-grams [12, 138], can be viewed as piece-wise parametric *models* that get some sense of overall structural relations. The problem is that approaches such as these 'linearize' behavior structure of these piece-wise *models* and do not address how to combine these linearized pieces structurally. Research in the numerical domain has also investigated structural means to numerical model modification, for example [27, 59, 80].

We approach representing temporal relations structurally by viewing events from the semi-interval level and how such atomic units of events relate temporally. Other research has investigated the use of semi-intervals for its representational flexibility with respect to incomplete or partial knowledge in identifying event interval sequences of interest [101]. Our prior work [95] investigated the use of n-gram processing of semi-intervals for providing comparison means for events in event sequences in which the processing takes advantage of the structure of the events.

With this structural approach, we provide automatically extracted temporally ordered relations within and between streams. This extracted information includes how semi-intervals

relate with respect to order (e.g., "<", and "=") and their likelihood with respect to a *model*. Given a *model* consisting of semi-intervals, *related* events to each semi-interval are classified occurring either *before*, *current* (concurrent), or *after*. This is the same classification as used in [95] with the addition of *current*.

To obtain this information we first segment all interval events into semi-intervals. Next, all the events from all data streams are serialized into one linear encoding with temporal ordering preserved through the use of ordered relations. An example of this can be seen in Figure 4.3 where the events from 4.2A and B are linearized. Due to this serialization process, results will contain temporal ordered relations within and between data streams as seen in Figure 4.3B. Here, the semi-interval events from multiple data streams are integrated into one sequence allowing comparison across streams and handling of overlap. Hence, given a focal point (*model*) in the data, one is able to view events across streams that *relate* to the focal point within-context. Semi-intervals that co-occur are set to be equal ('=') preserving their co-occurrence (e.g. B_e and C_s in Figure 4.3).

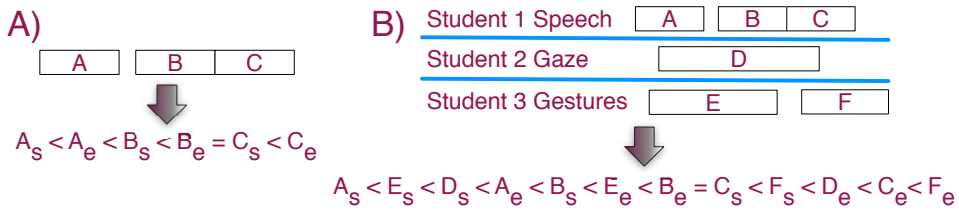


Figure 4.3: A) Example of linearizing events and preserving their temporally ordered relations. B) Example of how serialization of multiple streams allows for comparison across them.

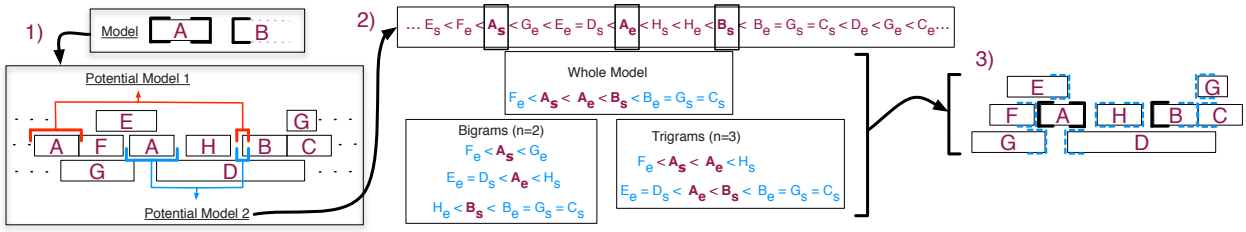


Figure 4.4: Localization of *model* instances and subsequence processing of temporal relations: 1) Begin with a *model* and identify where potential instance(s) reside at the semi-interval level. 2) Choose instance and perform processing on the whole *model* and select values of N. 3) Present *related* semi-intervals.

model Instance Identification

The process of how we identify an instance of a *model* and subsequently provide *related* event semi-intervals to the *model* can be seen in Figure 4.4. To identify an instance given a *model* (in terms of semi-intervals), we search the data-set for matches to the semi-intervals (1). In this step, multiple matches may be found as can be seen from the two potential *models*. This occurs because events may repeat. A first approximation may be to assume that the closer of two events of the same type is the better candidate. Hence, we search for the match where all the semi-intervals are temporally closest. This mainly applies to semi-intervals of a different event type, e.g., A_e and B_s or a gaze event versus a speech event. Hence, Potential *Model* 2 is chosen. We also process sub-structures (sections) of the *model* to detect overlap, e.g., D_s overlaps with A. Note that identified instances may have semi-intervals that "interrupt" the instance, e.g. event H occurs between A and B. After identification is complete, the *related* semi-intervals to the *model* are extracted using n-gram processing at the semi-interval level (2). The set of *related* semi-intervals are then presented as relating to the *model* (3). This set also includes events that may only relate to a substructure of the phenomenon but are

part of the same instance. This allows for identification of overlap and semi-intervals not seen by viewing the whole *model* solely, as seen in Figure 4.4 step (2).

4.2.6 Assisted Situated Analysis

The identification of an instance of the hypothesis (*model*) allows a within-context view of *related* events. However, the expert is interested in all instances of the hypothesis, leading to a need to view the multiple instances of the hypothesis within context and, potentially, to compare them. In this section, we address our third component through initial discussion of identifying all instances of a phenomenon, how an expert can toggle between them, and the resulting aggregate view of all the results. After which, we discuss how our approach facilitates comparison across contexts.

To identify all instances of a phenomenon, the process described in Figure 4.4 is repeated until all instances have been identified. The expert can then step through all unique instances, observing the *related* events and the differences and similarities. Through this process we can support analysis of the interrelationships among events with respect to the *model* wherever it occurs in situ, hence providing relations within-context and across data streams. We call this assisted situated analysis as we use automation to identify instances and the associated temporal relations for each instance and provide support to the expert in viewing and exploring this information.

We also allow the expert to view the *related* events from all contexts at once. An example

of this can been seen in Figure 4.5 where a *model* and sub-structures are identified in three contexts and the *related* events within each context are aggregated to allow a view of all *related* events. With the aggregation of all results, the likelihood that the events occurred given all the contexts is provided when comparing the *related* events. The likelihood calculation is with respect to the whole *model* and its sub-structures and follows the same procedure of our prior work [95]. This aids in comparison among the instances extracted and also provides information on the event frequency with respect to the *model*. The number of semi-intervals used in calculating the likelihood is given as a confidence level as it is a measure of the amount of information used in the calculation. We are more confident in a result that incorporated more information.

An interesting artifact of this process can be seen in Figure 4.5 where the frequency of G may suggest an event of interest. However, the frequency is only seen through comparing multiple contexts. The aggregation of the results from all instances of the *model* and sub-structures allows for the opportunity to discover new temporal relations between events and the *model*. Alternatively, H occurs only once but may hold equally or more important value than G's frequency (frequent vs. infrequent). Presenting the results from all contexts provides the opportunities for the researcher to investigate variations of the *model* (through inspection of their original context) and decide their relevance to the current analysis. Hence, given the above procedure, we are able to provide assisted support to multimodal situated analysis.

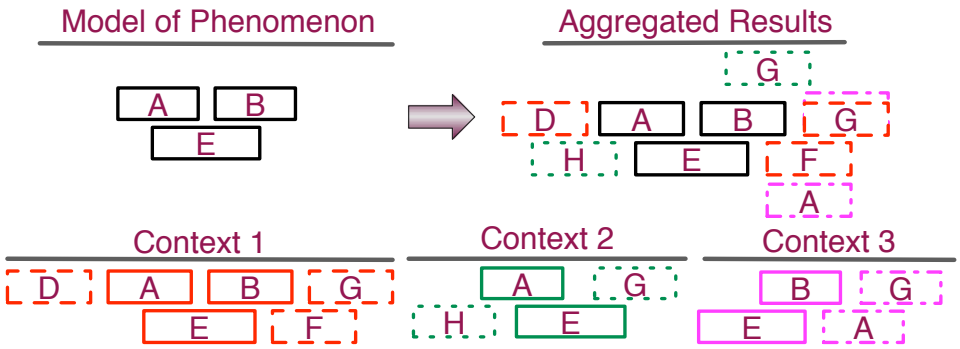


Figure 4.5: Identification of a *model* and sub-structures in several contexts and the aggregation of *related* events across contexts. Overlap of G after *model* event B represents its multiple occurrence.

4.2.7 Implementation and Use

Here we present example analyses that illustrate how our approach is able to provide support for multimodal situated analysis. A real data-set was chosen to exemplify our approach. The data-set consists of meetings of U.S. Air Force officers from the Air Force Institute of Technology (AFIT) engaging in military wargaming scenarios [24]. In this section we discuss the implementation of our approach, the data-set used, and example analyses.

Implementation

The core algorithm of our approach is based on the system described in [95] with a number of updates and enhancements. An interactive layer was added to allow user interaction and viewing of the context of any specific *model* instance and its respective *related* events. Implementation is in Python using QT 4.5 [118] for the user interface, and the Pymunk physics engine [117] (a Python wrapper of the C library Chipmunk [26]) is used as a means of collision detection for creating the spatial layout of the graphical semi-interval instances.

Although it is necessary to show our user interface to explain our results, the interface is not the focus of the current paper. Our focus is on the event processing.

Demonstration Domain

We demonstrate key aspects of our approach with a comparison of an original manual analysis of the AFIT data. There are multiple days of data recordings that have been carefully hand annotated by behavior domain experts. We discuss a study focused on exploring one session in which the officers (labeled C, D, E, F, and G) are discussing potential candidates for a scholarship. The scenario is that C, D, F, and G are department heads meeting with the institute commandant E to select three scholarship award recipients. It was discovered that in such meetings, the social interaction among the participants have as much to do with the outcome of the meeting as the specific merits of the scholarship candidates being discussed. The participants dynamically formed coalitions to support each-other's candidates through a process of mutual gaze fixations and back-channel expressions of support during discussions [89].

This session is approximately 45 minutes long. A coalition to support a proposed idea is initialized when the proposer of the idea seeks to make eye-contact with other participants while he is presenting the idea. Participants who supported the idea would return the eye-contact, while those who disagreed with the idea would avert gaze. When a return gaze is attained, the presenter's gaze moves to another member. This phenomena was recorded within this scholarship session and we want to compare the results of our extraction methods

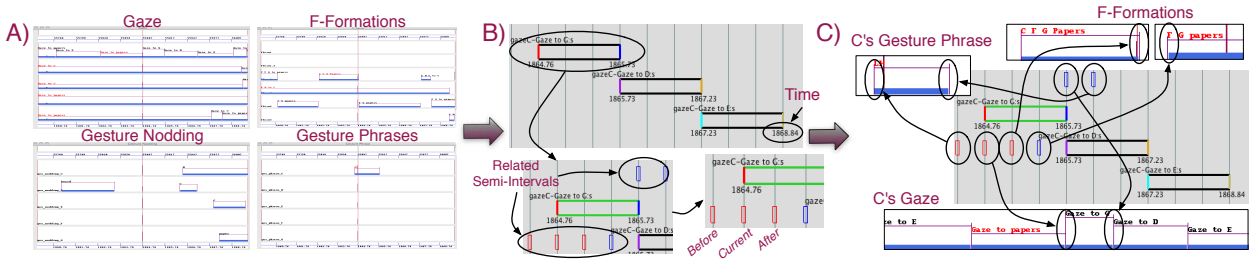


Figure 4.6: A) Music score event information for AFIT session. Vertical line is the current time. B) *Model* constructed of C's gaze sequence. C's gaze to G is highlighted showing all the *related* semi-intervals to the instance. C) Highlight of semi-intervals in the original data that are reported as *related* to the instance.

against this real world recorded scenario. This scenario describes C's sequence of gaze events starting with one to G, then D, then E. During this sequence, D and E's gaze is fixated on C.

There are three key areas we want to demonstrate. The first is our ability to automate the processing of cross-modal relations. Given a snapshot of a real analysis scenario, we want to compare our automatically extracted relation information with the relation information employed in the original analysis. Second, closely tied to the first, we would like to show how our approach is able to report *related* events that are within-context to a respective *model*. Then, switch between the different identified contexts to view different *related* events. Lastly, we will show how we can view the *related* events of all contexts at the same time allowing an aggregate view. Through this we will show how we can discover variations not seen in the original analysis.

Analyses Examples

Single Context Cross-modal Relations: First, we demonstrate the ability to automate processing of cross-modal relations. The emphasis in the original analysis was observing the gaze behavior. However, there are many other data channels available of which our approach also handles. One of the difficulties for this detailed data-set is there are many channels of information (approximately 37). Viewing the relevant information across all the streams within these channels is a challenge of wading through visual clutter. For this example (and subsequent ones) we use 22 channels consisting of F-formations, gaze fixations, nodding gestures, and gesture phrases. F-Formations, or focus formations, were first identified by Adam Kendon to be units of topical cohesion marked by gaze cohesion of participants to common objects or spaces [60]. Gesture phrases are atomic gestural motions marking a single motion trajectory that typically coincide with atomic speech phrases [87]. The data streams used numbered 6,500+ semi-intervals.

Table 4.3: Results for start of AFIT gaze *model*

Context #	Model semi-interval: C gaze to G:s		
	Before	Current	After
1	C's ges phrase LH:s	C gaze to paper:e	F-Formation C F G to papers:e
2	C gaze to papers:e C gaze to E:s	C gaze to E:e	E gaze to papers:e E gaze to D:s
3	E gaze to G:e E gaze to papers:s	C gaze to E:e	C ges phrase LH:s
4	G gaze to D:e G gaze to papers:s	C gaze to F:e	F gaze to papers:e F gaze to D:s
5	F Gaze to C:e F gaze to papers:s	C gaze to papers:e	E gaze to C:e E gaze to papers:s

The *model* for this example details the order of C's gaze events to G, then D, and then E

where each event has a start and end semi-interval. For our demonstration, we use the same situated instance from the original analysis. Figure 4.6 illustrates the connection between the annotation data and the results of our approach. In (A) is a visualization of the event data in the multimodal analysis tool MacVisSTA [126] (used for the original analysis). (B) graphically shows the *model* constructed using semi-intervals: [C gaze to G:s < C gaze to G:e = C gaze to D:s < C gaze to D:e = C gaze to E:s < C gaze to E:e], where ':s' or ':e' represents the start or end semi-interval, respectively. The first event, C's gaze to G, is highlighted showing three sets of *related* events for each semi-interval, each set corresponding to events that occurred *before*, *current* to, or *after* the semi-interval. (C) highlights how the *related* events link back to the multimodal data from (A). A detailed sub-set of results can be seen in Table 4.3, context 1. The table shows the results for one of the six semi-intervals of the *model* and exemplifies the information provided to the analyst. The reported events are the *related* event semi-intervals seen amongst the 22 data streams, allowing a quick view of these events with respect to the instance. These reported events are indeed events seen holding relation to the instance used (context 1) for the original analysis. With this information, an analyst can judge if the instance is deemed interesting enough to jump to the video and view the instance further. Also, identifying semi-intervals that adhere to equality ('=') seen in the *model* is not strictly upheld to allow for more flexibility in instance identification.

Multiple Context Cross-modal Relations: Our second example is motivated by judging the relevance of an instance. Here we show the viewing of and switching between results for multiple contexts. There were approximately 14 instances of C's gaze sequence *model*. To

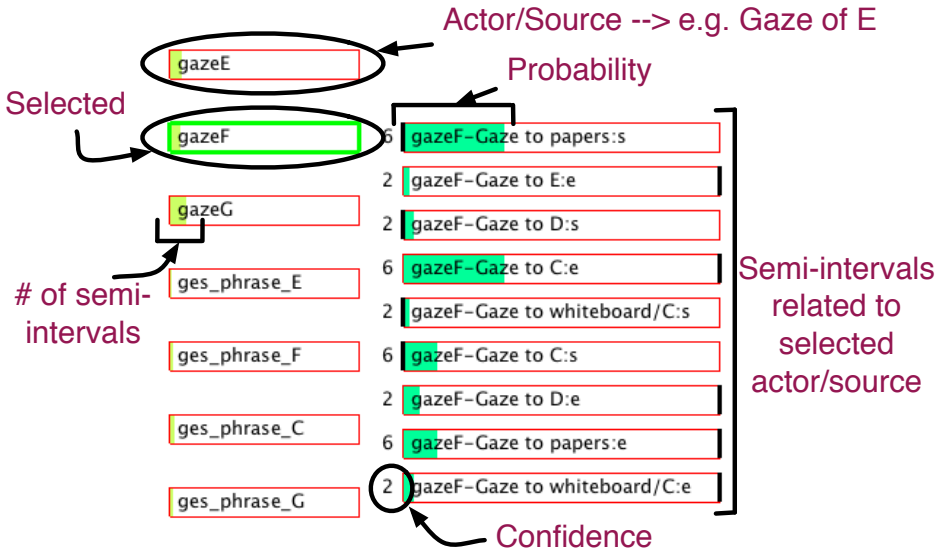


Figure 4.7: Example of *before* aggregated *related* results for the C gaze to G:s semi-interval.

exemplify the varying contexts, we list results from five of these in Table 4.3. By switching between contexts, it can be seen that the *related* events vary providing interesting variations of context. Each context can be used to query the video content and allow a more detailed look. Switching between the contexts can show the differing events that occur and may aid the analyst in judging the relevance between multiple instances. Some results contain more than one *related* semi-interval, e.g. context 2, *Before*. This shows when either multiple events started and/or ended at the same time marking a transition between events. For the case of context 2, *Before*, C's gaze to the papers ends then C's gaze changes to E.

Aggregated Results: Our third and last example takes the results from all contexts and aggregates them into one view to allow statistical comparison among the contexts and identification of variations. Here, instead of viewing one set of results at a time (e.g. one row of Table 4.3), we can view all sets (e.g. all rows) together. This combines the results into one

view where the likelihood of each *related* event is presented as a comparison metric allowing a statistical comparison view of all *related* events across all contexts. One powerful aspect of this approach is that results from both complete and partial matches to the *model* are presented together. This allows the analyst to view instances that have some commonality to that of the *model* of interest and may prove informative in the analysis as illustrated in Figure 4.5 and related discussion.

An example of this aggregation can be seen in Figure 4.7 where the presented results correlate to a sub-set of the *before* results for C gaze to G:s. The first column (left) represents the actor/source of an event with the percentage fill of the representative graphical rectangle being the total number of semi-intervals associated with the respective actor/source. When one actor/source is selected, a second column is displayed which illustrates all the semi-intervals *related* to the actor/source that occur *before* C's gaze to G:s. This second column displays each semi-interval's description and associated conditional probability and confidence. The probability is mapped to the percentage fill of the graphical rectangle where more fill equals a greater probability. A bold bar is displayed on either end of the graphical rectangle depicting whether the semi-interval is a start (left side) or end (right side).

Given this aggregated view of *related* events, we can identify variations of *related* events to the *model*. A variation is described as a sequence of semi-intervals not recorded in the original analysis or that differs from the original *model* of interest. The reasoning for these variations was exemplified in Figure 4.5 and related discussion. In this example, three variations were identified that were of interest. The first stemmed from the question of if G returns C's

gaze at some point. This is a valid point of interest as such gaze exchange is an important indicator for coalition building, as mentioned earlier. Upon inspection of *related* events in the aggregate view of C's gaze sequence detailed earlier, we see a gaze return from G during C's gaze to G and ending while C's gaze was on D. The resulting *model* of this variation (variation in bold) is: [C gaze to G:s < **G gaze to C:s** < C gaze to G:e = C gaze to D:s < **G gaze to C:e** < C gaze to D:e = C gaze to E:s < C gaze to E:e]. Using this modified *model*, we performed another query of the data-set and found two instances of this *model*. At this point the analyst can view the *related* events of those instances and jump to their situated context in the video data.

Next, we looked to see if D returned C's gaze. Similar to the prior case, D's return gaze begins during C's gaze to D and ends afterwards. This is in contrast to the original analysis where D's gaze was already fixated on C prior to the start of C's gaze sequence. Using this new *model*, we performed a query and found partial matches to the variation. No instances of this *model* variation were seen in the data-set. The only "instance" seen was matches found through identifying sub-structures of the *model* within the data (as seen in Figure 4.5). The resulting matches were very close to that of the desired variation leading to the identification of a variation not originally conceived in the data. This variation represents a new structure of how a coalition can be constructed with which such knowledge can be applied to other data-sets (e.g. the other AFIT sessions).

The detection of a coalition can be strengthened through using other data streams of interaction such as acknowledgement gestures in response to gaze. Hence, we identified another

variation where G responds to C's gaze with a gesture nod (absent of G's gaze fixation). However, no instances of this *model* variation were seen in the data-set. The only "instance" seen was found through identifying sub-structures (same as the previous variation). Hence, this variation represents another *model* structure of a potential coalition that can be used in searching other data-sets.

Discussion

Given a set of events describing multimodal behavior of multiple individuals, we were able to identify events that were potentially relevant to particular instances of a phenomenon *model*. In doing so, we support expert exploration of large corpora of multimodal behavior to understand and extend *behavioral models*. Our approach is situated in that the expert works with particular instances of the behavior in the data and is able to inspect the original video/audio record of the behavior. However, our approach transcends the confines of the current instance in that it reveals other occurrences of the behavior being studied in the data. It allows the user to explore these occurrences within their contexts to determine if recurring event *patterns* are indeed relevant to the behavior being studied. Hence, this situated analysis also allows the comparison of phenomenon instances across different contexts.

The aggregated view of results adds an extra level to situated analysis as it provides a means to view all the results to all instances at once. This allowed identification of variations not originally conceived, providing opportunities for new thought constructions of *models* to be used in future analyses.

4.2.8 Conclusion and Future Work

We were able to facilitate the situated analysis of multimodal corpus of human behavior. The expert is able to view *related* events of different instances of a phenomenon, view the cross-stream relations, compare across instances, and be assisted in the discovery of variations of phenomenon. We were able to illustrate how our approach is a beneficial aid in analysis through application to a data-set with known ground truth.

Through this investigation of situated analysis, we see areas where our processing can be improved to provide advantageous benefits to the analyst. The first is expanding the processing of our results to include more than boundary relations to events. Currently, our processing reports *related* semi-intervals with respect to the boundary (start/end) of events. Other equally important information is events occurring during a *model* but their start or end semi-interval is not temporally close. These events are on-going during instance(s) of the *model*. We are also interested in allowing more flexibility in identifying instances of a *model* with respect to how the order of *model* semi-intervals are realized in the data-set. For example, the hypothesis representation in Figure 4.2B may not surface in the data as all the gaze semi-intervals occurring at the exact same time, but very close to each other. This exemplifies the idea of temporal constraints, an area of research for this kind of processing and analysis discussed in [95]. Also, semi-intervals occurring in a slightly different order could pose a beneficial instance match, hence flexibility for this would be beneficial.

4.2.9 Acknowledgments

This research was partially funded by FODAVA grant CCF-0937133, and by NSF IIS-1053039 "Multimodal Corpus for Vision-Based Meeting Analysis".

4.3 Search Strategy

Our third component is the need to effectively search and identify *models* in event data. Our investigations into a search strategy uncovered an area in multimodal analysis that had received little attention: how does one search multimodal data for *behavior models* based on the structural and temporal characteristics of such *patterns*? Publication of this piece can be seen here [91] and is included below with minor adjustments. Due to the intended audience of the publication, "*pattern*" is predominantly used instead of "*model*". Subsequently, a more detailed definition of a *pattern* is presented. The presented work in this section was accomplished using *version 1* of our system detailed in Section 6.2.

Abstract: There are a multitude of annotated behavior corpora (manual and automatic annotations) available as research expands in multimodal analysis of human behavior. Despite the rich representations within these datasets, search strategies are limited with respect to the advanced representations and complex structures describing human interaction sequences. The relationships amongst human interactions are structural in nature. Hence, we present Structural and Temporal Inference Search (STIS) to support search for relevant

patterns within a multimodal corpus based on the structural and temporal nature of human interactions. The user defines the structure of a behavior of interest driving a search focused on the characteristics of the structure. Occurrences of the structure are returned. We compare against two pattern mining algorithms purposed for pattern identification amongst sequences of symbolic data (e.g., sequence of events such as behavior interactions). The results are promising as STIS performs well with several datasets.

4.3.1 Introduction

There is a multitude of annotated behavior corpora (manual and automatic annotations) available as research expands in multimodal analysis of human behavior. Many of these corpora and supporting visualization tools store, organize, and display multimodal data based on the structural nature of behavior. By structure we mean discrete events that hold ordered relations in time that may vary between occurrences. For example, the visualization tools MacVisSTA [126], ANVIL [64], and EXMarALDa [133] display multimodal data as interval events with support for continuous signal data. The input formats of these tools are based on discrete interval events (multivariate symbolic data). This organization strategy is also seen in domains where frequent episode mining [114, 115, 130] is applied (e.g., medical records, neural spike data,...). Frequent episode mining is normally based on identifying a sequence of atoms (e.g., symbols or descriptions) and identification of "relevant" patterns is based on frequency and/or statistical modeling. However, for analysis of multimodal data, a pattern's value to an expert may not be based on frequency or statistical significance but on

subjective relevance. Hence, a search strategy designed for an expert's interest in multimodal behavior data is motivated.

We present Structural and Temporal Inference Search (STI-S), a pattern search strategy for multimodal data built upon the structural nature of human behavior. A pattern defines a sequence of behaviors. Behaviors are encoded as annotated event intervals with temporal order being implicitly or explicitly defined. An example is a greeting among two individuals with the possible formulation: <A walks up to B>[within 1 second]<A shakes B's hand> and <A says "Hello">. We base STIS upon this representation using contextualized information. This is done through viewing a pattern that is of interest to an expert (i.e., a relevant pattern) as not only the focus point of analysis but also defining the search criteria. A pattern is deemed relevant by an expert dependent on the expert's interest in the behavior described by the pattern. Identification of a relevant pattern within a dataset (i.e., search) results in occurrences of the pattern.

The expert's definition of a relevant pattern incorporates his or her knowledge into the search criteria as opposed to relying on statistical modeling to bring to the surface a pattern that may or may not be of interest. Statistical models used to extract frequent and/or statistically significant patterns (episodes) [72, 115] do not address cases where a pattern may only occur a handful of times. As discussed in [93], an algorithm's results based on some automated metric (such as frequent episode mining) would require some form of explicit pattern search anyway. This motivates our interest in identifying pattern occurrences of interest to the expert.

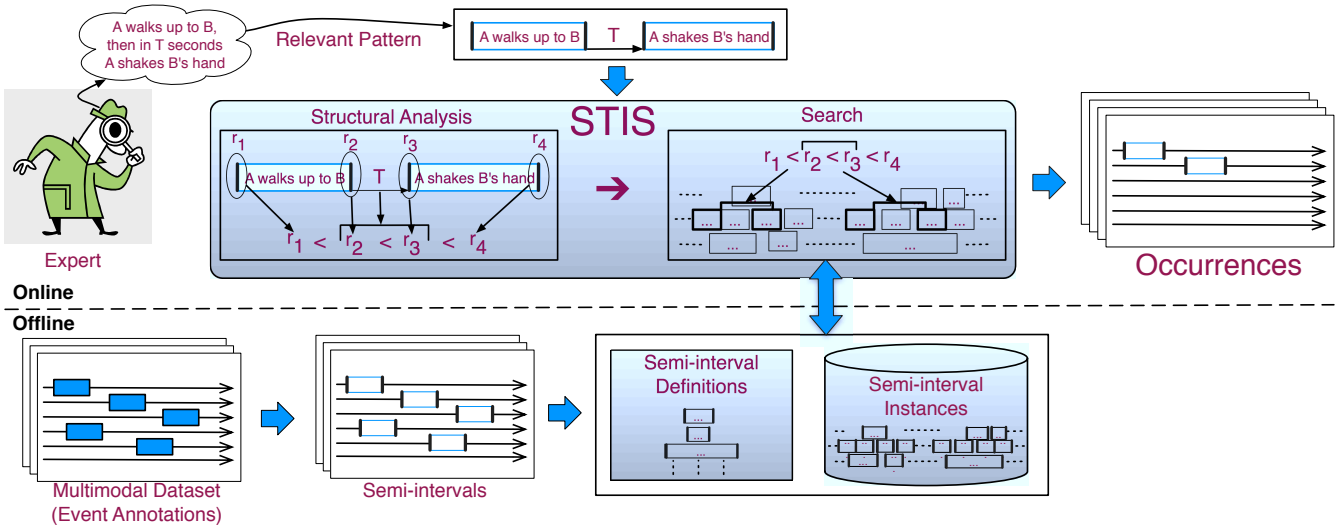


Figure 4.8: STIS overview: Offline, from a multimodal dataset create a semi-interval set organized as *definition* and *instance* tables. Online, an expert provides an event sequence that is converted into a *pattern* containing implicit search criteria. STIS performs structural analysis on the *pattern*, uses the results of the analysis to form search criteria, searches to identify occurrences based on the criteria and returns a set of occurrences.

The rest of our paper is organized as followed. In section Section 4.4.3 we present related work of multimodal corpora, analysis tools, visualizations, and temporal pattern mining approaches. The details of STIS are discussed in Section 4.3.3. Section 4.4.5 discusses our experiment methodology, implementation details, the datasets used, the baseline algorithms, the patterns tested, our results, and discussion. Conclusions and future work close our paper in Section 4.3.5.

4.3.2 Related Work

Our data domain is multimodal. There has been a strong trend toward creation and analysis of multimodal corpora. This is no surprise as the authors of [124] argue the value and

deeper understanding multi-modality adds to analysis of human behavior. Many multimodal corpora have been created in response to this observation which predominantly consist of sequences of descriptive events (behavior patterns). The VACE/AFIT [24] multimodal meeting corpus is a detailed recording of multiple sessions of Airforce officers partaking in war gaming scenarios in a meeting setting. The Semaine corpus [86] is a collection of emotionally colored conversations. The Rapport and Face-to-Face corpora [46, 103] are sets of speaker-listener interactions. One of the largest to date is the AMI corpus [18] which contains 100 hours of recorded meetings. Mörchen created a series of datasets of varying degrees of modalities [101]. These mentioned corpora and datasets are highlights of a growing community of such data.

With the increasing number of multimodal datasets, tools are needed to visualize the data for analysis. These tools have been developed to visualize multi-channel annotation information coupled with varying degrees of multi-channel support of audio and video. Well known examples of these tools are MacVisSTA [126], ANVIL [64], Theme [146], EXMARaLDA [131, 34], ELAN [154], C-BAS[1], Transformer, and VCode [51]. The AMI corpus uses a different approach through use of the Nite XML toolkit which provides extensive support for complex annotation representation and supportive interface. Nite XML toolkit visualization is centered around transcription text (e.g., dialogue) of a corpus being annotated and is linked to supportive media, e.g., audio or video.

It is common for behavior interactions to be described as a sequential sequence of observed

[1]Developed at Arizona State, http://www.cmi.arizona. edu/, but the url for C-BAS is broken.

events. Many experts in the field of behavior analysis express behaviors of interest in this fashion [18, 24, 64, 66, 86, 88, 89, 132, 151]. This is no surprise as such descriptions capture the sequence of events that define the behavior. Many large corpora have been produced to identify and understand behavior among humans interacting within a small group setting (e.g., [18, 24, 86]) One focus of the analysis of these corpora is identifying the structure of behavior patterns. However, there is limited support for searching based on the structure.

Currently, there are a few search strategies in this data domain. Some visualization tools such as Nite XML toolkit has a supportive query language for searching the annotations. Such an approach can be powerful but construction of queries can be complex and cumbersome. ANVIL supports searching amongst the text of annotated event labels. This can be useful when looking for a specific event. However, identifying a sequence of labels does not seem to be supported. Some tools, such as VCode, can export annotated events to a text file where search outside of the application can be performed. MacVisSTA has the ability to save an observation (notebook) and play it back but not find other occurrences of the observation. EXMARaLDA performs search using a tool created by the EXMARaLDA authors called EXAKT. Their search is modeled after KWIC (keyword in context) and has powerful support for regular expressions in text search (search transcription text, annotations, and descriptions). ELAN has similar search support to EXMARaLDA but has the added ability to add temporal relation constraints (Allen's constraints between two intervals [3]). Transformer is purposed for transforming data files for use in one tool to another. They do support text search in which different corpus files can be specified to search.

The search we are interested in is symbolic temporal pattern mining where the focus is discovering "interesting" patterns among *symbolic* time series data (not numerical) [71, 100]. There are a few approaches related to this aspect of STIS. The first is T-patterns developed by Magnusson [83] where a sequence of events will occur within certain time windows, e.g., $A_1 \xrightarrow{T_1} A_2 \xrightarrow{T_2} A_3$ for time intervals T_1 and T_2. T-patterns are used as the basis of pattern representation in Theme [146] where each T is set through various statistical methods. Time interval windows are used in the second related approach, Frequent Episode Mining (FEM) algorithms of [115, 130]. The FEM algorithms use one of two approaches: conditional probability or a frequency threshold, both on defined timing windows.

4.3.3 STIS method

Structural and Temporal Inference Search (STIS) is found-ed on creating a formalism of a pattern based on structure, timing, and ordered relationships. We operate on a pattern at the semi-interval level (start or end of an interval). This representation was first introduced by Freksa in [39] and later revisited by Mörchen and Fradkin in [101]. Semi-intervals allows a flexible representation where partial or incomplete knowledge can be handled since operation is on parts of an interval and not the whole. In this section we discuss how we use semi-intervals to describe a pattern and build a structured search based on the pattern to identify occurrences within a dataset. An overview of our method can be seen in Figure 4.8. Given a set of event annotations (e.g., from ELAN or MacVisSTA), create a semi-interval set which is organized in a database of definitions and instances. This is done offline. Then the expert

provides an event sequence that is converted into a *pattern* which contains implicit search criteria. This is given to STIS which performs structural analysis on the *pattern*, uses the results of the analysis to form search criteria, searches to identify occurrences based on the criteria and returns a set of occurrences. We will discuss the details of what occurs offline and online in turn.

Offline: Event annotations from a multimodal dataset are transformed into a set of semi-interval annotations. We define an event as:

Definition *(Event)*

An *event* is an interval $[r_i, r_j]$ with semi-intervals r_i and r_j, $i, j > 0$, representing the start and end points of the event, respectively.

Our representation of an event does not associate with a particular occurrence time of the event, i.e., r_i and r_j are not the times of the start and end points. This is necessary as many occurrences of the same event can occur. Identification with a particular occurrence time is discussed later. For organizing events, two look-up tables are used. The first, a *definition table*, stores semi-interval *definitions*. A *definition* stores characteristics of the event from which it originated. These characteristics consist of a textual description, the actor involved (or source of the event), the type of event, e.g. modality, and whether it is a start or end semi-interval. These descriptive characteristics are a subset of event aspects in [153], except for start/end. Such characteristics have been used as focal aspects during analysis of event-based multimodal data [24, 88, 101]. The *definitions* are used to store

descriptive information for each semi-interval without repetition (i.e., look-up table of unique *definitions*). The second look-up table, an *instance table*, stores of all semi-intervals in the dataset organized by temporal order. Each semi-interval in this table links to its *definition* in the first table. The *definitions* in the first table allows querying semi-intervals based on characteristics while the second table allows querying of events based on temporal criteria. Currently, our organization of event information is purposed to store and represent interval and semi-interval data. Point data can also be stored in which case a single semi-interval with no matching semi-interval is stored.

Online: An expert provides an event sequence to identify. The sequence is mapped to a *pattern* representation:

Definition *(Pattern)*

A *pattern* is a sequence S of semi-intervals r_i, $i \in \{1, ..., |S|\}$, such that for each $r_i \in S$, $\exists$ r_j such that r_i occurs before or is equal to r_j $\forall$ $i \leq j$; $i, j \in \{1, ..., |S|\}$. Each r_i and r_j has an associated *temporal constraint* $\hat{t}_i$ which is a time window between r_i and r_j such that r_j occurs within $\hat{t}_i$ time of r_i where r_i's time $(t_i) \leq r_j$'s time (t_j), i.e., $t_i \leq t_j \leq t_i + \hat{t}_i$.

An example *pattern* can be seen in Figure 4.9A which represents one rendition of the greeting between two individuals from Section 4.3.1. The *temporal constraint* T expresses r_3 and r_4 to occur within T time units of r_2. This is useful as one may only be interested when A shakes B's hand and says "Hello" within a certain time to A approaching B. If no constraint is given, then matches that are not temporally close will be found but do not represent the

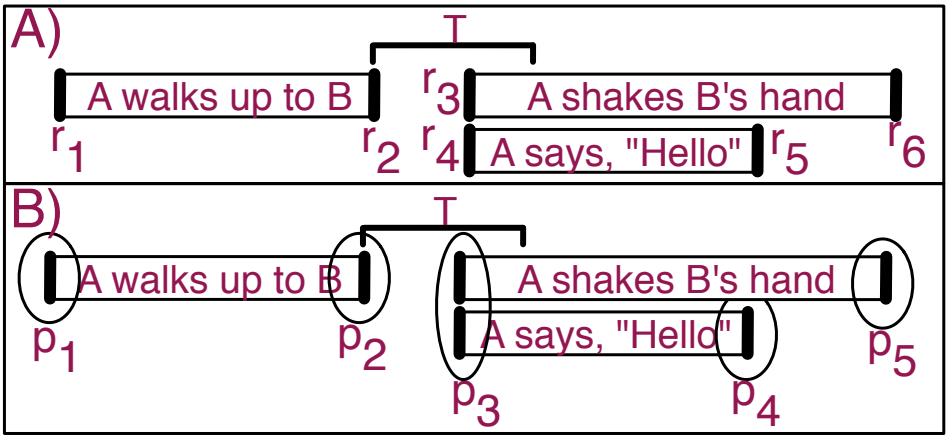


Figure 4.9: A) Example structure of a *pattern*. Note the temporal constraint between r_2, r_3, and r_4. B) Segmentation into *pockets* of equality.

desired greeting occurrence, i.e., ten minutes passes after A approaching B, then A shakes B's hand, etc., which does not represent the desired greeting structure.

Following from this, a *pattern* can be viewed in one of two ways: a complete *pattern* or key-parts of the *pattern*. A complete *pattern* contains complete intervals (i.e., matched semi-intervals). The key-parts represent relevant semi-intervals of the pattern that are key to identification of occurrences of the *pattern*, which could include complete intervals. For example, the *pattern* in Figure 4.9A is a complete *pattern* whereas r_2, r_3, and r_4 within time T represent the key-parts of the *pattern*. Note that the key-parts and the complete *pattern* could be the same and the key-parts need not be unique but their temporal constraints and relational order are relevant to identifying the *pattern*.

The expert's *pattern* is given to STIS as input. STIS then performs two steps: structural analysis and search. The structural analysis step consists of "dissecting" the *pattern* and extract ordered and temporal information. In this step, *temporal constraints* are stored and

the *pattern* is segmented into *pockets* of equality. We define a *pocket* as

Definition *(Pocket)*

A *pocket* p of *pattern* R is a set of semi-intervals $r_i \in p$ such that $\forall i, j \in \text{indices}(p)$, $0 \leq |t_i - t_j| \leq \varepsilon$ where indices(p) is the set of semi-interval indices within p, e.g., i and j.

This use of *pockets* follows from the observation that at the semi-interval level, semi-intervals in a sequence are either equal (within a certain small time window) or not (outside the time window). Hence, semi-intervals can be grouped accordingly into *pockets* of equality. All semi-intervals that are within an ε of each other are deemed equal and grouped in a *pocket*. These groups are separated by temporal order (inequality). The segmentation into *pockets* allows a simple method with which to implicitly store ordered relational information, i.e., using the structure to provide relational information. As can be seen in Figure 4.9, we can see the implicit (ordered) relationships amongst the r_i's. For example, there is no need to explicitly store (remember) that r_3 and r_4 are equal or that r_2 occurs before r_6.

After structural analysis, STIS creates a set of search criteria applied to the *instance table* in which instances are identified. The search criteria contain ordered relationships among the semi-intervals and any defined temporal constraints. An example of search criteria can be seen in Figure 4.10A, where the greeting *pattern* from Section 4.3.1 is revisited. Here, the ordered relational information and a temporal constraint are extracted from the *pattern* and a set of search criteria are created. Then in Figure 4.10B, these criteria are used to identify occurrences within the semi-interval instances (*instance table*). As can be seen, the search

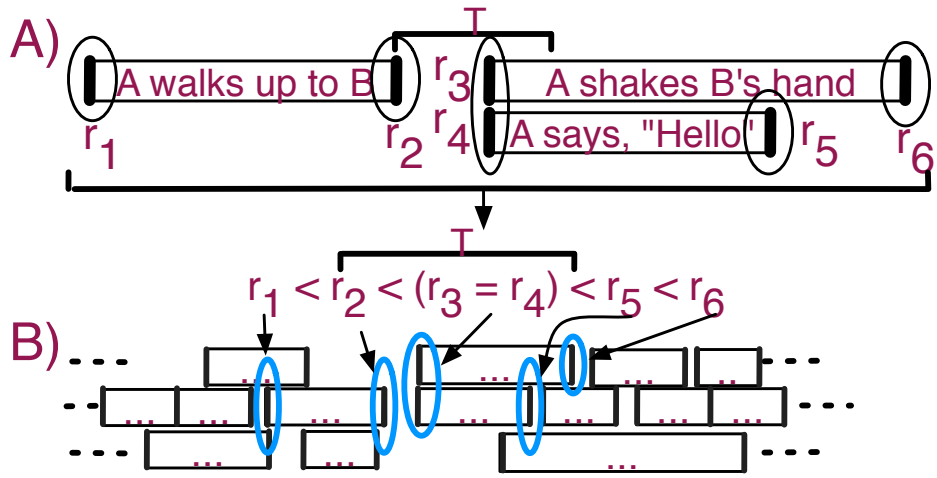


Figure 4.10: A) Example of search criteria and B) search within the semi-interval instances.

Table 4.4: Datasets' Contents.

Data-set	Length (min)	# Semi-interval	# Unique Semi-intervals	Speech Length (secs)			Gaze Length (secs)			Gesture Length (secs)			# Gaze	# Speech	# Gesture
				Ave	Min	Max	Ave	Min	Max	Ave	Min	Max			
Generated	~45	25590	240	1.3	0.06	5	1.27	0.1	5	1.29	0.1	5	10626	7438	7526
										Nodding, Phrase					Nodding, Phrase
AFIT 1	~45	7802	342	1.59	0.1	64.46	2.16	0.1	158.86	0.99, 0.84	0.23, 0.3	10.71, 11.68	4704	1414	1018, 666
AFIT 2	~42	13362	226	2.28	0.03	124.62	1.09	0.07	58.22	0.89, 1.0	0.27, 0.27	13.78, 9.21	11456	1126	610, 170

criteria collected from the *pattern* reflect the *pattern's* structure. STIS uses this criteria to scan all semi-interval instances finding occurrence matches and returning a set of matches. The situated context of each occurrence is preserved as is important for multimodal analysis. This is accomplished through storing the occurrences' times and the semi-intervals within a certain defined time window for each occurrence. This process is based on the work of [92].

4.3.4 Experiments

For multimodal data organized as multi-channel temporal events, we pose the following question: Can search based on a defined *pattern* structure identify occurrences of the *pattern* with greater accuracy than search based on conditional probability thresholds? We first outline our methodology for experimentation followed by a description of the implementations of the search strategies used. After which we describe the datasets used and the behavior categories our experimentation focuses on. The baselines are then discussed. Then we present our results and provide discussion.

Methodology

Since behavior analysis has many variables to consider, testing our search strategy must be done in a controlled environment. To accomplish this, we introduce occurrences of *patterns* with variation into several datasets at known locations, apply the search techniques, then see if the *patterns* can be identified. This is also necessary as exact known ground truth instances for the datasets used is limited. The techniques used are STIS, FEM frequency, and FEM conditional probability. We chose 5 categories of *patterns* in a meeting room setting deemed important by experts, i.e., relevant *patterns*. These categories come from analysis reported in [24, 88]. We then apply the same search techniques to unaltered real datasets with known ground truth.

We experiment with three datasets from the domain of behavior analysis in a meeting room

setting. The first is a generated (synthetic) dataset that is created based on the parameters of real datasets similar to bootstrap aggregating (bagging) [11]. The other two datasets are real datasets consisting of two sessions within a corpora (details in Section 4.3.4).

We introduce into each dataset occurrences of relevant *patterns* with variation based on the 5 behavior categories. Each *pattern* is based on relational structures observed by experts. For each *pattern*, we introduce 10 instances into its own copy of each dataset, i.e., there is no interference between the *patterns* of different behavior categories. We also ensure that none of the 10 overlap. Then we search each dataset copy for its respectively inserted *patterns*. The results are compared to the known inserted locations for accuracy. Power/penalty analysis is used as a metric (described in Section 4.3.4). We then take the two real datasets unaltered and search for occurrences of known ground truth. We conduct these searches using two versions of each *pattern*: the complete *pattern* and key-parts. This allows a comparison between using complete knowledge of a *pattern* and the relevant pieces according to the expert (sometimes complete knowledge is not needed or unattainable).

Since one of our datasets is generated, there is some concern that the *pattern* instances introduced already exist due to random generation. However, the probability that the generated dataset has many relevant *pattern* instances present is very low. This probability was explored in [93].

Implementation

STIS is implemented in C++ using Qt 4.7 [118] for the user interface and a SQLite database for the datasets. The current interface of our system is not shown as it is not the focus of this paper. The FEM frequency algorithm (*FEM1*) is implemented in C++ and the FEM conditional probability algorithm (*FEM2*) is implemented in Java. Both FEM algorithms are part of TDMiner (http://people.cs.vt.edu/pat naik/software). In deciding the appropriate *temporal constraints*, the choice depends on the events involved, what events mean to an expert, and the kind of data. Ultimately, it is up to the one performing the search. For our experiments we chose to use a global 3 second window as a *temporal constraint* between each consecutive semi-interval being matched. The timeframe of behavior patterns is normally temporally tight (on the order of milliseconds up to seconds). Using a 3 second window allows the identification of instances that are temporally tight and those a little longer without a flood of results with many potentially being irrelevant. However, this window can be user set.

Datasets

Here we describe the datasets used for our experiments. Our experiments were conducted using the VACE/AFIT multimodal meeting room corpus [24, 88].

VACE/AFIT: This dataset consists of several meeting sessions of Airforce Officers from the Airforce Insitute of Technology (AFIT) partaking in war-gaming scenarios. We focus

on two sessions (*AFIT 1* and *AFIT 2*). Each session is a scenario in which five officers (C, D, E, F, and G) are given a problem to discuss and resolve. The room where the sessions took place was instrumented for multi-channel audio and video along with motion capture of the officers (details of instrumentation in [24]). The officers in *AFIT 1* are discussing potential candidates for a scholarship. The scenario is that C, D, F, and G are department heads meeting with the institute commandant E to select three scholarship award recipients. The officers in *AFIT 2* are discussing options for exploiting an enemy missile that has been discovered. Each session is approximately 45 minutes with manual and automated annotations for speech, gaze fixations, F-formations, and several gestural forms (including gesture phrases) for each officer. F-Formations, or focus formations, were first identified by Adam Kendon to be units of topical cohesion marked by gaze cohesion of participants to common objects or spaces [60]. Gesture phrases are atomic gestural motions marking a single motion trajectory that typically coincide with atomic speech phrases [87]. These annotations are events that were extracted from the audio, video, and motion capture data and describe the officers' interactions. The sum of the annotations is a dataset consisting of multiple channels (21 for *AFIT 1*, 19 for *AFIT 2*) of overlapping event data extracted from various synched media streams. The sequences of behavior described by the annotations are rich and descriptive. Each dataset is summarized in Table 4.7. For our experiments, *A1* and *A2* are altered versions (i.e., *patterns* introduced) of *AFIT 1* and *AFIT 2*, respectively, and *A3* and *A4* are unaltered versions (i.e., original) of *AFIT 1* and *AFIT 2*, respectively.

Generated: We generated a dataset based on the parameters of the VACE data. For five

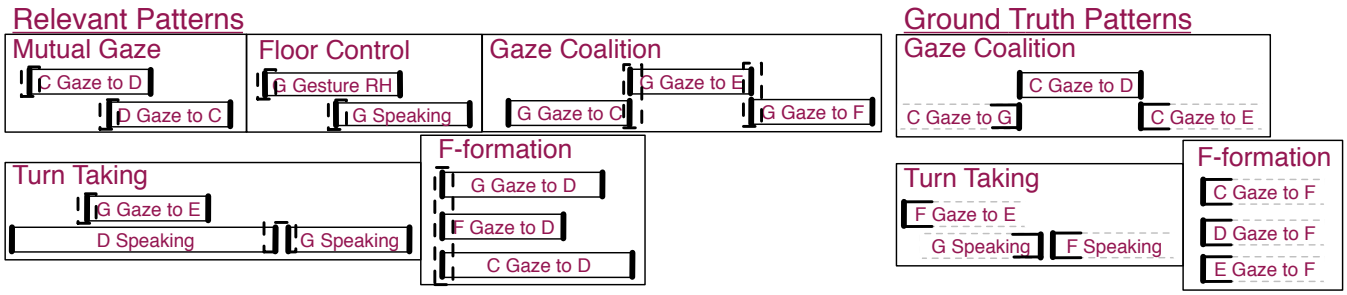


Figure 4.11: (left) *Relevant patterns* used for *G*, *A1*, and *A2*. (right) Ground truth *patterns* of *A3* and *A4*.

fictitious people, we randomly generate 45 minutes of annotations for parallel channels of speech, gaze fixations, and gesture phrases. For each person and each channel, we generate a timeline of events with varying lengths and gaps between them totaling 15 parallel channels. This is fewer channels but much denser as can be seen in Table 4.7 with significantly more semi-intervals. We label this dataset as *G*.

Relevant Patterns

Here we describe the general *patterns* that were deemed interesting by experts and introduced into *G*, *A1*, and *A2*, plus the ground truth *patterns*. The *pattern* structures used can be seen in Figure 4.18. The outlined semi-intervals represent the key-parts of the *pattern*. The actors used for the *patterns* introduced into *A1* and *A2* were chosen so that they did not match the original actors reported by experts in [21, 88]. For example, if a *pattern* we want to introduce was reported involving C gazing at F, then we did not use C and F but instead G and D. This was done to prevent interference from the actual *patterns* observed by the

experts.

Mutual Gaze (MG): In the AFIT sessions, different participants controlled the floor at different times (i.e., leading the discussion for the moment). When the control passed from one participant to the next, there was a mutual gaze exchange between the current holder of the floor to the next.

Gaze Coalition (GC): It was discovered in *AFIT 1* that the social interaction amongst the participants had as much to do with the outcome of the meeting as the specific merits of the scholarship candidates being discussed. The participants dynamically formed coalitions to support each-other's candidates through a process of mutual gaze fixations and back-channel expressions of support during discussions [89].

A coalition to support a proposed idea is initialized when the proposer seeks to make eye-contact with other participants while he is presenting the idea. Participants who supported the idea would return the eye-contact, while those who disagreed would avert gaze. When a return gaze was attained, the presenter's gaze moved to another member.

Floor Control (FC): During a session, a participant would gain floor control through a hand movement (gesture) and start speaking. This was deemed 'floor capturing' in [21].

Turn Taking (TT): As detailed above, each session had a meeting manager who normally was the dominant participant and facilitated the meeting. When a participant sought to take a turn speaking, the participant might look at the meeting manager while the current floor controller spoke. Once the current floor controller finished speaking, the participant

seeking a turn would then begin speaking.

F-formation (Ff): F-formations were observed throughout the sessions. The defining behavior for F-formations observed were concurrent focus on the same person (or object).

Ground Truth: In *A3* a GC *model* was known to exist from unpublished analysis. In *A4*, a TT and Ff *model* were reported in [88]. These represent the only occurrences of a known ground truth *pattern* in which the exact timing within the datasets can be verified (3 in total). The ground truth *patterns* consisted entirely of semi-interval key-parts.

Baselines

We compare STIS against two Frequent Episode Mining (FEM) algorithms [115, 130]. The motivation behind the particular kind of FEM algorithms (*FEM1* and *FEM2*) we use is the discovery of pattern sequences within temporal event data. The authors of *FEM1* and *FEM2* applied their algorithms to neural spike data (i.e., firing patterns of neurons in the brain). The patterns represented by this data have many similarities to our data domain: a sequence of firing times of neurons in sequence, i.e., a sequence of discrete events governed by temporal constraints.

Since FEM algorithms are meant for mining and not searching, we compromise by tuning them similarly to STIS's parameters, and then search their results for relevant *patterns*. This is necessary as there are no other approach like STIS for comparison. The closest is *FEM1* (discussed shortly). In a FEM algorithm, there are several methods for specifying whether a

pattern occurrence is deemed important. The two approaches most pertinent to our problem is frequency (*FEM1*) and statistical significance (*FEM2*). *FEM1* sets a frequency threshold reporting a *pattern* if seen at least as many times as the threshold. *FEM2* is based on the conditional probability of one event given another within a time window. In other words, if interested in A following B within 2 seconds, we would look for $Pr(B|A) \geq \alpha$ where B is within 2 seconds of A and α is a significance (connection strength) threshold. For more details see [130].

Interestingly enough, *FEM1* supports search by *pattern* definition where occurrences of a specified *pattern* are counted without using a frequency threshold. This is analogous to an expert defining a *pattern* to search. The results of this kind of search in *FEM1* would be the same if using *FEM1* to search by *pattern* frequency. The main difference is that using frequency results in a long list of *patterns* to sift through for an exact *pattern* (plus choosing an appropriate threshold) whereas defining a *pattern* to count is focused on the exact *pattern* of interest. Definition of a *pattern* in *FEM1* for counting is closely related to how STIS operates, hence, this approach of *FEM1* is used for comparison.

For *FEM2* , the conditional probability threshold ranged from 0.03 to 0.1 depending on the size of the dataset and the *pattern*. We noticed for smaller datasets, in general, a larger threshold could be used. We use the same 3 second window between semi-intervals for *FEM1* and *FEM2*. *FEM1* functioned on all the datasets in semi-interval form. However, *FEM2* had some limitations requiring the use of interval datasets in most cases. With an interval dataset, *FEM2* performs operations with respect to an interval's start.

Table 4.5: Overall Power/Penalty Analysis

	Complete	**Key-Part**	**Ground Truth**
Misses			
STIS	1 of 150	0 of 150	0 of 3
FEM1	9 of 150	17 of 150	2 of 3
FEM2	22 of 150	12 of 50*	3 of 3
Mean Power			
STIS	99.33	100	100
FEM1	94	88.67	33.33
FEM2	85.33	76*	0
Mean Penalty (Precision)			
STIS	18.67 (81.24)	36.3 (63.7)	75.56 (24.44)
FEM1	19.68 (80.32)	39.87 (60.13)	91.67 (8.33)
FEM2	40.22 (59.78)	23* (77)	100 (0)

Results

The performance of STIS is tested through *power/penatly* analysis of [119]. This is done for datasets G, *A1*, and *A2*. We then look at the results for the three ground truth *patterns* also with power/penalty analysis. In total, STIS was run on 33 *patterns*, *FEM1* on 33, and *FEM2* on 30; in total 96 *pattern* searches were performed. For simplicity, we use the naming scheme X_Y to reference each *pattern* where X is the dataset and Y is the *pattern* abbreviation. For example, *A1_MG* is mutual gaze *pattern* from *A1*.

For describing the results, we use power/penalty analysis which reports a power and penalty percentage. The idea behind power/penalty analysis is that if there are x known instances of a phenomenon in a dataset, y instances identified by a method or algorithm, and z number of instances common amongst the known and identified, $z \leq x$, then the power percentage is $z/x * 100$. For example, if $x = 10$, $y = 18$ and $z = 7$, then the power is $7/10 * 100 = 70\%$, i.e., the method's power is 70% in identifying the relevant instances. The other 11 identified

instances are part of the penalty. These are extra instances the expert must go through and, in turn, are an extra cost. The penalty $= (y - z)/y * 100$, in our example, penalty $= (18 - 7)/18 * 100 = 61.11\%$.

A precision/recall approach is not applied as such approach provides how accurate your *model* is in identifying an instance. However, in our case, we not only want to accurately identify an instance, but whether that instance represents a specific behavior of interest. For example, one can create a detection system for hand waving. However, an expert may not only be interested in hand waving, but when A waves at B. What is detected will either be related to the behavior of interest (power) or not (penalty). However, power/penalty is the same as precision/recall mathematically. Power is the same as recall and penalty is 1-precision. A social scientist does not think in terms of precision/recall but in the power of a *model*/*pattern* in identifying what is relevant and the penalty is anything extra he/she needs to sift through.

Table 4.5 presents the overall power/penalty analysis results. STIS was able to identify nearly all the occurrences (99.33% - only 1 missed). FEM1 and FEM2 missed a number more (94% and 85.33%, respectively). STIS had a higher mean power and a lower mean penalty for the complete *pattern* case. STIS also performed better than *FEM1* for the key-part case. The '*' for *FEM2*'s key-part results signify that these are only partial results. We were only able to run *FEM2* on key-part *patterns* for *A1* due to some limitations of *FEM2* (discussed later). Hence, the results presented in Table 4.5 are for this case. The corresponding sub-set of results for STIS and *FEM1* are 0, 100, 26.46 and 5, 90, 27.61 for

misses, mean power, and mean penalty, respectively. For this case, *FEM2* had a lower mean penalty.

For the ground truth, STIS was the only approach that was able to identify all 3 known ground truth occurrences. We would like to emphasize for the ground truth identification STIS's ability to search for a *pattern* and one of the results be a ground truth occurrence. The high penalty is due to verifying the identification of only one occurrence for each *pattern*. STIS returned at max 5 occurrences for the ground truth *patterns* whereas *FEM1* and *FEM2* returned up to 22 occurrences and for some, did not return any.

In Figure 4.20 we can see the details of the penalty for the complete *pattern*, key-part *pattern*, and ground truth *pattern* cases. As can be seen, STIS and *FEM1* have competing results. The limitations of *FEM2* caused it to struggle with the ground truth case. Not surprisingly, all approaches had their worst performance for the generated data. There is a noticeable difference between the complete and key-part *pattern* cases. The penalty increased for key-part. The reason for this is most likely because the key-part *patterns* contain mostly semi-intervals (not intervals) leading to a greater chance of having more matching occurrences. This is one of the characteristics of the semi-interval representation as a pattern defined using semi-intervals can match a greater number of *patterns* than an interval representation [101].

Comparing the penalty trends across datasets and *pattern* types, we see that STIS has a similar penalty trend between complete and key-part cases for each dataset. STIS seems to be least affected between the datasets but suffers from the same errors between them also.

STIS and *FEM1* display similar trends for *G*, *A1* key-part, and *A2* key-part. This suggests that they may have similar strengths and weaknesses. *FEM1* and *FEM2* had strongly correlated trends for *G* and *A2* complete but not for the other cases. Overall, the penalty trends suggest that there are commonalities in the algorithms that aid in identification but also shared pitfalls that hinder. Potential pitfall causes are the necessity to tune the *temporal constraints* and *pocket* size based on characteristics of the dataset and/or the kind of *pattern*, and making sure semi-intervals that are matched to an occurrence actually make sense, e.g., a start and end are matched to the same event occurrence and not two different occurrences of the same kind of event. Current measures in STIS to minimize this is to verify through the *instance table* that the semi-interval occurrences are matched appropriately. Further work in this area is needed to provide more robust matching. For ground truth, there was no trend seen other than STIS had the lowest penalty overall.

Discussion

In answer to the question posed previously, the results of STIS and *FEM1* confirmed that search based on event structure can identify *patterns* with high-accuracy, and search for *patterns* in multimodal data organized as multi-channel temporal events can benefit from expert input and specification as opposed to a conditional probability method (*FEM2*).

During our experimentation, we observed some limitations in the different algorithms. For *FEM1*, the ground truth *patterns* *GC* and *FF* could not be found as some semi-intervals of the *patterns* occur at the exact same time. *FEM1* does not handle this case, which was

Figure 4.12: Penalty for *G*, *A1*, *A2*, and ground truth.

also observed in [93]. For *FEM2*, significant effort was required to obtains results as we had to continually try different conditional probability thresholds (some as low as 3%). This challenge came from the frequency of the *patterns* being searched for. Compared to the size of the data set, 10 occurrences (or 1 for each ground truth *pattern*) is very small. Hence, why search based on event structure with expert input and specification performed well. The *FEM2* implementation had an inherent limit in the number of reported *patterns* that could be outputted for verification (∼50K). When this limit was exceeded, verification could not be performed as results could not be outputted. The larger the dataset, the greater the number of reported *patterns*, hence, the use of the interval versions of the datasets as they were half the size of their semi-interval counterparts. However, *FEM2* using intervals suffered since the algorithm would only match according to start times and not (seemingly) use the end times. This left *FEM2* operating as if only start semi-intervals were specified leaving a greater possibility for more matches.

For STIS, we encountered an initial identification error with ground truth *Ff*. Our default size of a *pocket* was temporally very tight as the original analysis of the behavior within *A3* and *A4* was focused at a small time scale (milliseconds). One of the gaze events of the *Ff pattern* was outside of our *pocket* size. Hence, we had to slightly change the structure of the *pattern* used to search in order to identify the ground truth *pattern*. This highlights the necessity for greater flexibility when searching using a structural approach, which is an observation we were aware of and this situation confirms such. Another observation is that *FEM1* had less identified patterns than STIS but still high power. We believe this is because *FEM1* returns

non-overlapping *patterns*, i.e., *patterns* that do not overlap each other. STIS does not filter for non-overlapping *patterns* as such *patterns* may contain variations of potential interest to the expert.

FEM2's search strategy is based on identifying *patterns* using defined parameters. We are interested in identifying *patterns* that match parameters *and* also match specific content. By content we mean what the events involved in the *pattern* mean. For example, the structure of the ground truth *models* (Figure 4.18) can match any number of *patterns* in the data. It is the provided content along with the structure that allows an expert to pinpoint occurrences of interest. This kind of search is supported by STIS and *FEM1* and despite the limitations observed, they performed well. Overall, STIS outperformed *FEM1* and *FEM2* and poses to be a beneficial search approach in multimodal analysis tools.

The pattern structures investigated in this paper are the beginning of our research into creation of a set of temporal relationship principles for describing interaction patterns in multimodal corpora. A subset was used in this paper but expansion is underway into representing more complex patterns. This expansion includes negation, pre- and post-conditions, interrupts, and many renditions of repetition of specific events. However, the creation of more complex *patterns* may result in very few matches, which may or may not aid the current analysis (flexible vs. rigid *pattern* definition). Another potential venue of pursuit is using STIS to search partially annotated corpora. Some events are easier to annotate than others (e.g., when someone is speaking, or a person's position in the scene), hence, searching partial annotations can provide likely probable occurrences of events not yet annotated. This

would identify focal areas where efforts can be applied for more detailed annotation creation.

4.3.5 Conclusions and Future Work

In this paper we presented a search strategy for multimodal data based on the structural and temporal aspects of human behavior. We were able to show that a search strategy based on these principles performs well. STIS demonstrated the ability to accurately identify occurrences of *patterns* with an expert defined structure with some (or all) the occurrences identified being ones sought after. *FEM1* was a tough competitor which motivates future investigations of potential incorporation of *FEM1* aspects into STIS. An example being support for non-overlapping events if desired by the expert. Another focus of future work is supporting flexible timing windows (for *pockets* and temporal constraints). Support for such is merely a question of implementation.

4.3.6 Acknowledgments

We would like to thank Debprakash Patnaik for his help in using FEM and Christa Miller for invaluable input as an outside observer and editor. This research was partially funded by FODAVA grant CCF-0937133, NSF IIS-1053039, and NSF IIS-1118018.

4.4 Interactive Process

Our fourth and final component is to incorporate the expert in the search and discovery process, allowing them to use their knowledge and expertise to guide *model* identification. The prior three components are utilized to produce an interactive and iterative process that is discussed in detail in this section. Publication of this piece can be seen here [93] and is included below with minor adjustments. A more detailed discussion of evolutionary strategies was also added in Section 4.4.4. The presented work in this section was accomplished using *version 1* of our system details in Section 6.2.

Abstract: This component investigates a technique for the discovery of temporal behavior *models* within multimedia event data. Advancements in both technology and the marketplace present us the opportunity for research in analysis of situated human behavior using video and other sensor data (media streams). By situated analysis, we mean the study of behavior *in time* as opposed to looking at behavior in the form of aggregated data divorced from how they occur in context. Human and social scientists seek to model behavior captured in media, and these data may be represented in a multi-dimensional event data space derived from media streams. The knowledge of these scientists (experts) is a valuable resource which can be leveraged to search this space. We propose a solution that incorporates the expert in an iteratively, interactive data-driven discovery process to evolve a desired *behavior model*. We test our solution's accuracy on a multimodal meeting corpus with a progressive three tiered approach.

4.4.1 Introduction

In this paper, we address how human and social scientists may derive *models* of human behavior that are captured in media streams as these behaviors are situated in time. Such situated analysis considers the study of behavior *in time* as opposed to looking at behavior in the form of aggregated data divorced from how they occur in context. This requires the traversal of two 'semantic gaps' from media stream to meaning-units of behavior [49, 57, 139]. First, relate the entities detected (e.g., image histograms, audio frequency processing, color blobs, face recognition, and person tracking) to events relevant to a particular analysis (e.g., person A has started looking in the direction of person B). Second, relate the detected events to the high level behaviors of interest to human and social scientists studying such behaviors. Our focus in this paper is the second of these.

Human and social scientists approach data from a theoretical viewpoint, forming hypotheses of what behaviors they see or expect to see in the data. Analyzing a large amount of event data derived from media streams, however, is a tedious endeavor, and it is desirable to apply machine learning approaches to assist in this process. Machine learning, on the other hand, often takes a *tabula rasa* approach that is not able to take advantage of the knowledge of human experts. Many machine learning approaches produce their own models requiring only that experts provide labeled data. Furthermore, many *behavioral models* are structural in nature requiring construction of relationships between events and component behaviors. Automated learning, on the other hand, is often framed as the learning of weights and parameters, as opposed to the structure of the *model* itself.

Our approach is to reason with temporal events (e.g., [A approaches B, start]) out of which a temporal *model* graph is constructed. A *model* of a greeting between two individuals may begin with a simple structural relationship (e.g., [A approaches B, start] ¡followed by within 1 sec¿ [B extends hand, start]) where one could potentially 'evolve' the *model* by successively adding/removing relationships with other events in the graph, and/or pruning graph connections until a desired formulation is reached. However, evolving this graph without guidance is a very large search space even for a small graph.

Our solution is to use real occurrences in data to help constrain the generation of alternatives and produce a convergence to a desired *model*. We engage the expert in an interactive data-driven discovery process to evolve a *behavior model* to a desired formulation. The expert brings to the table ideas and hypotheses with which she creates an initial exploration *model* (seed) to start narrowing the possibilities. The challenge is to find likely, relevant extensions to the *model* that somehow recur in the data-set, and to allow the expert to steer the evolution process iteratively. Given a seeded *model*, the system identifies situated instances that occur in the data. Potentially relevant extensions to the *model*, based on the situated instances, are presented to the expert. Upon choosing an extension, the *model* is updated. The system then identifies situated instances of the newly updated *model*. This process continues iteratively with the end result being a *model* representing a behavior as it exists in the data, and yet reflecting the expert's knowledge.

Our target domain is understanding the behavior interactions amongst humans. However, our approach is applicable to any temporal modeling involving graphs of discrete events,

such as seismic event data, Internet network traffic, and production process modeling.

In Section 4.4.2 we describe the data domain that motivates our research, the kinds of events we process, and connections to other multimedia data. We then describe related work in Section 4.4.3. Section 4.4.4 provides a detailed description of our *model* evolution process. Afterwards, in Section 4.4.5 we describe our experiments, results, and provide discussion. We then close with conclusions and future work in Section 4.4.6.

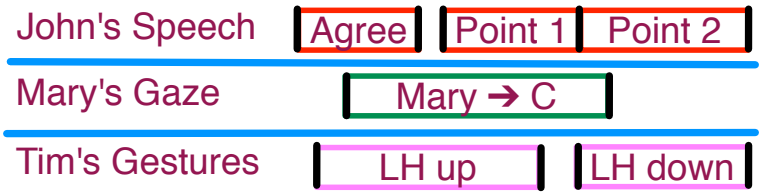


Figure 4.13: Example of multi-channel event data (*S-I's* highlighted as vertical bold lines)

4.4.2 Data Domain

The motivating domain of our research is a subset of multimedia analysis known as behavior analysis. We first explain our notion of an event. An event, or interval, is an occurrence in time that has a beginning, an end, a description, and an associated actor/source of the event (subset of event aspects in [153]). The atomic units (beginning and end) of an event are semi-intervals (*S-I's*) as described in [39]. A series of events in succession are an event sequence (Figure 5.1). These sequences may include overlaps between events, but all the events in a sequence have ordered relations (e.g. '¡' or '=') with respect to each other.

Behavior analysis includes the analysis and understanding of interactions among humans.

More specifically, our focus is on a meeting room setting in which a small group of individuals is gathered to discuss topics of interest or solve a proposed problem. The meeting is normally recorded using multi-channel video and audio and sometimes with other sensor data such as motion capture [18, 24, 124, 151]). The analysis of the signals recorded is known as multimodal analysis which integrates multiple data channels (e.g., gaze, gesture, and speech) where incorporation of such information provides more accurate analysis results [124]. The events within these data channels (i.e., events that describe the gaze, gesture, and speech of the participants) are produced through segmenting and annotating the media signals. This process is performed through manual and automated means as discussed in [24, 88]. It is these annotations (segmented events) that our approach is applied to. At a more abstract level, these events represent segmented temporal events from multimedia signals and in turn application of our approach to analyze a wider range of multimedia data is feasible.

Since human behavior is variant, the idea that represents an interaction, e.g., a greeting, may be formulated many different ways in the data making modeling difficult. How does one identify situated instances of behavior when the way they exist in the data may be unknown? Also, every observed behavior has the potential to be relevant to an expert depending on her analysis goals. Hence, there is no concept of "noise" but rather one of relevance. For example, consider the situation where three students are working together on a math problem when a door slams nearby and draws their attention. One expert may analyze the co-construction of space based on the students' aligned gaze while another may analyze interrupting events. The door slam is "noise" to the first expert but not the second.

These aspects of behavior analysis pose challenges to modeling.

4.4.3 Related Work

Our proposed approach is to evolve a *model* from a seed *model* graph. This is related to the general domain of Evolutionary Computing (EC) where our *model* evolution strategy is most closely tied to Genetic Programming (GP)[68]. GP is applied to change and update the structure of models iteratively to arrive at a solution. Bastian in [6] employed GP for creating fuzzy logic models which generates models that represent relationships between nominal data and uses pre-defined relational models. Gray et al. [47] comment on how expert knowledge is useful in molding solutions during a trial and error approach and how such input is necessary in initializing the models and appropriate functions for GP. The idea of operating on a model's structure has also been explored in other domains. The predominant application has been to numerical contexts where numerical model modification has been used, as in [27, 59, 80].

Our data is inherently multimodal. A prominent area of research for multimodal interaction data processing is human behavior analysis. Two different classes of approaches, unimodal (e.g., speech-only [17, 52]) and multimodal (e.g., speech, gesture, and gaze [24, 88, 89, 124]), have been employed to categorize and analyze interactions among humans. These approaches capture different signal sources and present the data to experts to analyze through automatic and manual means. This analysis domain exemplifies the synergy of human and machine.

The coupling of human and machine is realized in varying degrees. On one side is full automation for learning *models* where the most common approach is explicating a training data-set and constructing a representative *model* based on the data. The results are then presented to the user. Learning, in this class of approaches, is often construed as learning the weights or parameters of the *model*, rather than changing the *model's* underlying structure. Some approaches approximate structural modification by over-specifying the mo-del, and allowing some weights to go to zero (or infinity) through the learning process, but such approaches cannot make additions to the original *model*. Classic examples are Hidden Markov Models and Neural Nets [128].

The other side is incorporation of human input such as Relevance Feedback [127] where a *model* is refined by adjusting weights according to user choices. These weights are hidden to the user making it difficult to understand the *model* structure. Other research, such as [31, 134, 152], have incorporated expert knowledge as a guiding force in discovery. Notably [152] investigates different visualization techniques for supporting exploratory search of temporal event sequences.

Ordered relations inherent in temporal event data are an active area of research. Allen in [3] formulated thirteen relationship principles that express all the possible orderings between two event intervals. These orderings describe how events relate. Research, such as [50, 58, 98, 135], has focused on processing numerical or descriptive univariate, multivariate, or symbolic temporal data with the goal of discovering events and temporal relationships from temporal data. Others have explored the discovery of temporal knowledge and associated

patterns, such as [54, 71, 100].

The stream data of multimedia data-sets is a mixture of ordinal and nominal data. As per the temporal data models reported in [100], gesture traces are a collection of univariate numerical time series (ordinal) while speech and gaze target fixations are multivariate symbolic interval series in which their meaning is categorically descriptive (nominal). Explorations into the relational structures of nominal (event) data can be seen in [25, 74, 159]. More specifically, [95] explored how to structure event data based on semi-interval processing using n-gram techniques from speech processing. Other research [39, 101] has explored the concept of breaking temporal events into core units of beginnings and endings through *S-I* temporal ordered processing. Previously mentioned research ([3, 58, 98, 135]) has investigated patterns and relations among interval data irrespective of the data being ordinal or nominal.

4.4.4 *Model* Evolution

The idea of *model* evolution is motivated by GP, an EC approach that operates on a *model's* structure to reach a maximum fit for a problem [68]. In GP, there is a population of individuals consisting of potential solutions to a problem. Each individual's fitness is based on how well they solve the problem. This "wellness" is usually determined by some numerical error function. Individuals are chosen based on their fitness to undergo different genetic operations and produce the next generation. The less fit "die" off leading to successive generations converging to an optimal solution. Genetic operations usually consist of reproduction, crossover,

recombination, and mutation. However, due to the nature of our problem, we approach a genetic operation as an incremental change to the *model* structure.

The next section provides a detailed discussion of Evolutionary Computing Strategies and provides background which influenced the final choice for our *model* evolution process.

Evolutionary Computing Strategies

Evolutionary Computing (EC) has been an area of research providing unique and ingenious solutions to many types of problems. The motivation of EC comes from the observed biological process of natural evolution where a population of organisms evolve or change between successive generations in response to their environment [32]. Each individual in a population is able to survive based on their fitness in the current environment. The individuals with a better fitness are more likely than the less fit ones to survive to reproduce and create the next generation, and hence, pass on their traits representative of their better fitness.

The approaches in EC are called Evolutionary Algorithms. They incorporate evolution principles such as a population of individuals, reproduction, fitness, selection, and mutation.

A population of individuals consists of potential solutions to the problem of interest. Each individual is seen as a potential solution and has a fitness based on how well it solves the problem. Individuals are chosen to reproduce based on their fitness and produce children. The children inherit different traits of the parents that contributed to the parent's fitness value. The survival of these parents and children are based on how fit they are. Which

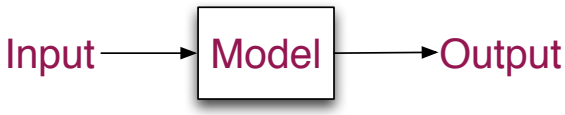


Figure 4.14: Problem Template, motivated from [32].

Table 4.6: Problem Descriptions

Problem Domain	**Input**	**Model**	**Output**
simulation	known	known	unknown
optimization	unknown	known	known
modeling/system identification	known	unknown	known

parents and children survive is based on a defined selection process dependent on the problem. These reproduction cycles generate different generations of the population and over time as more and more generations are produced and less fit individuals "die" off, the population approaches an optimal solution for the problem. During these generational cycles, some individuals can undergo mutations which introduces a random change to the individual which may or may not improve their fitness.

A convenient breakdown of the application of EC to specific problem domains is presented by Eiben and Smith [32]. A given problem can be deduced to a simple generic template as seen in Figure 4.14. There are three parts: the input, the output, and the model that connects the two. The model can be see as the system being used; knowledge of the model brings knowledge of how the system operates. The system's response is the output given a particular input. Two parts of the problem must be known in order to solve for the third. By iterating through the permutations possible by keeping two parts known and the third unknown leads to a taxonomy of three problem domains, which can be seen in Table 4.6.

This problem taxonomy includes simulation, optimization, and modeling problem domains. There are four areas of Evolutionary Computing that have been developed and can address these three problem domains: Evolutionary Programming, Evolution Strategies, Genetic Algorithms, and Genetic Programming.

Evolutionary Programming (EP) had the original goal of producing a form of artificial intelligence (AI) in which prediction of one's environment is a key component [37]. The idea behind EP is to learn about the environment, predict it, and adapt to it [36] or in other words, learning adaptability to the environment of the problem [32]. EP is used for simulation problems where the input and model are given and the output is predicted.

Evolution Strategies (ES) were originally developed for shape optimization [9]. This is a class of optimization techniques based on evolutionary principles in which mathematical models or representations of a problem are iteratively evolved to approach and reach an optimal solution to a given problem [8]. This EA is used for solving continuous optimization problems where the model and the desired output of a problem is known, but an optimal input must be discovered.

Genetic Algorithms (GA) were originally developed by Holland to study adaptive behavior [53]. GAs have been viewed mostly as a means of function optimization [32]. They are the most widely known of the EC strategies [32]. This EA shares a common goal with ES with a slightly different application to combinatorial optimization problems [32] where the model and the desired output of a problem is known, but an optimal input must be discovered.

Genetic Programming (GP) is the youngest of the EAs and has a different application area. The previous EAs are used for some form of optimization or simulation, but GP is applicable to machine learning problems. GP can be viewed as seeking the maximum fit of a model to be used [32]. The input and output are known with the goal being to discover the optimal model. GP has been called the process of evolving programs [68, 1], which can be viewed as a means of creating the operator for the input to produce wanted output.

Overall, the ideas behind EC strategies all share the common goal of iteratively searching for an optimal solution to a defined problem by application of evolution principles. Since GP operates on the model itself, it was the natural choice as a template for our *model* evolution strategy.

Evolving a *Model*

We view the creation of a *model* as "evolving" its structure towards a goal. We start with an initial *model* and modify it incrementally to arrive at a desired structure. We perform evolution at the *S-I* level, which we call a *1-step* change. This represents the smallest modification possible given our event representation. Each *S-I* can be viewed as an element in a sequence. A pathway, or link, between *S-I's* represents the occurrence of order such as a branch in a temporal graph. Figure 4.15A represents an example sequence of four *S-I's* where the arrows show possible pathways. The pathway chosen depends on the order of the *S-I's* observed in the data. A *1-step* change could be: ($C1$) add a *S-I*, ($C2$) replace a *S-I*, ($C3$) remove a *S-I*, ($C4$) add a link, or ($C5$) remove a link. However, we only want

the changes that will better advance the *model*. A brute force calculation of all possible variations could be performed with the "best" chosen, but, the possible variations of *1-step* changes are exponential with respect to the size of the *model* as is explained in the following discussion.

Let G be a graph *model* of *S-I's*. For each *S-I* $X \in G$, a new *S-I* Y can be placed in three general positions with respect to X: *before*, *after*, or *concurrent-with* (Figure 4.15B). These are a subset of Allen's principles [3]. These general positions stem from the left-to-right ordered properties of G where relative position infers specific order. Since G has this ordered property, all *S-I's* in G can be spatially organized. Each *S-I* can be placed at its respective unique ordered position *S-I*$_t$, $t \geq 0$, with *S-I's* sharing a t placed concurrently. Given the above, the following theorem can be stated.

Theorem 1. *Let G be a graph* model *of ordered* S-I's *and $|G|$ be the number of* S-I's *in G. Each* S-I *in G has a* S-I$_t$, $0 \leq t < T$ *where T is the last left-to-right ordered unique position for any* S-I. *Then $0 < T \leq |G|$ since some* S-I's *in G will occur concurrently. Then for a new* S-I *Y, the number of possible positions x in G for Y is $2T + 1$.*

The proof for this theorem is in Appendix A.

A new *S-I* can be potentially linked to all other *S-I's* in G assuming no restrictions from matching *S-I's*. For every new *S-I* Y, there are $\binom{n}{k}$ possible links, where $n = |G|$ and $1 \leq k \leq n$. Then, for every Y, the number of possible link combinations is $\sum_{k=1}^{n} \binom{n}{k}$. Since

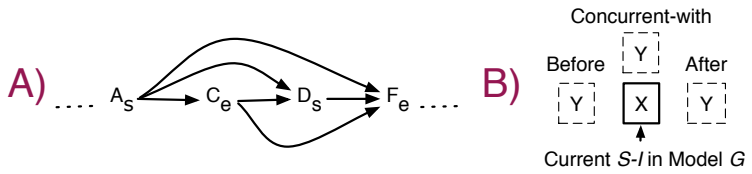


Figure 4.15: A) Example sequence of *S-I's*. Arrows are connecting links. B) General positions for a new *S-I* Y with respect to a current *S-I* X in *model* G.

there are a maximum of $2T + 1$ possible placement positions for Y, there are

$$(2T + 1) \sum_{k=1}^{n} \binom{n}{k} \tag{4.1}$$

possible unique *1-step* addition changes for G. Since some additions for Y may not be feasible due to relative order restrictions, equation 4.1 is an upper bound.

Using equation 4.1 and $T = |G|$, a *model* with $T = 10$ has 21,843 possible unique *1-step* changes and 2.5480×10^{32} for $T = 100$. These calculations only include the *1-step* changes that add a *S-I* and potential links (C1 and C2) and do not include the other possible modifications (C3-C5).

It is difficult to fully automate informed selection for *1-step* changes absent of human knowledge. The relational order of the data automatically prunes possible link connections. Adding expert guidance provides further pruning. However, even with expert guidance, there are still many potential modifications to choose from.

Comparison Metrics: The appropriate *1-step* change at any point depends idiosyncratically on the behavior being *modeled*. The comparison metrics we used extend from [95] where n-gram processing is adapted from speech processing. N-grams are a probabilistic

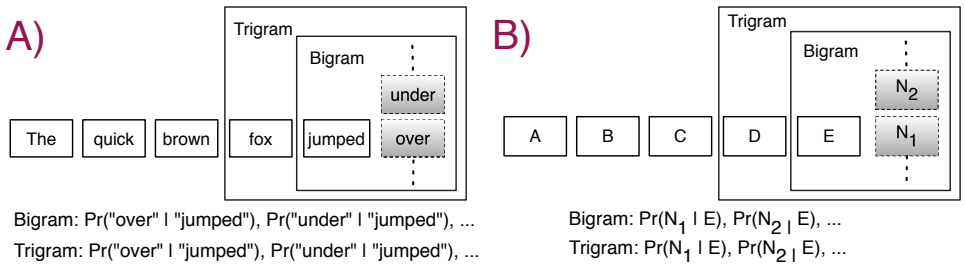


Figure 4.16: A) Example of n-gram use in speech processing. B) Simple example of applying n-gram processing to a sequence of *S-I's*.

method based on conditional probability where history and context are used to provide an informed probability of what might occur next in a given sequence [12]. Figure 4.16A shows how n-grams are used in speech processing where bigrams and trigrams ($N = 2$ and $N = 3$, respectively) are used to provide conditional probability of what word will come next given what has been previously seen. Figure 4.16B shows a simple example of how n-gram processing can be applied to a sequence of *S-I's*.

N-grams provide a probability metric to compare temporally near *S-I's* to a *model* instance's *S-I's*. These temporally near *S-I's* are *related* to the instance. These *related S-I's* can be labeled under three relational categories: *previous* [95], *current* [92] and *next* [95]. Each *S-I* of the *model* has *related S-I's* potentially falling into these three categories. These relational categories stem from the general positions described in Figure 4.15B. The *related S-I's* to a particular instance are *suggestions* to consider for extension. Selecting a *suggestion* for an extension represents a *1-step* change. We extend the relational categories by defining an ε time window for determining and constraining each category similar to the **window-size** of [58]. For *current* this is [-ε,+ε], and for *previous*/*next* this is [-$\hat{\varepsilon}$,-ε) and (+ε, +$\hat{\varepsilon}$],

respectively.

Result Rankings: The convention in most speech processing research is to use n-grams of order $N = 2$ and $N = 3$ [12, 55, 85]. Sun and Applebaum [143] used $2 \leq N \leq 6$. We combine the results of several n-gram processing passes of varying N values: 2, 3, and the whole *model*. Our rationale is that N values of 2 and 3 provide local and internal results with respect to the *S-I's* within the *model*. Meanwhile the whole *model* provides results pertinent to the current chosen sequence of *S-I's* overall. Due to multi-pass processing, *suggestions* will occur in duplicate. However, the results from one processing pass will differ in meaning with respect to another. Each pass uses a different N, leading to varying degrees of *confidence* in the pass results. The greater the N, the more *confidence* there is in the results, because the greater the N, the more history and context is used in the calculation. For example, in Figure 4.16B, the bigram results will have a confidence ranking of 1, the trigram 2, and the whole *model* 5. Hence, the *suggestion* results are reported with an associated n-gram probability and a *confidence* level.

GP Variation

We developed a variation to the standard GP formulation because a direct mapping to GP was impractical for our problem domain as our data is not entirely numerical and interactions with the expert requires a more easily understandable *model* structure. In our solution, an individual in the population is viewed as a *model* instance in the data-set (the population). The fitness for an individual is the subjective *1-step* change chosen by the expert. An

operation is a *1-step* change based on the context of instance(s). A *1-step* change focuses the evolution to individuals that better match the interest of the expert. A generation is produced after each *1-step* change, signifying one evolution iteration.

An added facet of our approach is the analysis focus can be performed from two different views. The first focuses on one instance (an individual in the population) of a *model* and views the instance situated in context. This is a situated view. The second views all potential evolvable directions in an aggregated view. All results from each instance are shown to the user. The utility of both have been studied in [92].

The pseudocode of our approach can be seen in Algorithm 1. Given the user specified initial *model* M and N events in a data-set, all x possible instances of M and all b bigrams and t trigrams are identified in the data-set. N-gram processing is then applied resulting in *previous*/*current*/*next suggestions* for each *S-I* $\in M$, bigrams and trigrams with associated probabilities. From these *suggestions*, the user can make a choice of which *1-step* change to accept. When one is accepted, new *suggestions* based on the updated *model* are presented signifying the end of one evolution iteration. As new *suggestions* are accepted, choices for new modifications dynamically change based on the evolving *model*.

The time complexity of our algorithm is linear with respect to the input (e.g., number of events in the data-set). The identification of all instances of M and each bigram and trigram, x, b, and t, respectively, each require a pass through all the events. For *suggestions*, each instance has a list of p *previous*, c *current*, and n *next suggestions* for the first and last semi-interval (6 in total). The algorithm requires a visit to each item in the lists. The size

Algorithm 1 Pseudocode for *Model* Evolution Algorithm

Start with initial *model* $M_i, i = 0$, data-set of N events
repeat
  Identify instances of M_i, bigrams, and trigrams
  for all Instances **do**
    Extract the *prev/current/next suggestions*
  end for
  Gather *suggestions* and calculate probabilities
  Display for user, user chooses a *suggestion*
  M_i is updated to $M_{i+1}, i = i + 1$
until the user is satisfied with the *model*

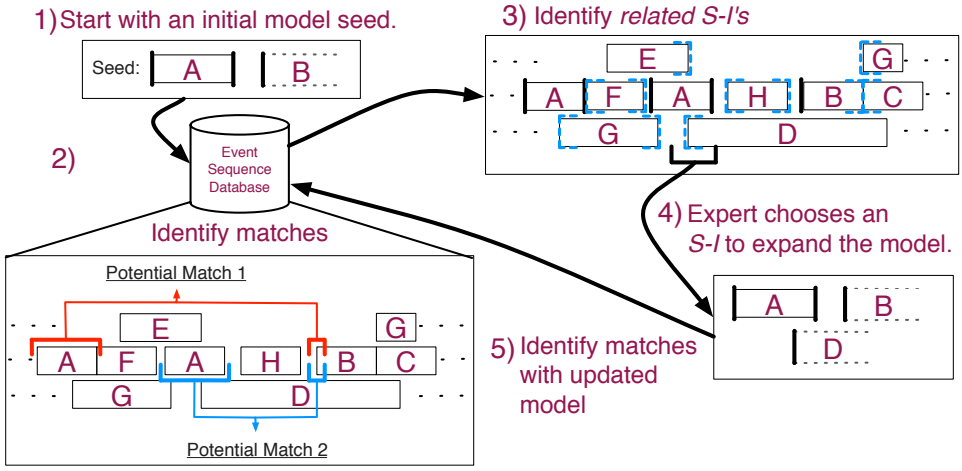


Figure 4.17: Overview of iterative *model* evolution.

of each list depends on the time window and the data-set density. For determining time complexity, we assume they are the same size for all instances. Hence, the size of each list is $\alpha_x = 2(p_x + n_x + c_x)$, $\alpha_b = 2(p_b + n_b + c_b)$, and $\alpha_t = 2(p_t + n_t + c_t)$, for M, bigrams and trigrams, respectively. Therefore, for each M, bigram and trigram, there is one iteration through the data-set (N) plus an α for each instance, leading to:

$$(N + x\alpha_x) + (N + b\alpha_b) + (N + t\alpha_t)$$

For probability calculations, the complexity follows from [12] where the members of each *suggestion* list of each instance are visited to count a total and sum the unique events in the lists. Then one calculation is performed for each unique event found with the worst case being all events are unique (same number of calculations as prior step). Hence, for each α and its respective number of instances, e.g., x, the calculation is $2x\alpha_x$. Adding to the prior equation:

$$(N + x\alpha_x) + (N + b\alpha_b) + (N + t\alpha_t) + 2(x\alpha_x + b\alpha_b + t\alpha_t)$$

Which simplifies to:

$$3N + 3x\alpha_x + 3b\alpha_b + 3t\alpha_t$$

The time complexity will not be linear if any of the α's and/or x, b, or t's are N, which either means the entire data-set is included in one of the *suggestion* lists or the data-set consists of the same *model* repeated. Either is an unlikely scenario for the applicable domain.

An example of the operation of this algorithm is in Figure 4.17. The expert starts with an initial *model* seed (1) which is used to identify matches within the event sequence database (2). From the matches, *related S-I's* are identified given the defined timing windows (3). The expert then chooses a *related S-I* to expand the *model* (4) signifying one evolution iteration. Then matches of the updated *model* are identified in the database marking the next iteration (5).

4.4.5 Experiments

Here we present the results of experiments focused on three aspects: 1) The accuracy of identifying relevant *models* conducted using several data-sets in a controlled environment, 2) How our approach compares to a traditional pattern mining algorithm, and 3) The ability to identify relevant *models* amongst extracted events from a media source.

Methodology

The nature of our problem domain makes experimentation a challenge. Quantitative metrics are difficult to devise for evaluation of *model* discovery dependent on an expert's interest. In order to verify the accuracy of our approach to identify *models*, data-sets with known ground truth (i.e., known relevant *models*) are required. Such *models* must also have significant interest to an expert. Hence, to provide a controlled environment where our results can be measured, we insert *models* into data-sets at known locations and apply our approach to see if they can be identified. We choose 5 categories of *behavior models* in a meeting room setting deemed important by experts [24, 88]. Our test of accuracy is three tiered; each builds upon the prior.

Tier 1: The first tier consists of generating a data-set based on the parameters of a real data-set similar to bootstrap aggregating (bagging) [11]. We then insert *models* based on the 5 behavior categories. Each *model* is based on relational structures observed by experts. For each *model*, we insert 10 instances into its own copy of the generated data-set, i.e., there

is no interference between the *models*. Then, for each *model*, we seed our approach with a relevant starting point of the *model*, identify its locations, then evolve the *model* to the desired structure and see how many inserted instances are identified. Since the data-set is generated, there is some concern that the *model* instances inserted already exist due to random generation. However, the probability that the generated data-set has many *model* instances present is very low (See Appendix B). This tier allows us to see how our approach operates in a dense data-space as the generated data-set is denser than the real ones it is modeled after.

Tier 2: The second tier comprises two real data-sets and follows the same procedure of the first tier where the only difference is these data-sets are real. This tier allows us to see how our approach operates within a real data-space. The first and second tier will demonstrate the accuracy of our approach to identify relevant *models*.

Tier 3: The third tier uses the same two real-world data-sets and does not insert any *model* instances. Instead, we search for known, relevant *models* that are reported by experts who have analyzed the data-sets. In some cases, we do know the exact location of a *model* instance, however, in most cases we only know the general structure of a behavior as described by the experts in their reported analysis. Since most results cannot be verified as in the prior two tiers, this tier allows us to demonstrate the flexibility of our approach to identify *model* structures without exact knowledge of how the *models* are realized in the data.

Baseline: As a baseline, we apply a Frequent Episode Mining (FEM) algorithm [115] to discover the *models* in all tiers. FEM is a well tested and widely used approach to finding an

event sequence (pattern) within temporal event data. Our interest is in comparing against a mining approach based on frequency. Hence, we choose to compare against an FEM algorithm that specifically performs frequency of patterns. We tuned the FEM algorithm to use a 3 second window and identify *models* with a frequency of at least 7 or 8 (depending on the data-set). We tested with a *S-I* and interval version of each data-set.

Implementation

Our approach is implemented in C++ using QT 4.7 [118] for the user interface and a SQLite database to store the data-sets. The current interface of our system is not shown as it is not the focus of this paper. We set the timing window parameters to: $\varepsilon = 0.033$, $\hat{\varepsilon} = 1.0$ and $\dot{\varepsilon} = 1.0$. The value of ε represents the time of 1 frame in video (33 ms at 30 fps). This is the smallest granularity of the camera allowing a 1 frame buffer on either side. The timeframe of *behavior models* is normally temporally tight (on the order of milliseconds or seconds), hence, setting $\hat{\varepsilon}$ and $\dot{\varepsilon}$ to 1 second. When matching *S-I's* of a *model* to instances in the data-set, we use a 3 second window between each *model S-I*. This allows identification of instances that are just outside our 1 second window or identify instances that are temporally longer.

Data-sets

Here we describe the data-sets used for our experiments. Our experiments were conducted using the VACE/AFIT multimodal meeting room corpus [24, 88].

VACE/AFIT: This data-set consists of several meeting sessions of Airforce Officers from the Airforce Insitute of Technology (AFIT) partaking in war-gaming scenarios. We focus on two sessions (*AFIT 1* and *AFIT 2*). Each session is a scenario in which five officers (C, D, E, F, and G) are given a problem to discuss and resolve. The room where the sessions took place was instrumented for multi-channel audio and video, and motion capture of the officers (see [24] for instrumentation details). The officers in *AFIT 1* are discussing potential candidates for a scholarship. The scenario is that C, D, F, and G are department heads meeting with the institute commandant E to select three scholarship award recipients. The officers in *AFIT 2* are discussing options for exploiting an enemy missile that has been discovered. Each session is approximately 45 minutes with manual and automated annotations for speech, gaze fixations, F-formations, and several gestural forms (including gesture phrases) for each officer. F-Formations, or focus formations, were first identified by Adam Kendon to be units of topical cohesion marked by gaze cohesion of participants to common objects or spaces [60]. Gesture phrases are atomic gestural motions marking a single motion trajectory that typically coincide with atomic speech phrases [87]. These annotations are events based on low-level feature extraction and manual inspection of the synched audio, video, and motion capture data and describe the officers' interactions. (further details in [24]). All annotations were either created or validated by human annotators (behavioral psychologists) with cross validation. The speech was automatically segmented and time-aligned with manual inspection to correct errors. No uncertainties associated with the annotations were recorded. McNeill in [88] did report gaze coding issues with officers wearing glasses, however, there

Table 4.7: Data-sets' Contents.

Data-set	S-I's	Unique S-I's	Speech Length			Gaze Length			Gesture Length			# Gaze	# Speech	# Gesture
			Ave	Min	Max	Ave	Min	Max	Ave	Min	Max			
Generated	22812	240	1.26	0.1	5	1.28	0.1	5	1.28	0.1	5	7560	7678	7574
									Nodding, Phrase					Nodding, Phrase
AFIT 1	7802	342	1.59	0.1	64.46	2.16	0.1	158.86	0.99, 0.84	0.23, 0.3	10.71, 11.68	4704	1414	1018, 666
AFIT 2	13362	226	2.28	0.03	124.6	1.09	0.1	58.22	0.89, 1.0	0.27, 0.27	13.78, 9.21	11456	1126	610, 170

was only a problem with 8% of gaze judgements for such officers and less than 3% for the best officer. The sum of the annotations is a data-set consisting of multiple channels (21 for *AFIT 1*, 19 for *AFIT 2*) of overlapping event data. The sequences of behavior described by the annotations are rich and descriptive. Each data-set is summarized in Table 4.7.

Generated: We generated a data-set based on the parameters of the VACE data. For five fictitious people, we randomly generate 45 minutes of annotations for parallel channels of speech, gaze fixations, and gesture phrases. For each person and each channel, we generate a timeline of events with varying lengths and gaps between them totaling 15 parallel channels. This is fewer channels but much denser as can be seen in Table 4.7 with significantly more *S-I*s.

Models

Here we describe the behavior categories used to create the *models* inserted. One observation to take note of when looking at the structures of the different behavior categories is that it is not just the structure that provides meaning to the *models* but the meaning of the events in the specific temporal order. The specific *models* used for all tiers can be seen in Figure

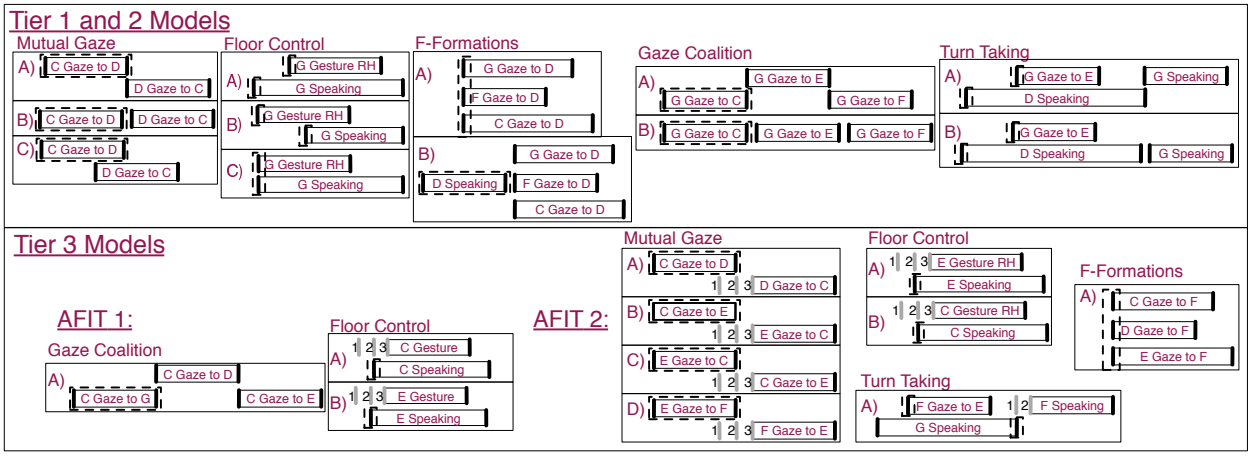


Figure 4.18: *Models* for Tier 1, 2, and 3. Outlined *S-I's*/intervals were the initial seed.

4.18. The outlined *S-I's* and intervals were the initial *model* seeds. Also, the actors chosen for the *models* inserted into *AFIT 1* and *2* were intentionally chosen to not coincide with those involved in the *models* reported by the experts in [21, 88]. This was done to prevent interference from the actual *models* observed by the experts.

Mutual Gaze (MG): In the AFIT sessions, different participants controlled the floor at different times (i.e., leading the discussion for the moment). When the control passed from one participant to the next, there was a mutual gaze exchange between the current holder of the floor to the next.

Gaze Coalition (GC): It was discovered in *AFIT 1* that the social interaction amongst the participants had as much to do with the outcome of the meeting as the specific merits of the scholarship candidates being discussed. The participants dynamically formed coalitions to support each-other's candidates through a process of mutual gaze fixations and back-channel expressions of support during discussions [89].

A coalition to support a proposed idea is initialized when the proposer seeks to make eye-contact with other participants while he is presenting the idea. Participants who supported the idea would return the eye-contact, while those who disagreed would avert gaze. When a return gaze was attained, the presenter's gaze moved to another member.

Floor Control (FC): During a session, a participant would gain floor control through a hand movement (gesture) and start speaking. This was deemed 'floor capturing' in [21].

Turn Taking (TT): As detailed above, each session had a meeting manager who normally was the dominant participant and facilitated the meeting. When a participant sought to take a turn speaking, the participant might look at the meeting manager while the current floor controller spoke. Once the current floor controller finished speaking, the participant seeking a turn would then begin speaking.

F-formation (Ff): F-formations were observed throughout the sessions. The defining behavior for the F-formations observed were concurrent focus on the same person (or object) or an action causing such synchronized focus.

Tier 3 *Models*: In *AFIT 1* a GC *model* was known to exist from unpublished analysis, and FC *models* were reported in [21]. In *AFIT 2*, MG, FC, TT, and a Ff *model* were reported in [21, 88]. For the FC and MG *models*, the exact actors are unknown. Therefore, given the participation statistics of each actor in [21, 88] we choose the actors with the most and least frequent activity (e.g., most gaze events or least frequent gaze target). Note that some *models* for Tier 3 in Figure 4.18 have numbered, grey *S-I's*. These represent the pieces of the *model*

our evolution strategy aimed to identify. Since we did not know how the structure existed in the data, we tried these relational possibilities. They also represent the key part of the *model* to discover from the initial seed, hence, the complete structure was not sought after their discovery. The results discussed later use these numbers to identify which structure(s) were found. Also, the seed for TT differs in this tier because of how the *model* exists in the data (this was discovered during our discovery/evolution process). At first, our timing window did not allow discovery of the *model* with the original seed. We then modified the seed to the relevant part of TT: transition from one participant to another (correlation between F's gaze start, G's ending speech, and F starting to speak).

Results

The performance of our approach was tested in several ways. First, we observe a convergence behavior of our algorithm during the evolution process. Then we perform a power/penalty analysis. For Tiers 1 and 2, each algorithm was run on 36 *models* (72 total) and for Tier 3, our algorithm ran on 52 *models* and FEM for 10; in total 134 *models* were explored. For simplicity, we use the naming schema *X_Y_Z* to reference each *model* where *X* is the data-set, *Y* is the *model* abbreviation, and *Z* is the *model* variation. For example, *A1_MG_A* is mutual gaze *model* A from *AFIT 1*.

Convergence: During our experiments, we observed a convergence behavior of our algorithm. Overall, each *1-step* change decreased the number of identified situated instances (Figure 4.19). The convergence of each *model* is grouped by the number of *1-step* changes

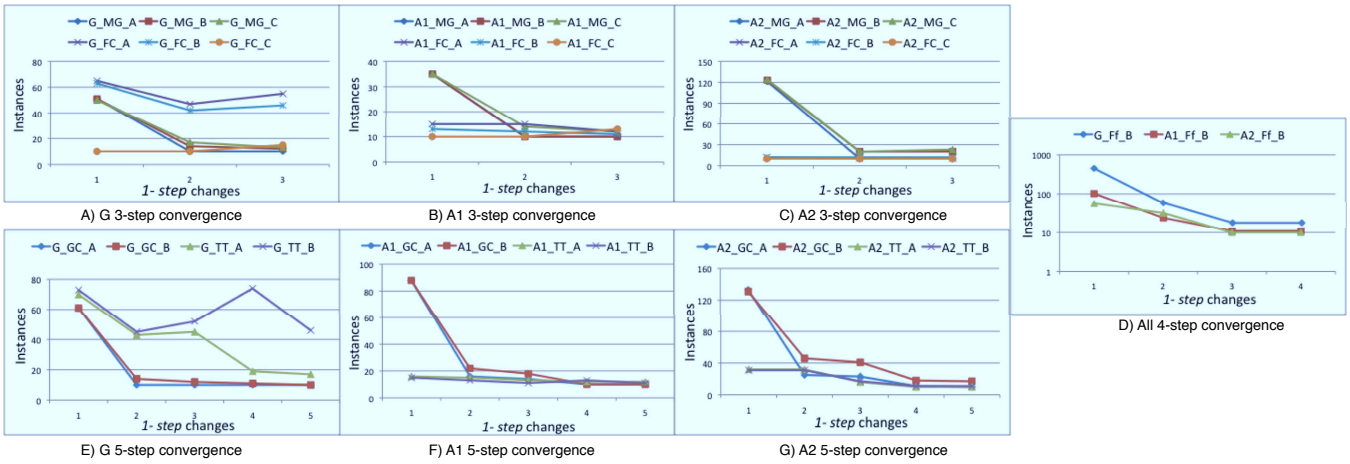


Figure 4.19: Tier 1 and 2 convergence for *model* evolution (note scale differences).

needed and the data-set. Convergence in most cases was obtained, even if the resulting instances identified were more than the 10 inserted. Outliers were seen in Figure 4.19A (*G_FC_A* and *G_FC_B*) and 4.19E (*G_TT_B*). This is due to the large number of speech events in the generated data-set and a logic error later found in our algorithm which is currently being addressed. The ordered relations with the speech *S-I's* in these *models* are flexible (*previous* and *next*) leading to many matches. The more constraining ordered relation *current* aided with convergence. This is why *G_TT_A* and *G_FC_C* had better convergence. Even though some *models* did not converge to the desired 10, a nicely pruned sub-set resulted. Also, there was no need for a convergence graph for *Ff_A* as the initial seed was the desired structure, converging instantly, exemplifying how an informed starting point can significantly aid in relevant instance identification.

Power/Penalty Analysis: For describing the results, we use power/penalty analysis of [119] which reports a power and penalty percentage. If there are x known instances of a

phenomenon in a data-set, y instances identified by a method or algorithm, and z number of instances common amongst the known and identified, $z \leq x$, then the power percentage is $z/x * 100$. For example, if $x = 10$, $y = 18$ and $z = 7$, then the power is $7/10 * 100 = 70\%$, i.e., the method's power is 70% in identifying the relevant instances. The other 11 identified instances are part of the penalty. These are the extra instances the expert must go through and, in turn, are an extra cost. The penalty $= (y - z)/y * 100$, in our example, penalty $= (18 - 7)/18 * 100 = 61.11\%$. However, power/penalty is the same as precision/recall mathematically. Power is the same as recall and penalty is 1-precision. A social scientist does not think in terms of precision/recall but in the power of a *model/pattern* in identifying what is relevant and the penalty is anything extra he/she needs to sift through.

The overall results can be seen in Table 4.8. Our approach was able to identify all inserted *models* (Tier 1-3), giving it 100% power. For FEM, in Tier 1 and 2, all but 6 *models* (Figure 4.21) had a 100% power rating. The most feasible explanation for the two with the lowest power (*A2_Ff_A* and *G_FC_C*) is these *models* included *S-I's* that were "equal" and the *models* that were identified by FEM were matches that coincidently included pieces of the inserted *models* (i.e., the start *S-I* of a few of the inserted *models*). Another observation for *A2_Ff_A* is FEM found 18 instances but none of them were the *models* inserted. What was found were other occurrences of the three events in the *model* within 3 seconds of each other. Since the FEM algorithm only considers start *S-I's* of an interval, this explains the trouble with *model* *A2_TT_B*. In some cases, the start *S-I's* of the *model* were temporally too far away. For the other three, the reason is likely the high number of gaze and speech

Table 4.8: Overall Power/Penalty Analysis results.

	Tier 1	Tier 2	Tier 3
Misses			
Ours	0	0	0
FEM	30 of 120	65 of 240	63 of 208
Mean Power			
Ours	100	100	100
FEM	75	72.92	71.48
Mean Penalty (Precision)			
Ours	33.09 (66.91)	11.41 (88.59)	N/A
FEM	62.26 (37.74)	37.72 (62.28)	41.74 (58.26)

events in the generated data-set and the high number of gaze events in *AFIT 2*. The high concentration of such events were observed to cause some challenges for both algorithms.

The penalty percentage results can be seen in Figure 4.20. *Model* names (x-axis) with an '*' in 4.20 are *models* identified using *S-I's* in the FEM algorithm. The rest were identified with an interval representation (reason discussed in Section 4.4.5). Penalty of 100% meant none of the inserted *model* occurrences were identified or the algorithm failed (result of its limitations). Each graph compares the penalties of each data-set. For the generated data-set, our approach's penalty was lower than FEM's for 9 of the 12 *models*. For *AFIT 1*, our approach had comparable penalty to FEM, however, FEM had 100% penalty for 3 *models* and our max penalty was 23.08% (mean: 8.36%). For *AFIT 2*, our approach had a lower penalty for 8 of the 12 *models*. Overall, our approach had zero penalty for 15 *models* while FEM had zero for 5.

For Tier 3, there were only three known locations of ground truth *models* which our approach found. For the other *models*, the instance count ranged from 1 to 33. We could only compute

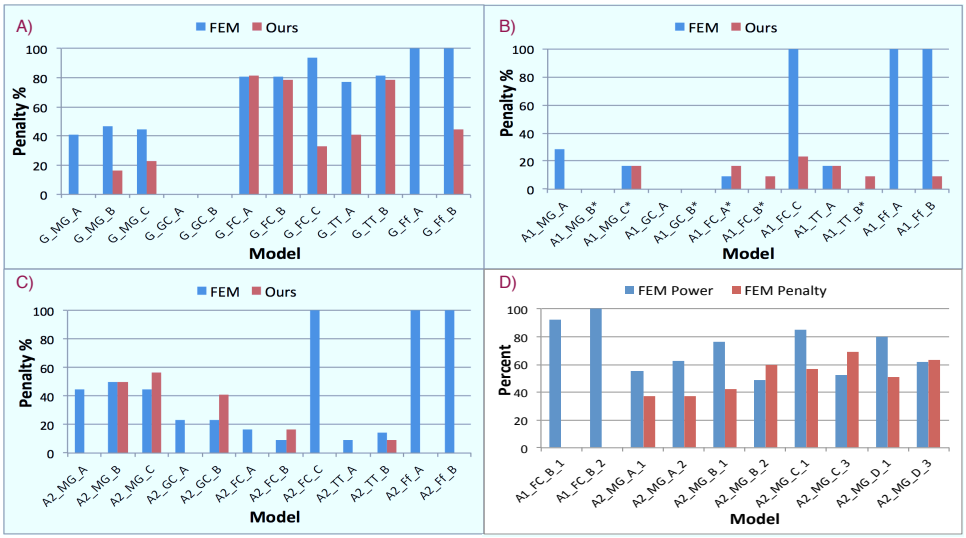


Figure 4.20: A) Tier 1 Penalty. B) Tier 2 A1 Penalty. C)Tier 2 A2 Penalty. D) FEM Tier 3 Power and Penalty as compared to our Tier 3 results.

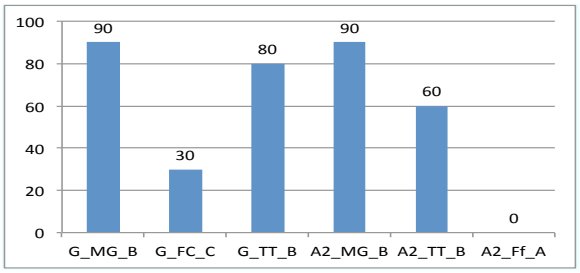


Figure 4.21: FEM *models* with Power percentage less than 100.

the power/penalty of FEM as compared to our approach's results, hence, Table 4.8, Tier 3 is FEM's results as compared to ours. However, given our approach's results for Tier 1 and 2, we are confident in its Tier 3 results. Since FEM's purpose is mining frequency, we did not attempt to identify instance occurrences less than 8. Doing so caused an explosion of results. In Figure 4.20D, we see the power/penalty analysis of FEM compared to our Tier 3 results. As can be seen, FEM had a fairly high penalty with an adequate power.

Discussion

Overall, we observed our approach performed very well compared to the FEM algorithm. We were able to provide a solution with high power and substantially less penalty than the frequency approach of FEM. There were some cases where FEM performed better, however, the successful use of *S-I* granularity allowed our approach to identify all *models*. The *models* FEM did not identify can be explained by limitations discovered during experimentation. It was unable to identify any *S-I's* occurring at the same time or intervals sharing a start time. The FEM algorithm did not appear to be built for this situation. When creating the *models* to insert into the data-sets, if two *S-I's* were "equal", we set their times to be the same. However, in real *model* instances, this may not occur due to granularity limitations of the recording equipment. But, it may at the discretion of whomever is creating the data-set. Also, most of the FEM results were from the interval version of the data-sets since when using *S-I's*, too many results were returned (on the order of 40-100K) which overloaded the output functionality of the FEM program or the program ran out of memory on machines with 8 and 10GB of RAM. Overall, significant effort was required to tune the FEM algorithm for our purpose.

In order to identify the relevant *models* within the FEM results, an expert would have to search through them. Arguably, an expert could seek out relevant *models* within the FEM results, however, doing so is non-trivial as she would need to define a starting point of interest and sift through the results (which is the motivation of this paper). Hence, an expert would need an approach similar to ours anyway.

As for identifying more instances than inserted (penalty %), the cause was a combination of several aspects of the data and tuning of the algorithms. Several of the data-sets had a large percentage of specific event types (e.g., gaze or speech). Matches for the algorithms also were on a 3 second window between each *S-I* or start time of an interval. Hence, the mixture of a high number of certain event types and the time window allowed for many matches. The number of events cannot be controlled (unless some filtering is introduced), however, the time window can be tuned for the specific data-set and characteristics of the *models* sought. Our approach aided in limiting *suggestions* at each evolution iteration by reporting them within a 1 second window beyond the equality window. However, during instance identification between *1-step* changes, a 3 second window is used allowing flexibility of matching but possibly causing many results. We also observed (notably for Tier 3 *A2_Ff_A*) the necessity to have dynamically controlled timing windows as the *model* sought was slightly out of range.

We observed for our approach that several aspects of the *model* structure aided in identification of the relevant *models*. The *current* relation caused better convergence during the evolution process. This makes sense as it is the relation that is the tightest, hence, including this relation in the initial seed (if feasible) facilitates *model* identification. Also, the initial seed may actually lead the expert to the *model* instances desired as observed in *Ff_A*'s case.

During the evolution process, the probabilities for the *S-I's* sought were generally high, however, they were not so in some cases. What an event represented (conceptually) aided in choosing. This was true even when the probability was high exemplifying that the importance lies in the expert's interest and that we provide an informed view of the data. In some cases,

completing an interval (both start and end *S-I's*) increased or decreased the number of instances found. Hence, the usage of partial knowledge through *S-I's* can be advantageous depending on the *behavior model* sought. Sometimes, an expert is only interested in when a behavior (event) starts or ends in relation to another behavior. This was observed for Tier 3 $A2\_TT\_A$.

We chose not to perform a run-time comparison since each approach comprises two different timings. For our approach, the first is the processing time for instance identification and *suggestion* processing. The second is how long the user spends choosing a *1-step* change, sparking the next processing step. This time is relative. For FEM, the first is a processing time for identifying frequent *models* (automatic). The second is the user looking through the list of *models* manually, hoping to find what is wanted (assuming this is known). Although, the initial processing time would seem to be relevant for comparison, our method processes a sub-set of the data-set with each evolution step instead of FEM's one-time processing of the whole data-set making meaningful comparison difficult. In light of these differences, such timing information may not be beneficial or informative.

The data we operate on have varying aspects, such as uncertainty or granularity of the annotations. Such aspects are at the discretion of the creator of the data. For uncertainty, or "fuzziness", this may be a uncertainty score (e.g., $0 \leq e \leq 1$) associated with each annotation. For granularity, this is the level of detail captured by the annotations. This does not affect the processing but the level of meaning possible from the annotations. For example, the resolution of an image determines the detail level extractable from the image.

A 640x480 image and a 2560x1440 image each provides a certain level of detail in which the latter can potentially provide more information. The resolution of the annotations dictates the resolution of the resulting *model(s)*. Overall, what is provided dictates what can be done and represented.

4.4.6 Conclusion and Future Work

In this paper we presented a *model* evolution technique for identifying relevant temporal *behavior models* within temporal event data. We demonstrated the accurate identification of relevant *models* while comparing against a well established temporal mining algorithm (FEM). We found our approach was well suited for the data domain and, through power/penalty analysis, we saw our approach performed overall better than FEM. We demonstrated that our approach is well suited for relevant *model* instance identification necessary in situated behavior analysis and in doing so, displayed its ability to identify relevant *models* amongst extracted events from a media source. However, our approach can be applied to a wider scope of temporal event data extracted from media in which challenging questions about the media contents may be answered. One such domain we will be applying our approach to is a news broadcast corpus consisting of several news networks over multiple years. We will be able to analyze topic and transition structure amongst the networks, how different news shows transition between topics, and the structure of anchor/reporter interactions for different topics. The creation of this corpus is in progress.

For future development, we identified implementation improvements which should decrease erroneous *model* instance matches allowing more robust convergence. Supporting other aspects of the data, such as uncertainty, is being investigated. We are currently developing a set of temporal relationship principles for defining interactions amongst *S-I's* for *behavior models*. This paper used a subset with extensions under development.

4.4.7 Acknowledgments

We would like to thank Debprakash Patnaik for his help in using FEM and Christa Miller for invaluable input as an outside observer and editor. This research was partially funded by FODAVA grant CCF-0937133 and NSF IIS-1053039.

Chapter 5

Evaluation

In this chapter we discuss the evaluation of our approach. This evaluation comprises of four phases where the first three are satisfied in this dissertation and the fourth is future work. The *first* phase consists of precision/recall testing (known as power/penalty analysis to some social scientists) in a controlled environment on several similar datasets. The *second* is application to multiple datasets. The *third* is verifying the relevance of instances identified through asking experts. *Lastly*, compare our results on a particular dataset to that of *models* identified by users. In the following sections, each is described in more detail.

5.1 Phase 1 Evaluation

The bulk of the work completed for this dissertation was done in preparation for this phase. Before evaluation could be done, investigation was needed into how to provide event rankings

for suggested additions to a *model/pattern* [93, 95]. We also had to investigate aspects of situated analysis [92]. After this was done, we had enough knowledge to implement a system and perform this first phase of evaluation.

The nature of our problem domain made experimentation a challenge. Quantitative metrics are difficult to devise for evaluation of *model* discovery that is dependent on an expert's subjective interest. In order to verify the accuracy of our approach in identifying *model* occurrences, data-sets with known ground truth (i.e., known relevant *models*) are required. Such *models* must have significant interest to an expert. Hence, to obtain a controlled environment to measure results, we introduce occurrences of *models* based on known relevant *models* into data-sets at known locations and then test to see if the occurrences can be identified. Each experiment used 5 data-sets: one synthetic with introduced *models*, two real data-sets with introduced *models*, and the same real data-sets unaltered. The real datasets (*AFIT 1* and *AFIT 2*) consist of multimodal annotations of several meeting sessions of five Airforce Officers from the Airforce Institute of Technology (AFIT) partaking in war-gaming scenarios (meeting room setting). The *models* introduced are based on *behavior models* deemed important by experts in the analysis of the *AFIT* data-sets [24, 88]. More details on the AFIT datasets can be seen in Section 4.4.5.

We measured performance using *power/penalty* analysis of [119] which reports a power and penalty percentage. If there are x known instances of a phenomenon in a data-set, y instances identified by a method, and z number of instances common amongst the known and identified, $z \leq x$, then the power percentage is $z/x * 100$ and penalty percentage is $(y - z)/y * 100$.

Power/penalty is the same as precision/recall mathematically. Power is the same as recall and penalty is 1-precision. A social scientist does not think in terms of precision/recall but in the power of a *model*/*pattern* in identifying what is relevant and the penalty is anything extra he/she needs to sift through.

Our initial experiments focused on comparing our approach with similar symbolic temporal pattern mining methods. These experiments were completed in two steps. The first compared our *model* evolution technique to that of a FEM algorithm using a frequency threshold. Published results can be seen in [93], and also seen in Section 4.4.5. Our approach was able to perform comparable to that of the FEM algorithms with an overall better performance. These results were very encouraging as supporting our approach as a viable method for relevant *model* identification within temporal event data.

This success lead to our second set of experiments where we compared our search core (STIS) to a FEM algorithm using frequency count of a defined *model*, i.e., count the number of instances of a particular *model*, and one using conditional probability. Published results can be seen in [91], and also seen in Section 4.3.4. Once again, our approach was able to perform comparable to that of the FEM algorithms with an overall better performance.

These two sets of results demonstrated the viability of our approach to identify relevant *models* within temporal event data. Hence, we proceeded with the next phase of our evaluation discussed in the next section. To prepare for the next phase, we took what we learned from the above experiments and improved our system. The details as to what was improved can be seen in Section 6.3.

5.2 Phase 2 and 3 Evaluation

Both phase 2 and 3 will be addressed through use-cases. These use-cases consisted of working closely with three fellow student researchers who were interested in analyzing their own unique human behavior and interaction datasets. The use-cases were approved by the Virginia Tech Institutional Review Board (IRB) under IRB number $12-1005$ and the approval letter can be seen in Appendix C, Figure C.1 and C.2.

Our approach presented in this dissertation is motivated from experts viewing and analyzing data by analysis of temporal relationships between events, their order, and what the event orders conceptually mean. For simplicity, we will call this analysis approach Temporal Event Behavior Analysis (TEBA). The motivating literature for this analysis approach is [122, 124, 89, 88, 24]. However, the student researchers we collaborated with were not experts in TEBA but experts in their data. We hypothesize that supporting experts with an approach motivated from TEBA will aid them in identifying and discovering relevant *models* in their data. This will show three results: 1) provide further support for TEBA as an analysis approach, 2) that experts not as familiar with TEBA can benefit from it, and 3) our support for TEBA aids in analysis. We realize statistical significance cannot be attained through three participants, however, through a longitudinal study, we will show that our approach to supporting TEBA has promise that merits further study. Plus, through it we will also satisfy our phase 2 and 3 evaluation requirements.

We first describe the demographics of the student researchers followed by the details of each

of their datasets. After which, the methodology of our use-cases is discussed. Next, the results are presented leading to a discussion of the outcome of the use-cases. We close this section with discussion. The presented work in this section was accomplished using *version 2* of our system detailed in Section 6.3.

5.2.1 Demographics

Three student researchers within Human-Computer Interaction at Virginia Tech were independently recruited. These students are not part of our research group meaning they were not aware of the presented research in this dissertation. Hence, presentation of this analysis approach was new to them. Because of this, we were able to offer a way of analyzing their data that is different from other approaches and allow them to identify meaningful analysis results not (as easily) possible through other approaches.

The demographics of the student researchers can be seen in Table 5.1. This information was gathered from a background questionnaire given at the beginning of the use-cases (Figure C.3). Each participant had conducted research for at least 1.5 years using some form of data analysis prior to working with us. These prior analysis were using standard analysis techniques and software packages.

Table 5.1: Use-Case Participant Demographics

Participant	Gender	Age	Academic Status	Research Experience	Previously Conducted Data Analysis	Previously Used Data Analysis/Visualization Software
P1	Male	28	Masters	2 years	Study of self-report data through transcription and coding	Statistical packages (e.g., R and JMP) plus Excel
P2	Male	23	PhD	1.5 years	Statistical and visual, Spotfire	JMP, SAS, and Spotfire
P3	Female	24	PhD	2.5+ years	Text analytics, geospatial, quantitative analysis, multimedia analysis, social network	Jigsaw, IN-SPIRE, Canopy, Palantir, Force-SPIRE, Analyst's Workspace, Excel, JMP, MySQL, Tableau/Eureka, Spotfire, Light_SPIRE

5.2.2 Datasets

In this section we describe the datasets of each of the three student researchers who participated in our use-case. For simplicity, we will refer to the student researchers as P1, P2, and P3. Each participant's data needed some formatting before our system could use it. We developed a separate program that would take as input the participant's data in various forms and output the representation (SQLite database) that our system uses. The details of this formatting is not presented, however, we will describe in detail throughout this chapter the multiple, faceted characteristics of their data. We will first describe their data characteristics in general.

Data Characteristics: The data of each participant are multi-channel events represented by either time points or intervals (or a mixture). The events were annotated from their media sources either automatically and/or manually. Each channel represents events of a certain type (i.e., event type). This is illustrated in Figure 5.1. This kind of data is also known as multimodal data [92, 132, 140]. Our approach, Interactive Relevance Search and Modeling (IRSM) operates at the semi-interval level (an intervals start and end points) as

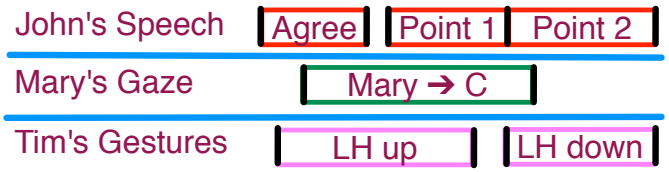


Figure 5.1: Example of multi-channel event data (semi-intervals highlighted as vertical bold lines)

Table 5.2: Use-Case Datasets' Contents Overview

	Semi-intervals	Unique Semi-intervals	Channel Min	Channel Mean	Channel Max
P1 Original	2218	252(max)	7	14.22	23
P1 Filtered and Clustered	1305	6	3	3	3
P2 Original and Normalized	2784	6	3	3.83	4
P3 Original	8545	25(max)	10	12.43	14
P3 Filtered	1163	25(max)	10	11.57	14

described in [39] and successfully used for unsupervised pattern mining in [101].

P1's data: P1's data consisted of 23 sessions with each session consisting of three participants given the task to collaboratively build a story from pictures to describe the design of a new dining hall. Each participant had their own laptop in which a shared space and a private space were provided for viewing and placing pictures. The participants took turns contributing to the shared space. Each session was video recorded and transcribed for contributing features to the story. The data of each session consists of a sequence of events depicting when in time each participant (A, B, and C) contributed a feature. Table 5.2 provides an overview of the dataset contents. Here is provided the total number of semi-interval events across all sessions, the max number of unique semi-intervals per session, and the minimum, mean, and maximum number of channels across all sessions. A more detailed look can be viewed in Table 5.3 where a breakdown across all sessions is provided. A visualization overview of all sessions can be seen in Figure 5.8, top-left. P1's analysis focus was on identifying interrup-

Table 5.3: P1's Detailed Datasets' Contents

P1 Original

Session	Total Semi-intervals	Unique Semi-intervals	A's Semi-intervals	B's Semi-intervals	C's Semi-intervals	Channel Count
1	124	124	30	36	58	19
2	108	108	18	40	50	16
3	38	38	12	14	12	10
4	94	94	52	14	28	12
5	74	74	26	30	18	17
6	78	78	18	34	26	10
7	74	74	24	24	26	14
8	252	252	78	58	116	17
9	108	108	30	42	36	14
10	54	54	10	30	14	12
11	126	126	58	32	36	23
12	170	170	50	82	38	15
13	84	84	34	34	16	15
14	36	36	16	12	8	7
15	56	54	24	20	12	9
16	154	154	48	62	44	22
17	94	94	38	36	20	18
18	130	130	26	78	26	16
19	90	90	38	36	16	13
20	84	84	30	22	32	14
21	46	46	14	18	14	11
22	88	88	26	36	26	13
23	56	56	24	22	10	10

P1 Filtered and Clustered

Session	Total Semi-intervals	Unique Semi-intervals	A's Semi-intervals	B's Semi-intervals	C's Semi-intervals	Channel Count
1	62	6	18	18	26	3
2	75	6	14	30	31	3
3	34	6	8	14	12	3
4	48	6	24	10	14	3
5	46	6	16	16	14	3
6	66	6	16	30	20	3
7	44	6	10	12	22	3
8	106	6	32	28	46	3
9	72	6	18	28	26	3
10	48	6	10	24	14	3
11	66	6	30	20	16	3
12	56	6	22	20	14	3
13	56	6	26	18	12	3
14	34	6	16	12	6	3
15	48	6	16	20	12	3
16	92	6	30	30	32	3
17	32	6	18	10	4	3
18	74	6	16	38	20	3
19	52	6	20	18	14	3
20	52	6	20	12	20	3
21	40	6	12	14	14	3
22	68	6	22	22	24	3
23	34	6	12	16	6	3

tions or out-of-turn instances that exhibited collaborative behavior. Hence, he hypothesized that by looking at *models* of contributions that did not follow the simple turn-taking of the group, he could find evidence of collaboration among the participants.

P2's data: P2 was studying a new multi-scale interaction technique for large, high-resolution displays. The interaction technique consisted of using 1, 2, or 3 fingers on a trackpad to control the speed of the cursor, e.g., 1 finger is normal speed, 2 is faster, and 3 is fastest. There were 8 sessions where each session consisted of three trials where participants used a combination of 1, 2, or 3 fingers (according to participants personal choice) to reach targets that appear on the display. Once a target is reached, a new one appears elsewhere on the display. Each trial consisted of 17 targets. Event logging was used during each trial to record the different finger modes used by each participant. The events recorded from the logging were used to create time sequential intervals representing the finger mode used at

Table 5.4: P2's Detailed Datasets' Contents

P2 Original and Normalized

Session/Trial	Total Semi-intervals	Unique Semi-intervals	F1 Semi-intervals	F2 Semi-intervals	F3 Semi-intervals	NT Semi-intervals	Channels
User 1/Trial 1	96	8	34	16	12	34	4
User 1/Trial 2	126	8	40	36	16	34	4
User 1/Trial 3	134	8	42	36	22	34	4
User 2/Trial 1	110	8	40	24	12	34	4
User 2/Trial 2	112	8	38	28	12	34	4
User 2/Trial 3	118	8	40	32	12	34	4
User 3/Trial 1	116	8	48	32	2	34	4
User 3/Trial 2	122	8	50	36	2	34	4
User 3/Trial 3	108	8	46	22	6	34	4
User 4/Trial 1	108	8	36	26	12	34	4
User 4/Trial 2	130	8	36	40	20	34	4
User 4/Trial 3	124	8	40	36	14	34	4
User 5/Trial 1	110	8	36	26	14	34	4
User 5/Trial 2	102	8	36	20	12	34	4
User 5/Trial 3	110	8	38	28	10	34	4
User 6/Trial 1	112	8	42	20	16	34	4
User 6/Trial 2	132	8	50	32	16	34	4
User 6/Trial 3	140	8	44	40	22	34	4
User 7/Trial 1	116	8	32	34	16	34	4
User 7/Trial 2	82	8	6	32	10	34	4
User 7/Trial 3	96	6	16	46	N/A	34	3
User 8/Trial 1	134	6	60	40	N/A	34	3
User 8/Trial 2	112	6	54	24	N/A	34	3
User 8/Trial 3	134	6	64	36	N/A	34	3

a given time. Table 5.2 provides an overview of the dataset contents. Here is provided the total number of semi-interval events across all sessions, the number of unique semi-intervals per session, and the minimum, mean, and maximum number of channels across all sessions. A more detailed look can be viewed in Table 5.4 where a breakdown across all sessions is provided. Here F1, F2, F3, and NT represent finger mode 1, 2, 3, and new target, respectively. A visualization overview of all sessions can be seen in Figure 5.8, middle-left. P2's analysis focus was identifying finger mode trends/behaviors among the participants (*models*) that explain good/poor performance. Hence, he hypothesized that participants with good performance had different finger mode behaviors than that of participants who did not perform as well.

P3's data: P3 was studying cooperative use of a large, high-resolution display for performing an intelligence analysis task. There were 7 sessions consisting of 2 people per session that shared a large, high-resolution display, each with their own mouse for control. All sessions were video recorded from which annotations were hand coded for apology events, possessive speech events, location discussion events, significant speech events, and events for re-finding either by computer or physically on the display. A mouse log was also created during each session for the pair of participants, however, P3 choose to only process the mouse logs and create events for three of her sessions as she was unsure if they would be useful. Inclusion of the three sets of mouse log data was a test to verify if inclusion of the mouse data from the other sessions would be beneficial. Details for this is discussed further in Section 5.2.7. Table 5.2 provides an overview of the dataset contents. Here is provided the total number of events across all sessions, the max number of unique semi-intervals per session, and the minimum, mean, and maximum number of channels across all sessions. A more detailed look can be viewed in Table 5.5 where a breakdown across all sessions is provided. S1 and S2 are the different participants in each group. A visualization overview of all sessions can be seen in Figure 5.8, bottom-left. P3's analysis focus was whether the display employed would be instrumental in facilitating common ground among the pair of participants in each session. Hence, she hypothesized that the display would serve as a medium for common ground.

There was an eighth group in P3's data (G), however, during this group's session, there was a malfunction of the system they were using. This required a system reboot and the moderator for the session stopped the video recording during the reboot. This caused the

Table 5.5: P3's Detailed Datasets' Contents

P3 Original

Group	Total Semi-intervals	Unique Semi-intervals	S1's	S2's	Channel Count
A	1829	24	790	1031	13
B	4020	22	1908	2094	14
C	1944	25	808	1117	14
D	333	25	201	123	14
E	88	20	37	30	12
F	69	14	19	38	10
H	262	18	131	96	10

P3 Filtered

Group	Total Semi-intervals	Unique Semi-intervals	S1's	S2's	Channel Count
A	158	16	79	71	11
B	116	14	27	71	12
C	137	17	79	39	12
D	333	25	201	123	14
E	88	20	37	30	12
F	69	14	19	38	10
H	262	18	131	96	10

data to be segmented into two sections which posed more of a challenged than expected to create one continuous session to process by our system. Due to this complication, P3 decided not to include analysis of group G as correcting the data would take too long. However, she does plan to continue analysis of her data using our system which does include group G (discussed in Section 7.3).

5.2.3 Methodology

In conducting our use-cases, several types of sessions were conducted. Each session had a moderator that ran the system and worked with each participant. The screen of the computer used was captured and event logging performed. After each session, a semi-structured interview with the participant was conducted. The questions asked during the interview can be seen in Appendix C. We started working with our participants 2-4 months before any sessions were held. This time was used to learn about each participants data and to transform it into the proper format. The details of each type of session is discussed in turn.

Pilot Sessions: We began with pilot sessions. These were meant to test the approach

with full cooperative help from the moderator of the sessions. The moderator helped them analyze a subset of their data. At the beginning of the pilot, each student was given a brief introduction to the analysis system. After which, several more sessions were conducted where the moderator worked closely with the participant. Since the pilot sessions were exploratory, there was no initial limit as to the number of sessions. We conducted enough to learn what was needed to proceed. We ended up running five sessions for P1 and two for P2 over a period of one week. Each session ranged from 30 minutes to 2 hours.

Training Sessions: We learned from the pilot sessions that there was a need to minimize the influence of the moderator. At the time of the pilot sessions, IRSM was fully functional but the user interface (UI) was not meant for independent use by participants. IRSM needed an operator. We were interested in testing IRSM (the approach) and not the UI, hence, the moderator became part of the UI as the system operator. The moderator could take commands and requests and perform them, and, in doing so, fulfilled the functionality that allowed the participants to utilize IRSM.

After the pilots, we created a training script (see Appendix C) so the participants could learn to use IRSM independent of the moderator's input. The moderator would then act only as the system operator. Three training sessions were conducted for each participant. The first training session consisted of going over the detailed training script that was purposed to familiarize each participant with IRSM and its capabilities. Since this first training session was purely for training, no semi-structured interview was performed at the end. Also, no screen capture was performed during the training sessions. Each participant was

also provided with a *feature list* (see Appendix C) for reference that summarized IRSM's capabilities. The second and third training sessions consisted of each participant analyzing their data with minimal input and help from the moderator. The participants were allowed (and encouraged) to ask the moderator any questions they had.

Independent Sessions: After training, the participants performed independent sessions where the moderator provided no help unless necessary. The sole purpose of the moderator was to run the system (be part of the UI) and to take notes. Four independent sessions were run for each participant. The one exception was P2 who was satisfied with his results by the end of his second independent session, and hence, only two independent sessions were run for P2. The training and independent sessions were conducted over a period of four weeks.

Sessions' Structure: The sessions where the participants performed analysis of their data followed a general structure. Each participant would start with initial *model(s)* either based on an initial hypothesis or from a prior session. The *model(s)* would be tweaked to find matches in the data. Identification of matches sometimes lead to a new understanding of the *model* and potential variations. The variations were explored, leading to modifying the *model(s)* and finding further matches to the modified *model(s)*. Hence, a cycle of analysis was created. At certain times, different *formatting strategies* (Section 4.3.4) were applied when necessary. Each training and independent session ranged from 30 minutes to 1 hour.

5.2.4 Gathering Results

The results presented were gathered through several means throughout the use-cases. The moderator of each session carefully observed the participants actions and took notes. The screen capture videos were consulted when necessary. We transcribed the semi-structured interviews and annotated the transcriptions based on categorical observations. Also, after all participants concluded their sessions, we asked some follow-up questions to capture a summary of their experience with our approach and what they learned. The follow-up questions can be seen in Figure C.7 in Appendix C. Through our observations, notes, screen captures, and the annotated transcriptions, we were able to provide the below presented results and observations.

5.2.5 Results

The main focus of the use-cases was to see if the instances matching the participants' *models* were relevant to them. We focus on answering this question first. After each session (training and independent), we asked each participant to rank the relevance of the instances found on a 5-point Likert scale (strongly disagree (1), disagree (2), neutral (3), agree (4) and strongly agree (5)). The results can be seen in Figure 5.2. We see that the mean for both training, independent, and overall are all the same, 4.75. This is encouraging as our participants deemed the results providing by our system relevant to them.

We should note why P2 did not have any rankings for his independent sessions. First, he

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P1	5	5	5	5	5	4	4	4.5
P2	4.5	5	4.75	N/A	N/A			N/A
P3	4	5	4.5	5	5	5	5	5
		TS Total Mean	4.75				IS Total Mean	4.75
							Total Mean	4.75

Figure 5.2: 5-point Likert scale results for relevance of instance matches of the participants' *models*.

was content with the results of the analysis of his data at the end of independent session 2, or IS 2. Hence, why P2 did not partake in an IS 3 or IS 4. Next, P2's approach using our system was more focused on whether a *model* existed or not. Either way it was important to him. Hence, he felt the the question about relevance of instance matches did not apply to him. During training (TS 2 and TS 3), P2 was looking at the instance matches and rating their relevance. However, once P2 reached his independent sessions, he had figured out a strategy to answer his hypothesis that did not required looking at the instance matches but the absence or presence of matches. This explains why P2 did and sometimes did not answer this question. P2 was using our system to answer a hypothesis by seeing what was and was not present in the data. So, indirectly what was found (or not found) was relevant to him.

The other two participants (P1 and P3) were very happy with the instances matching their *models*. They were able to find evidence for their hypotheses through the matches without needing to manually search through all of their data. Our system was able to focus on a small subset of their data that matched their *models*.

The rest of the results focus on two aspects of our approach we were interested in testing.

The *first* is analysis strategies developed by our participants. We are interested in seeing the kinds of analysis strategies developed and how they are similar and different from traditional TEBA. This includes how each participant approached their data according to their hypotheses and how they navigated datasets with challenging temporal characteristics. The *second* is discussing the use of the software and the results of its functionality. Here we want to see if the features and functionality of our system and approach aided the participants in their discovery task and if the software was used according to its intended use or did participants use features and functionality in unexpected ways. We also report pros and cons presented by our participants.

5.2.6 Strategies of Analysis

Here we begin our presentation of the analysis strategies developed by our participants. Each Participant exhibited different strategies for analysis. Some participant's strategies had similarities with other participant's. We will discuss the strategies of each participant individually first, then proceed to compare similarities and differences.

We noticed there was no real discernible difference between the training and independent sessions in terms of analysis progress. The main difference we noticed was during the independent sessions, each participant took a little time to remember certain functions of the system and how to perform some actions. The only time we had to intervene during any independent sessions was for P3 and this just consisted of making sure P3 was aware of some

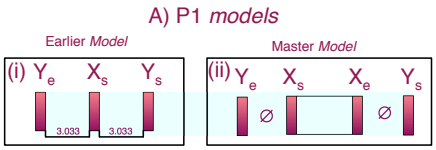
functionalities and remind her of some temporal constraints that were set.

P1 started out with trying to figure out how to define a *model* that would capture an interrupting/out-of-turn behavior in his data. He solely used predicate *models*, as opposed to descriptive *models* (described in Section 5.2.10). From his follow-up questions (Figure C.7), he noted that since his data represented turn taking, he was "..looking for places where turn taking were not followed or had *patterns* that were more complicated than simple rotation." He started out with an initial *model*, saw how it matched the data, and looked at the matching instances. Through this he learned more about the characteristics of his data and how the interrupting/out-of-turn behavior can exist in his data. As a result, he was able to refine the specified initial *model*. One of his initial *models* can be seen in Figure 5.3A. He went through several iterations until one master *model* was developed (seen in Figure 5.3A). The development of this master *model* was conducted by P1 during his pilot sessions and initially tested during his training sessions. By the time P1 was in training, he had established his master *model*. This master *model* was solely used during his independent sessions. However, during his training and independent sessions, he realized that his master *model* was only able to capture some cases. This realization came near the end of his sessions. Using his master *model*, he continued looking through the data. When matches were found, he would look at the video to verify the legitimacy (relevance) of the instances and save the relevant ones. He continued this through a majority of his sessions. Through this procedure, he was able to find evidence of collaboration through identifying interrupting/out-of-turn behavior instances.

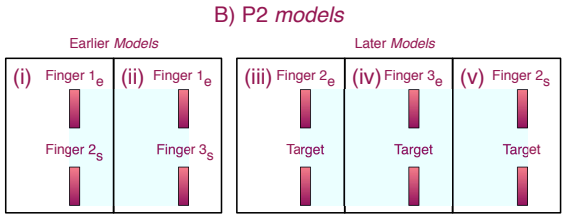
P2 defined several initial *models* to look at different behavior transitions in his data. A behavior transition in his data was looking at how his users transitioned from one finger configuration to another or what finger configuration was used to reach a target or begin the trek towards a target. He solely used descriptive *models*. P2 continued trying different transition *models* until he found a set of four *models* that were pertinent to his hypothesis (Figure 5.3B (ii)-(v)). With this set of *models*, he counted up how many times each *model* occurred in all his datasets. He compared these results to that of previous analysis he conducted to confirm/deny his hypothesis. Through this procedure, he was unable to find evidence supporting his hypothesis, leading to him to rethinking his hypothesis as it may not be correct. Once his hypothesis was refuted, he concluded his analysis.

P3 started with a few simple *models* to see how the users in her data were interacting. P3 tried a few iterations to find the *models* that were most pertinent to answering her hypothesis (this was an exploring phase). She solely used predicate *models*. During this process, she would see what existed in her data with respect to the *models* that interested her. She was not sure what was in the data in terms of her *models*, hence she went through an exploring phase until she had a better idea of what existed. Once this was completed, she defined eleven *models*. P3 then counted up the occurrences of these *models* within her sessions. She compared her results from these *models* to that of previous analysis and was able to provide evidence for her hypothesis and successfully identify *behavior models* that explained previous analysis. A progression of P3's *models* (not exhaustive) can be seen in Figure 5.3C.

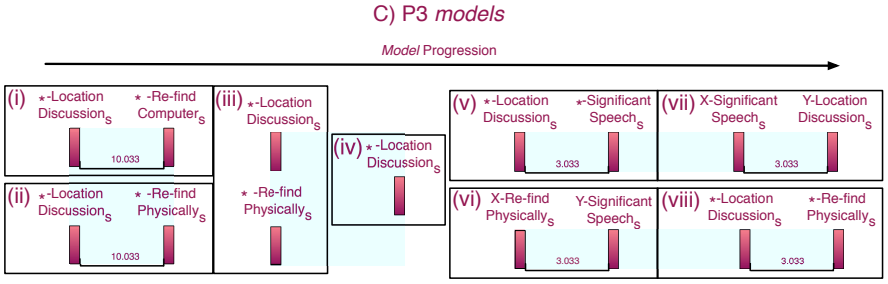
Similarities: All three participants began with an initial set of *models* representing their



Since P1 was interested in identifying interrupting/out-of-turn behavior, he began by creating an interjection *model* (i) that represented when person X interrupts person Y within a certain timeframe. This represents when X starts to speak between when Y finishes a statement and starts his/her next statement within a small timeframe. Over time, P1 learned more about how this behavior can exist which resulted in (ii). Here, the *model* represents when X interrupts Y (notice the complete interval of X) and X is the sole interrupting event between Y ending and starting his/her next statement. Relaxing the temporal constraints to focus solely on absence and not length restriction on X's interval allows the provision for a more flexible *model* that matched more instances and reflected more closely the behavior P1 was interested in identifying.



The *models* presented here show how P2 was focused earlier in his analysis on finger mode transitions (i and ii). As time went on, his later *models* (iii, iv, and v) reflected an interest in how a participant approached a target (iii and iv) and how a participant began a new target once the previous target was reached (v).



Earlier in her analysis, P3 was exploring and seeing what existed in her data. She was also experimenting with larger temporal constraints (i and ii) which she tightened up later (v through viii). The *models* she used focused on the differing interactions with the display. These included *models* representing significant speech following/preceding a location discussion of a document on the display (v and vii) or significant speech/ location discussion coupled with physical re-finding on the display. Identifying such *model* instances provided support for her hypothesis.

Figure 5.3: *Models* created by participants with accompanying descriptions of meaning. A) Earlier *model* of P1 and his master *model*. B) Earlier patterns of P2 and the later *models* used to finalize his analysis. C) *Model* progression of P3.

respective initial hypotheses, and as they learned more about their data and what to look for, they refined the *models* through re-specification of their respective initial *models(s)* to reflect what was important to them and how instances actually occur in their respective datasets. P2 and P3 followed a similar trend of using many *models* to look for different evidence. They both counted the number of times each *model* occurred in each of their datasets in order to look for evidence confirming/denying their hypotheses.

Differences: P1 ended up using a single master *model* to look for evidence supporting his hypothesis while P2 and P3 used multiple *models*. This shows that P2 and P3's focus was on multiple *behavior models* within their data while P1 was focused on one single *behavior model*. P2 solely used descriptive *models*. He appeared to be more interested in looking for *model* instances with certain values, while, P1 and P3 were not quite sure of the values but had an abstract *model* (predicate) of what was interesting to them and wanted to see what values would be bound to the predicates.

Growth Strategies: P1 and P2 did not use our abstract segmentation representation for viewing aggregate/situated suggestions. The original purpose of this was to allow the participants to look through the contexts of matching instances and add to their *model* based on the presented suggestions (i.e., grow the *model*). However, instead of "growing" in the sense we originally intended, they used the matches as informing them in how to update their *models* manually. P3 also exhibited this behavior, but P3 did use abstract segmentation to view the varying contexts of her *model* instances. In some cases, P3 used abstract segmentation in Aggregate View to grow her *model* to look at matches, then take a step back (i.e., remove

what was added) and grow another direction. We did observe that the context in which *model* instances were found did influence *model* growth, however, in ways not expected. The influence on growth was a more gradual process as our participants explored and learned about their data. Such exploration informed them as to how to grow/adjust their *model(s)*. An example is P3 in which she spent all of her first three independent sessions exploring her data and learning what did and did not exist. After she felt adequately informed about her data, she then defined a set of *models* based on what she learned. Through this she performed *model* growth with respect to the *models* she used during her exploration phase.

5.2.7 Search Strategies Developed

During our use-cases, we observed that our participants employed different search strategies to navigate and explore their datasets. These strategies were focused on navigating the challenging temporal characteristics of our participants' datasets. The search strategies discussed in this section describe how the participants navigated their data. These strategies were used by our participants to help them perform their analysis strategies described earlier. In this section we detail the successful and unsuccessful search strategies developed and used during our use-cases. Some strategies were not known at the beginning, but evolved over time as our participants learned more about the characteristics of their data.

Data Formatting

During our use-cases, several data manipulation methods were useful for viewing and searching the participants' data. Each participant's data exhibited various temporal characteristics that required some data manipulation to facilitate searching it. In this section we discuss these temporal characteristics and the various formatting strategies applied (and why) during our use-cases.

Temporal Characteristics:

We saw varying temporal characteristics within the datasets of our participants. Overall, we saw the datasets exhibited either dense events, sparse events, or event clusters. Examples of each, respectively, are illustrated in Figure 5.4. Dense event data is characterized my many events among one or more channels for a prolonged period of time. Sparse event data is the opposite with very few events among one or more channels. Lastly, clustered event data is characterized by dense groups of events among one or more channels with time gaps between each group. These characteristics were not new, however, the challenge was each participant had datasets consisting of multiple sessions where each exhibited combinations of these characteristics. Their data was also multi-channel where some channels were sparse, others dense, others had clusters, and some channels even exhibited all three characteristics depending on the timeframe of the dataset being viewed. Since the data was multi-channel, every channel had and did exhibit these characteristics while in parallel with other channels with their own characteristics.

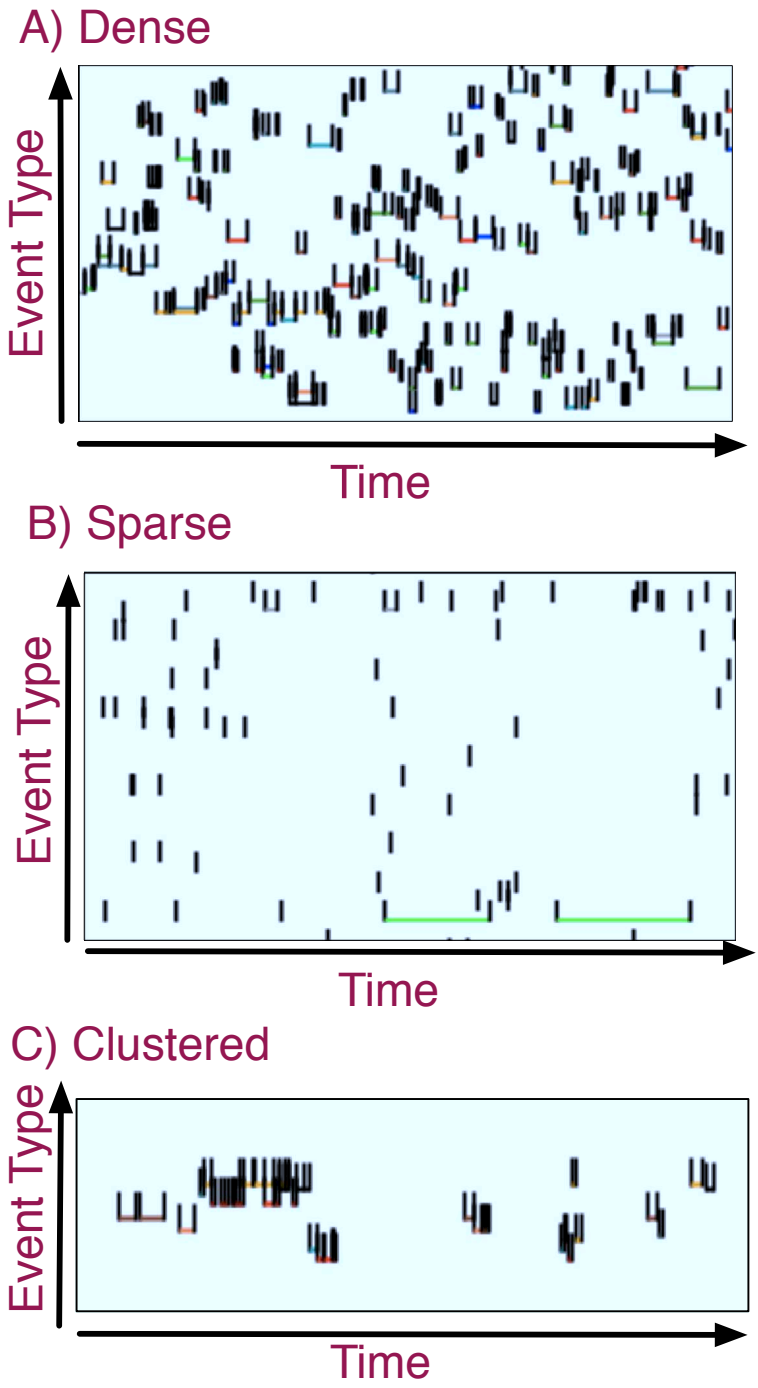


Figure 5.4: Examples of the different temporal characteristics.

Hence, the search strategies that worked for one part of their dataset did not for others. This led to the participants discovering how to address this challenge. Needless to say, these temporal characteristics provided a challenging data-space to search effectively. Next, we will discuss the different formatting strategies applied to aid in searching.

Formatting Strategies:

The idea of formatting data is not new, however, given the varying temporal characteristics of our participant's data, carefully considered formatting was invaluable. The focus of this discussion is how they were applied given the different temporal characteristics of the data.

Normalization: The first formatting strategy is *normalization*. In the context of our use-cases, normalization refers to transforming all event to be the same length, preserving their order, removing any time gaps between events, and produce a simple event sequence that captures the events' order. This is illustrated in Figure 5.5. Here we see an excerpt from P2's data in which the before and after effects of normalization are illustrated. All events were normalized to 1 time unit intervals, allowing an easy view of the ordering trends in the data. This is useful to ignore any temporal variability between event lengths and time gaps between events and focus just on the order (which may only be the relevant focus). This was successfully applied to P2's data as the initial focus of P2's analysis was the order of certain events. After certain event orders (i.e., *models*) were identified, P2 could look at the unaltered datasets to view the relative event lengths to gain more insight into the *models* identified.

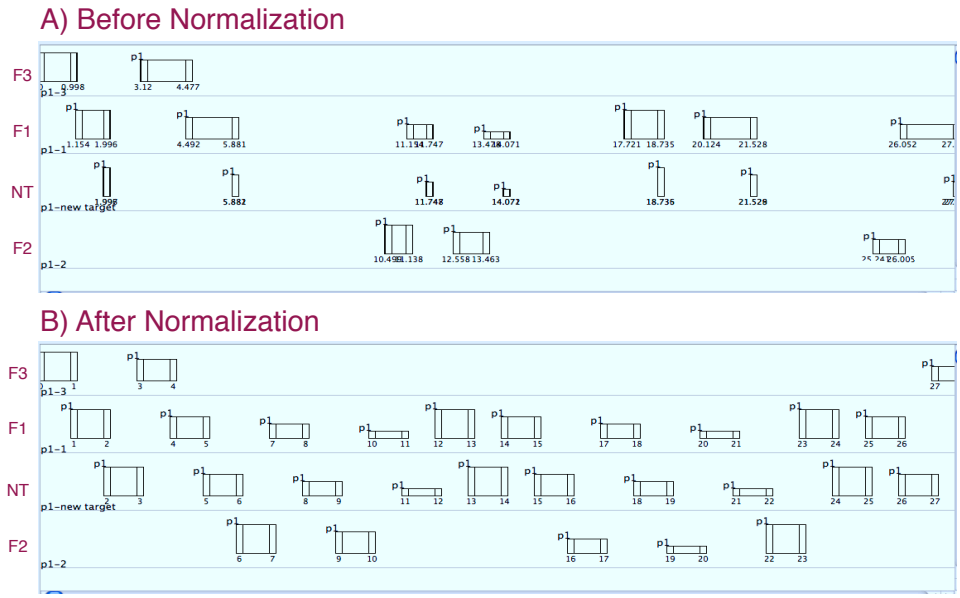


Figure 5.5: An excerpt from P2's data. A) Before normalization, and B) after Normalization. For both, F1, F2, and F3 represents the different finger modes and NT a new target.

Filtering: The second formatting strategy is *filtering* out details. We realize this also is not new, however, filtering out finer details (e.g., transcription speech content and highly reoccurring events) can allow a better view of the structure of the data at a high level, after which, the details can be added back in to better inform specific occurrences in the data. This was applied for P1 and P3 as both of their datasets had information that was distracting from viewing their data in a way that was meaningful to them. For P1, this was filtering out all the details except for the information necessary for identifying turn-taking. P1 was interested in identifying interrupting or out-of-turn behavior among the participants in his data. Hence, P1 filtered out all superfluous details that distracted from this goal. This can be seen in Figure 5.6, where in Figure 5.6A we see the unfiltered data where each of P1's users have multiple channels. User B actually has more channels than is depicted and user C is not even show since there is not enough space. In Figure 5.6B, filtering is applied and each

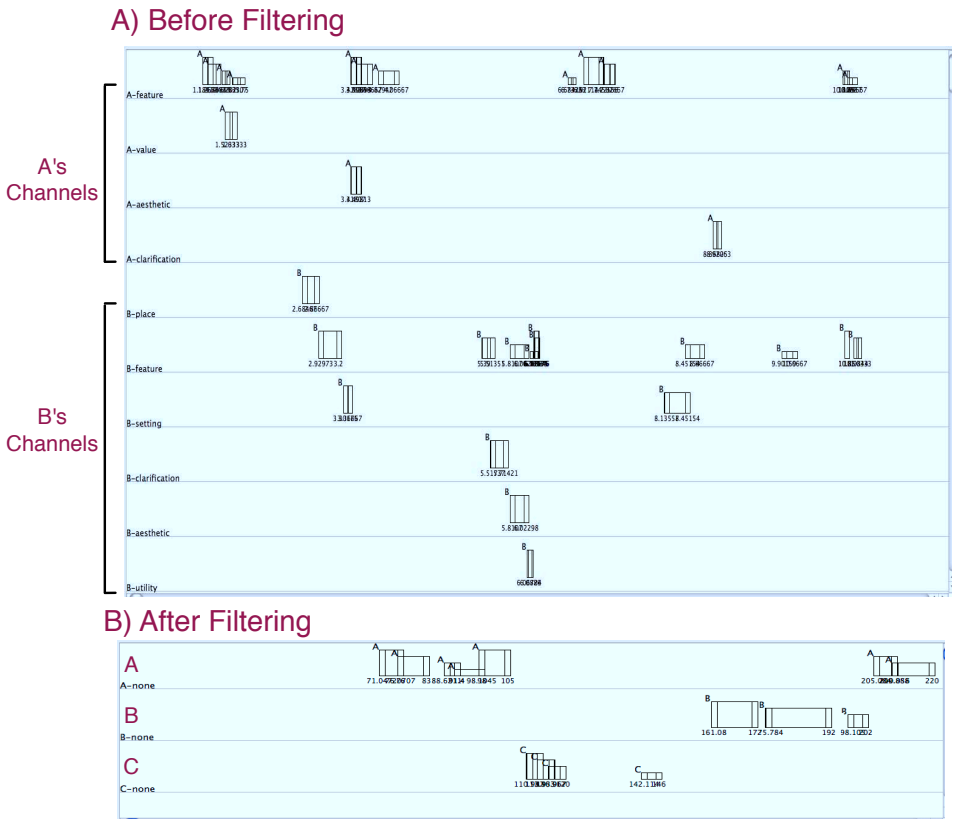


Figure 5.6: An excerpt from P1's data. A) Before filtering. Note that not all channels are shown as there are too many. B) After filtering where each user in P1's data has only one channel (total of three).

user has one channel that represents when they took a turn. Performing filtering allowed P1 to successfully search through the data and identify evidence for his analysis. For P3, this was filtering out events from the interaction mouse logs, which had a high occurrence. The mouse log data was distracting at the beginning of the analysis and P3 chose to filter out these events. This allowed P3 to focus on the aspects of her data that were relevant to her analysis and to identify evidence for her analysis goal. The affect of filtering in terms of the number of channels can be seen in Table 5.2.

Clustering: The third formatting strategy is *clustering*. As discussed earlier, this was an

offline feature where the participant's data could be clustered within channels. This was useful when a channel had a string of events that a participant wanted to be coalesced into one event, or clustered together. This is illustrated in Figure 5.24. P1 was the only participant who used this feature and it was applied after he had *filtered* his data. In his data, each turn of his participants had several events, however, P1 just wanted each turn to be seen as one event. Therefore, he applied clustering for each of his participants to achieve this. Some cases were not captured since the cluster threshold (chosen to be 3 seconds) was too small but P1 was afraid of making the clustering threshold too loose resulting in skewing the turn-taking structure. Applying clustering to P1's data facilitated P1 in identifying occurrences in his data. The reader should be careful not to be confused between *clustering* as a formatting strategy and clustering as a description of the data.

Time Scaling: The last formatting strategy is *time scaling*. As discussed earlier, this feature was originally developed to aid in viewing datasets with long timeframes (e.g,. years). It allows the participant to define a base time unit (the time unit the data is stored in, such as seconds or minutes) and then define a viewing time unit. For example, if the events are stored at the minutes level, one can view the data at an hour time scale, in essence zooming out. P3 was the only participant who used this feature. Her data was stored in seconds, but she found it useful to view the data at the minute timescale. This was especially helpful for her since each of her datasets represented two hour-long, video captured sessions and so viewing in minutes gave her an easy way to index what part of a session she was viewing, e.g., either at the beginning, near the middle or at the end. It also allowed her to zoom to

a better viewing level.

Temporal Constraint Categories

As seen in Figure 5.8, right column, even after applying formatting strategies, each participant's dataset still posed a challenging search space (notably P1 and P3's). Hence, application of different temporal constraints was necessary. As discussed earlier, a temporal constraint is a constraint applied to govern the temporal relationships between events. For example, if searching for event A followed by event B, one may only be interested in matches where B occurs within 5 seconds of A. Such a temporal constraint defines the relationship desired between A and B. This is an example of a temporal constraint class seen in [83, 115, 130]. There are several temporal constraint categories our participants employed and we will discuss them in turn. For ease of reference, the figure from Section 5.2.10 is repeat in Figure 5.7.

Model Constraints:

The following temporal constraint categories are constraints that are applied to matching search criteria (i.e., a *model*) in the data. These aided our participants in expressing their search criteria and identify results that were relevant to their analysis goals. The following temporal constraints were discussed briefly earlier. Here we provide a more in-depth discussion.

Strict Time Windows: This is the most well-known temporal constraint in which the

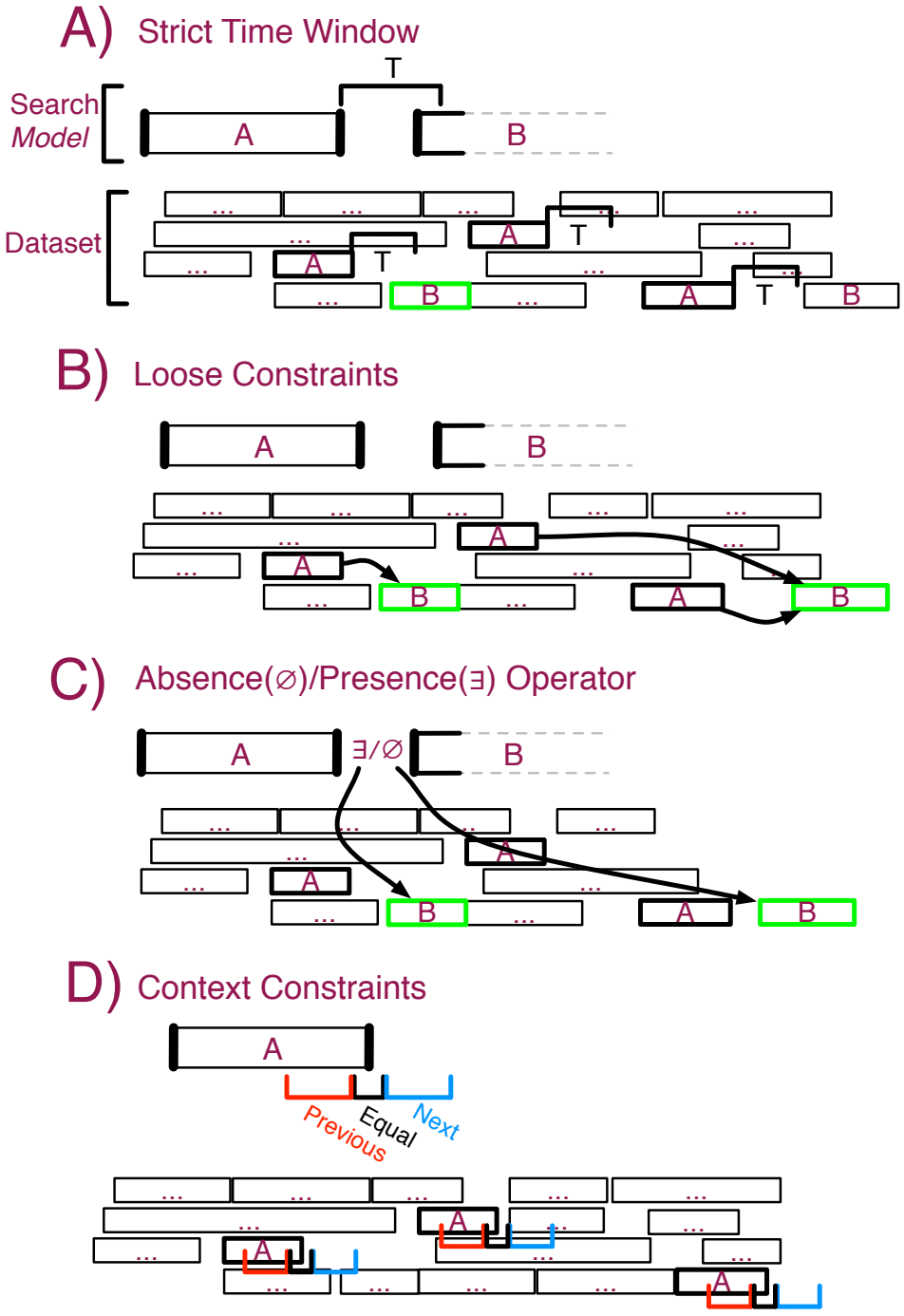


Figure 5.7: A) Example of *strict time window* constraints. The bold, green B event is the match. B) Example of *loose constraints*. C) Example of *absence* and *presence* operator. D) Example of *context constraints*.

relationship between events is defined through a strict time window. The above describe example with event B occurring within 5 seconds of A is an example of this temporal constraint category. An example of this can be seen in Figure 5.7A with $T = 5$. This category has been successfully applied in symbolic temporal mining [115, 130] and *pattern* discovery in behavioral data [83]. In our use-cases, there were two classes of strict time windows applied. One in the classic sense of sequence order, e.g., B follows A within 5 seconds, and the other in terms of equality. The first, which we will call a *next constraint*, allowed our participants to put limits on their search results as their focus may be on temporally tight relationships and is the example in Figure 5.7A. P1 and P3 applied this constraint as they explored their data, figuring out how temporally tight or loose their *models* were within their respective data.

The second class, which we will call an *equal constraint*, allowed our participants to define what it meant for two events to be equal. The events are segmented/extracted from data streams that can have uncertainty and/or variability, hence, testing equality between two events must take this into account. The participants were provided the ability to define an equality window for their data. Due to the nature of either the participants' data or transformations applied to the data, or both, none of the participants required adjusting the *equal constraint* from its default value of 1 frame, i.e., 33 milliseconds.

Loose Constraints: If events have temporal variability, then a strict time window may not be realistic. Hence, applying *loose constraints* where order between events is preserved but no constraint is given for the timeframe in which the events occur may be necessary.

This is illustrated in Figure 5.7B. However, when this is done many search results may be meaningless as they contain overlap. In Figure 5.7B, there are three matches but two of them overlap with each other. What is sought is the match with A temporally closest to B. To address this issue, one must apply the concept of non-overlapping support for the search results. An idea from data mining [115], non-overlapping events are a set of events that do not have any overlap. We adopt this idea and return results that are the temporally most tight. This allowed our participants to have loose constraints and not be presented with irrelevant, overwhelming results. P1 found this very useful as his data varied much from dense, to sparse, to clustered. Hence, he was successfully able to apply *loose constraints* in searching and identifying relevant matches.

Absence/Presence Operator: P1 was very interested in *models* where events of the *model* were not interrupted by any other event. This resulted in an absence (and presence) operator, illustrated in Figure 5.7C. This allowed our participants to specify whether or not other events were present (or not) between specified events of their search *model*. This was heavily utilized by P1 who desired to identify matches where specified events in his *model* had no other events between them.

Context Constraints:

Temporal constraints were also needed in governing the context returned with identified matches. Our participants were not only interested in identifying matches to their searches but also the context in which those matches occurred. They were interested in a situated view of their matches, i.e., viewing the context of each of their matches. Hence, the development

of *Context Constraints* that depicts how much context is desired for each match. These constraints are the suggestion categories we developed in [95, 93] in which events within certain time windows *previous*, *concurrent to*, and *after* each match are also provided. This is illustrated in Figure 5.7D with the different time window categories. Each match of the *model* has a certain amount of its situated context returned based on how the time windows are defined. *Context constraints* were applied to each semi-interval of the participants' *models*, as seen in the figure.

Choosing *Context Constraints* wisely can be challenging if the data is sparse, dense, or contains clusters. One may receive too little or too much context if the constraints are not properly chosen. P3 found such controls useful as her data was sparse and clustered at varying time intervals. Hence, she used it to widen the context window when her data was sparse.

Search Strategies

In this section, we discuss how our participants successfully (and unsuccessfully) applied the above formatting strategies mixed with the temporal constraint categories in order to search for relevant *model* matches in their data. In prior sections, we have discussed how each participant successfully utilized the formatting strategies or temporal constraint categories in isolation. Here we will present how they applied them in combination.

For reference, an overview visualization of the participants' data is show in Figure 5.8. Here

the original of each participant, P1, P2, and P3, are in the left column (top to bottom, respectively). The result of the applied formatting strategies are along the right column, respectively. The x-axis is time and the y-axis is the event types grouped by the sessions within the respective participant's dataset.

P1 had data characterized by dense sections and sparse sections, and everything in between. At the beginning of his analysis, P1 knew he initially was interested in turn-taking in his data. Hence, he applied *filtering* to filter out unnecessary elements of his data so he could view the turn-taking structure. He then found the need to apply *clustering* so each turn taken by the users in his data was seen as a single interval. Each turn consisting of multiple events was causing matching problems with his master *model* (discussed below). Applying clustering facilitated P1 in searching for matches for his analysis. The contents of P1's datasets after applying the formatting strategies can be seen in Table 5.2 and a visual overview is illustrated in Figure 5.8, top-right.

He originally tried using *strict time windows* as a *model constraint*. An example pattern of this can be seen in Figure 5.9A (i). Here P1 has defined three semi-intervals, one after the other within 3 seconds (outside of the 33ms *equal constraint*). He also defined variables (X and Y) for matching the participants in his data (predicate logic). The 's' and 'e' represent a start or end semi-interval, respectively. Due to his data temporal variability, this was not very successful. Hence, he tried *loose constraints*, which provided better results. However, he was still not satisfied. He then turned to applying the *absence operator* in tandem with *loose constraints*, each carefully placed in his *model*. This resulted in a master *model* that

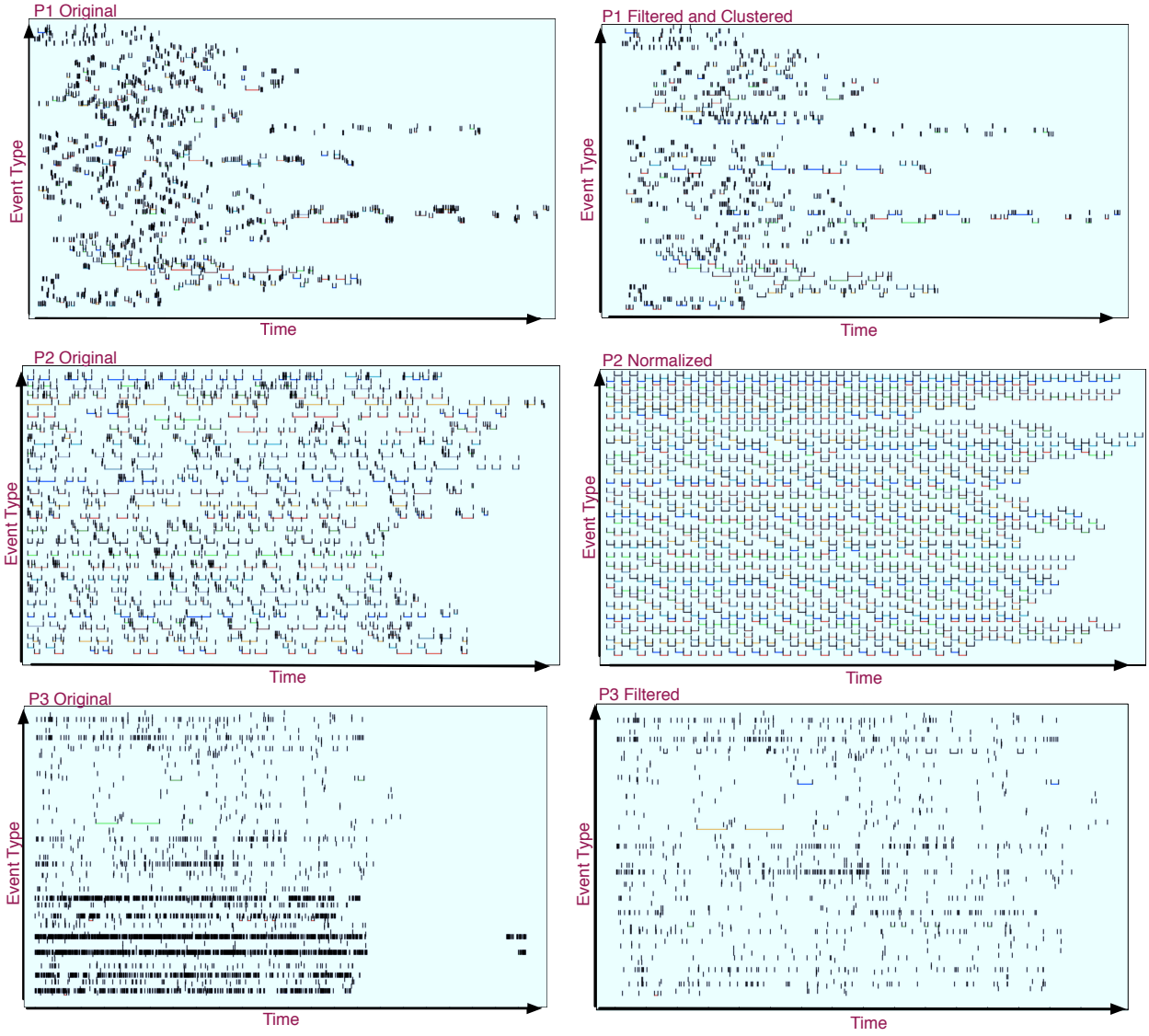


Figure 5.8: Overview visualization of each participant's data displayed as sequences of semi-intervals. The original of each participant, P1, P2, and P3, are the left column, respectively. The result of the applied formatting strategies are the right column, respectively. The x-axis is time and the y-axis are the event types mapped to y-values and grouped by session. Graphs created with TDMiner [144].

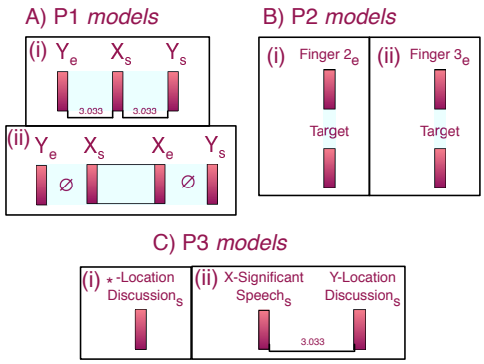


Figure 5.9: Example *models* each participant defined as a search query.

P1 used for a majority of his analysis sessions. This *model* can be seen in Figure 5.9A (ii) - repeated from earlier for the ease of the reader.

P2 had data characterized by wide temporal variability in terms of event length and gaps between them. P2's interest was looking at specific orderings between finger mode usage. Hence, he applied *normalization* to streamline his data so the ordering structure would be more easily apparent. Doing so allowed him to search based on boundaries (*equal constraint*), i.e., transition points between finger mode events. An example *model* of this can be seen in Figure 5.9B. Here P2 has defined two sets of two semi-intervals designed to identify when one of his participants reached a target using either finger mode 2 or 3, respectively. Since *normalization* removed time gaps between finger events, this allowed high accuracy for identification of the boundaries of interest. The high accuracy of an *equal constraint* was noted in our earlier work [93]. The contents of P2's datasets after applying the formatting strategies can be seen in Table 5.2 and a visual overview is illustrated in Figure 5.8, middle-right.

P3 had data characterized by sparse, dense, and clustered events between and within chan-

nels. P3 was concerned about her mouse log data being distracting since it was very dense, hence, she first *filtered* out her mouse log data. She then began exploring her data with simple *model* searches to get a feel for what existed in her data. She experimented with using *next constraints*, *loose constraints*, and *context constraints*. The *loose constraints* did not operate well in her data as most of the matches were temporally too far apart to be meaningful. The reason for this was later realized to be the matches P3 sought actually did not occur very often. She tweaked the *next constraints* according to a timeframe that was meaningful to her, which was successful. After which, she adjusted the *context constraints* to report a context window that was also meaningful to her, which was also successful. Through her exploration and analysis, she discovered her data was characterized further as having little tight pockets of activity. This explains why applying *next constraints* and *context constraints* was successful.

Examples of two of P3's *models* can be seen in Figure 5.9C. The first one is an example of a single semi-interval P3 used to "sniff" through the data by looking around occurrences matches of this *model* using *context constraints*. In this case, it was one of her users (hence the use of '*' as in a regular expression) discussing the location of something on the display. The second *model* is an example of one of the *models* P3 used later in her analysis to identify when her participants had a significant speech event within 3 seconds of the other participant discussing the location of something on the display.

During her analysis, P3 also made use of *time scaling*. The base time unit of her data was seconds, however, she was more familiar with looking at the data in terms of minutes.

This was especially meaningful to her as the data she was searching were 2-hour long, video recorded user studies. Hence, indexing based on minutes was more meaningful to her since she was able to quickly identify which part of the study she was looking at instead of needing to mentally map seconds to minutes. The contents of P3's datasets after applying the formatting strategies discussed can be seen in Table 5.2 and a visual overview is illustrated in Figure 5.8, bottom-right.

5.2.8 Strategies Summary

The analysis strategies developed by our participants follow a common thread: hypothesis testing. Each would specify initial *model(s)* based on their initial hypothesis, use that to learn about their data, then re-specify the *model(s)* based on knowledge gained. Two variations were observed. The first can be described by the following:

1. Formulate an initial hypothesis to investigate in the data.
2. Specify an initial *model* based on the initial hypothesis.
3. Explore the data based on the initial *model*.
4. Based on knowledge gained through exploration, re-specify initial *model* and continue exploring.
5. During exploration, the participant may stumble upon an unrelated hypothesis and then focus in that direction. Such new hypothesis (or new set of hypotheses) may trump the previous.

6. Potentially, multiple *models* are specified and iterated over for each hypothesis.

7. Re-specification and exploration of all *models* are performed until a representative set of *models* are molded.

8. Iterate through the representative set of *models* and count the number of times they exist in the data.

This first variation was used by P2 and P3 to discover relevant *models*, and then count the existence of such *models* to support their hypotheses. We will discuss how P2 and P3 fit into this variation in turn.

P2's original hypothesis was focused on finding finger mode *models* that explained good performance. During his exploration he stumbled upon a new hypothesis that became his new analysis goal. This new hypothesis was looking for a correlation between finger mode duration and performance of participants. Hence, he focused his analysis on identifying patterns that supported this hypothesis. P2 already knew performance measures from previous analysis and he was trying to find support for this potential correlation. At the end of his analysis, he focused on a few molded *models* which he counted their existence amongst his various participants. Unfortunately, his new hypothesis was not supported by what he found in his data, but P2 did learn some valuable information. He found out that in order to find any correlation he needed more users, plus he learned that his data had a lot of variety and inconsistency since he observed that people do random things (the human element). When asked in the follow-up questions (Figure C.7) whether the above analysis strategy represents

how he conducted his analysis, P2 stated that it was vaguely similar. He pointed out that during the verification of his initial hypothesis, he came up with a completely unrelated hypothesis and began investigating the new one. Hence, the portion described in this analysis strategy about potentially discovering a new hypothesis came from P2's feedback during his follow-up. Two excerpts from P2's final independent session interview nicely wrap-up his impression of the approach and how it helped him:

> "So all these sessions were pretty much part of one big exploration which was getting numerical values of my interest [based on defined *models* that were meaningful to him] and once all those numerical values were gathered, I tested my hypothesis and completed my hypotheses testing..."

> "...this process helped me in two things. First, kind of helped me to look at these values, these numbers in a different way in the sense that this tool broadened my horizons in the sense that I could look at more things that I couldn't normally see. Second good thing about this tool is that it allowed me to gather information that I needed for my hypothesis testing"

P3's hypothesis was that the display used by her participant groups would serve as a medium for common ground. Initially P3 focused on learning about what was coded in her data, specifically events related to interactions with the display. Once she learned about the contents of her datasets, she focused on a specific set of *models* representing different interactions with the display. The *models* P3 identified and counted at the end of her analysis provided

behavior models that explained why different groups performed better than others. Plus she found "... there were plenty of instances of *patterns* representing a shared understanding of the spatial layout of the display, and they were able to use that for sense making." Hence, she found evidence for her hypothesis, however, she comments on how more analysis is needed. When asked in the follow-up questions (Figure C.7) whether the above analysis strategy represents how she conducted her analysis, P3 stated she agreed with it. An excerpt from P3's final independent session interview nicely wrap-up her impression of the approach and how it helped her:

> "...a lot of it was learning and exploring, not only about the tool's functionality but also my data. ... my *patterns* got a lot more specific as the time went on and I was able to direct ... my approach. It also, this tool helped me find out what actually was going on in the data instead of me just guessing *patterns* in the beginning. So I'd say I'm a lot more comfortable with it now and I'm actually trusting the results that it gives me."

The second variation can be described by the following:

1. Formulate an initial hypothesis to investigate in the data.
2. Specify an initial *model* based on the initial hypothesis.
3. Perform the following in a non-specified order including repetition of step(s):
   (a) Look at identified matches and investigate relevance (i.e., whether the matches are related to the analysis goal, mis-codings, etc...).

   (b) Based on knowledge gained through exploration, re-specify initial *model* and start exploring again.

   (c) When a relevant match is found, record the match with a descriptive label.

4. The result of the above will be a "master *model*" where focus is on exploration and investigation of matches to this *model*.

5. The "master *model*" may need to be updated during exploration/identification in which repetition of some (or all) of the above steps will be conducted.

This second variation was used by P1 to discover how interruption/out-of-turn behavior existed in his data. This variation was driven by P1 identifying how behavior of interest exists in his data, which is similar to the first variation. However, in this variation, the focus is on identifying the formulation and existence of one specific type of *behavior model* instead of a set. His hypothesis was that evidence of collaboration among the participants could be found through identifying *models* of contributions that did not follow the simple turn-taking of the group. Hence, P1 focused on molding a *model* to identify such interrupting/out-of-turn occurrences. As P1 learned more about his data, he would adjust his focal *model* until a "master *model*" was achieved. Even this *model* was adjusted when needed. Such updating and molding were needed as P1 found that the *behavior model* sought was not as simplistic as originally thought. This lead to looking at more complicated *models*. Through this approach P1 was able to find the evidence he desired for this kind of collaboration, however, he did note that more analysis is needed. Even at the end of his analysis, P1 noted that an even

more complicated *model* may be needed to focus in on the occurrences he is interested in. When asked in the follow-up questions (Figure C.7) whether the above analysis strategy represents how he conducted his analysis, P1 stated that "it does represent and describe how I conducted my analysis". Two excerpts from P1's final independent session interview nicely wrap-up his impression of the approach and how it helped him:

> " Anytime you take another look at your data you are going to learn something more so it's a nice opportunity to do that. A lot of my analysis of this data so far has been mostly quantitative and this [our approach] is relatively quantitative as well but I am able to translate some of my qualitative interests into these *patterns* we are searching for and that's pretty cool "

> "It has been helpful in finding evidence of collaboration for sure ... that was my goal."

The analysis strategies described above differ from the intended use of the approach. The intended use was based on how TEBA experts from our motivating domain, multimodal behavior analysis, conducted their analysis [122, 124, 89, 88, 24]. The general difference can be seen in Figure 5.10. Here, the difference is seen in how a *model* is updated. A *model* is intended to be updated directly from suggestions found in the context of *model* occurrences. However, the actual use was to respecify the *model* representing the initial hypothesis based on exploring the data and learning about its contents. There were some cases (mainly with P3) where the provided context (i.e., context presented through *context*

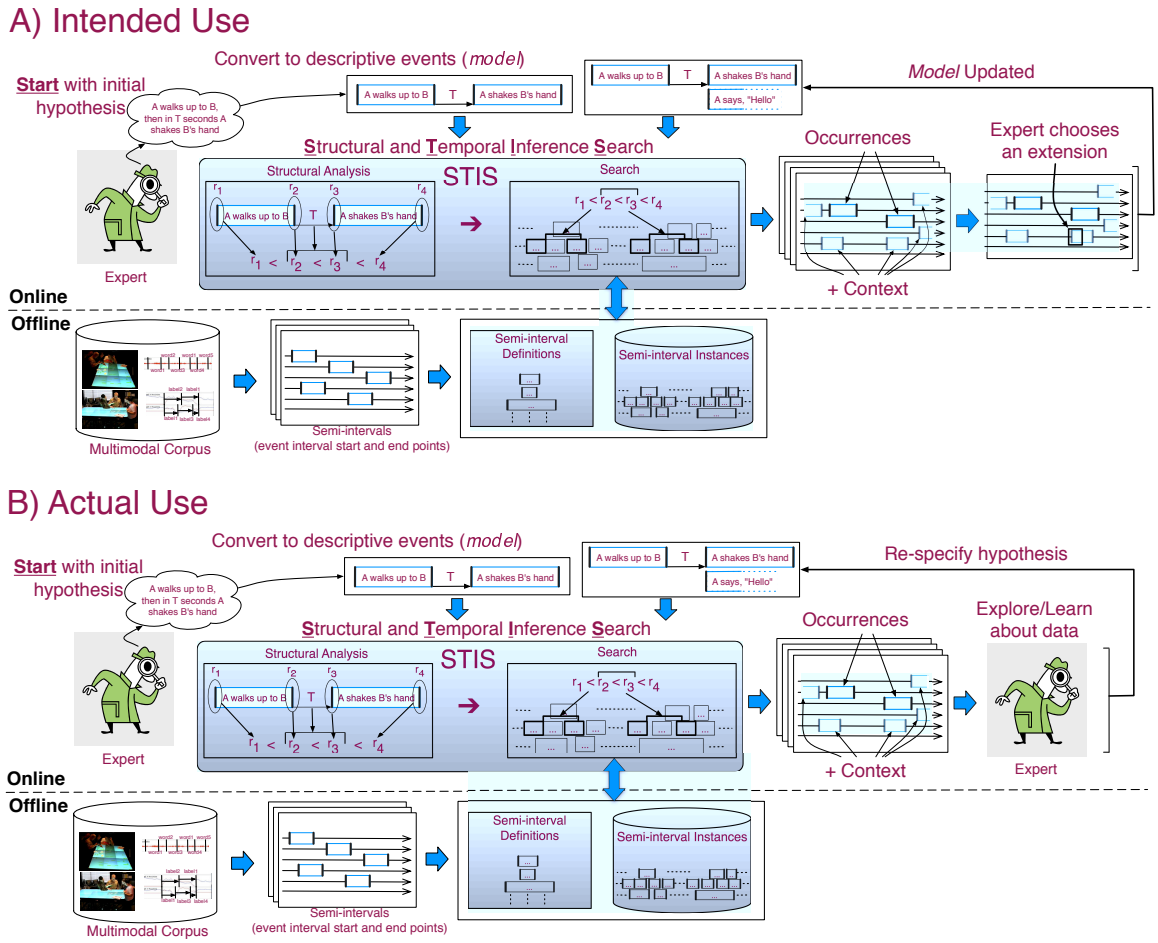


Figure 5.10: Intended use of our approach (A) and the actual use (B). The difference is found in how the *model* is updated: A) Direct updated from suggestions found in context of occurrences, or B) Re-specification of initial hypothesis (*model*) directly by the expert based on their growing knowledge of the data.

constraints) provided the source for spurring on exploration and *model* growth. However, the view into the data that was provided (mainly through the Situated View and Event Sequence Overview described in Section 5.2.10) was the major factor in informing *model* updating (evolution).

5.2.9 Aid in Discovery

Here we begin our discussion of software use by our participants and the functionality utilization. We start with discussing how our approach aided our participants in discovery. During each semi-structured interview, we asked them if our approach aided them in discovery (Question 4 - see Appendix C). The results can be seen in Figure 5.11 and provide support that our approach did aid them in discovery.

Separate from the Likert results, we observed for each participant how our approach aided them in discovery. P1 was able to discover the various variations of interruption/out-of-turn behavior that can occur in his data. The exploration supported by our approach allowed him to identify how the behavior he was interested in existed in his data. P2 was able to discovery the *models* of finger mode usage that were exhibited by users with different rankings. He was able to do this through initial exploration of his data then narrowing down to search for *models* of certain structure and meaning. P3 was able to discover the relevant *behavior models* in her data through similar means to P1 and P2. She performed some initial exploration and then discovered what *behavior models* existed in her data. After

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P1	5	5	5	5	5	5	5	5
P2	4	5	4.5	3	3			3
P3	5	4	4.5	5	5	5	5	5
		TS Total Mean	4.67				IS Total Mean	4.33
							Total Mean	4.5

Figure 5.11: 5-point Likert scale results for Q4 of participants' semi-structured interviews: Did the *model* growth/exploration strategy help the process of discovery?

which, she narrowed down her search just like P2 to *models* with certain structure and meaning. Overall, our approach was able to give each participant a view into their data that informed them of its contents and allowed them to guide the analysis process and discover how certain behavior actually existed in their data.

5.2.10 Feature Use

In this section we will discuss how each of the features of our system were used, whether for its designed purpose or not. The features we will discuss here are Aggregate View, Situated View, Event Sequence Overview, Temporal Constraints, Descriptive *Models*, Predicate *Models*, Abstract Segmentation, Video View, Time Scaling, and Clustering. A more detailed description on each feature can be seen in Chapter 6. Each feature is accompanied with the 5-point Likert rankings from the semi-structured interviews. The "N/A"'s were sessions where the respective participants did not use the respective feature.

Aggregate View: This was purposed to view the results from all of a *model's* instances

A) Capability is useful

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P3	N/A	N/A	N/A	5	5	5	N/A	5
		TS Total Mean	N/A				IS Total Mean	5
							Total Mean	5

B) Improves ability to do research

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P3	N/A	N/A	N/A	5	5	5	N/A	5
		TS Total Mean	N/A				IS Total Mean	5
							Total Mean	5

Figure 5.12: 5-point Likert scale results for Aggregate View.

in one view, hence, aggregate view. This feature was only used by P3 and in a way not anticipated. P3 used it for its intended purpose to see all the results in one place in order to see what was in the data but also used the results to count the number of situated instances of what did occur. P3 did this through the Abstract Segmentation view of the suggestions. She would look through the suggests and count how many occurred based on the Instance (Time) boxes provided in Abstract Segmentation, discussed below and illustrated in Figure 5.20. We never thought about this functionality but viewing the Instance boxes in Aggregate View can give the user an idea of the number of situated instance of a particular growth direction. In other words, the user can see how many instance exist with a given *model* extension from the suggestions without actually adding it to the *model*. P3's rankings of the Aggregate View can be seen in Figure 5.12.

Situated View: This view was purposed to view the results and suggestions for instances

A) Capability is useful

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS	IS Mean
P1	N/A	5	5	5	5	5	5	5
P2	5	5	5	5	5			5
P3	5	4	4.5	5	N/A	5	5	5
		TS Total Mean	4.83				IS Total Mean	5
							Total Mean	4.92

B) Improves ability to do research

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P1	N/A	5	5	5	5	5	5	5
P2	4	4	4	4	4			4
P3	5	5	5	5	N/A	5	5	5
		TS Total Mean	4.67				IS Total Mean	4.67
							Total Mean	4.67

Figure 5.13: 5-point Likert scale results for Situated View.

one at a time. This also employed Abstract Segmentation to view the suggestions, however, none of the participants used this feature in Situated View. The only use was in tandem to the Event Sequence Overview (ESO, discussed below) where each situated instance in turn could be graphically displayed in context. Hence, the participants used this link to the ESO to view the context instead of using Abstract Segmentation. This makes sense as the instances being displayed in the ESO gave an easy visualization of the contexts the instances were found.

P2 and P3 used the Situated View to obtain the instance count of different *models*. P1 and P3 focused on looking at the situated instances in context with the video they were extracted from to verify whether they were relevant to their analysis. The participants' rankings of

A) Capability is useful

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P1	5	5	5	5	5	5	5	5
P2	4	3	3.5	3	3			3
P3	5	5	5	5	4	5	N/A	4.5
		TS Total Mean	4.5				IS Total Mean	4.17
							Total Mean	4.33

B) Improves ability to do research

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P1	5	5	5	5	5	5	5	5
P2	4	3	3.5	3	3			3
P3	5	5	5	5	5	5	N/A	5
		TS Total Mean	4.5				IS Total Mean	4.33
							Total Mean	4.42

Figure 5.14: 5-point Likert scale results for the ESO.

the Situated View can be seen in Figure 5.13.

Event Sequence Overview: The purpose of the Event Sequence Overview (ESO) was to view the participants' datasets in a "music score" style visualization like many other multimodal analysis tools. All three participants really liked the ESO and it helped them browse their data (e.g., visual scan) and view the context in a simple yet powerful graphical representation. P1 and P3 used it to scan for potential *model* occurrences when their queries did not turn up any results. In at least one case, P1 did find a match our system did not find but this was due to bad clustering of the data (see the Cluster section below). The participants' rankings of the ESO can be seen in Figure 5.14.

Temporal Constraints: This feature started as a way to constrain how temporally close

(or far apart) two semi-intervals should be. For example, if one is looking for event A followed by event B, the interest in such a match may only be when B follows A within 5 seconds. This we call a *model constraint* and is illustrated in Figure 5.15A. However, the idea of Temporal Constraints branched out to several forms of constraints. Here we will briefly describe the constraints. Further details are discussed in Section 5.2.7. Aside from the previously mentioned temporal constraint, the first was controlling the time window that events are deemed suggestions around matching instances. For example, one may only be interested in extensions to a *model* if such extensions (suggestions) are within 1 second of instances of the *model*. This we call a *context constraint* (or a *suggestion constraint*) and is illustrated in Figure 5.15B.

P1 wanted to relax the *next constraint* between two semi-intervals of a *model* and not care how far (or close) they are (this is a *loose constraint* discussed in Section 5.2.7). This resulted in too many results and hence we applied non-overlapping support for matches (more details in Section 6.3.5). P1 also requested the ability to specify whether any semi-intervals exist or not between specified semi-intervals of a *model*. This resulted in the development of an *absence* and *presence* operator (also discussed in Section 6.3.5 and illustrated in Figure 5.15C).

Overall, P1 and P3 used temporal constraints. P2 performed some transformations to his data to aid viewing his data and hence temporal constraints were unnecessary (discussed in Section 5.2.7). P1 used a mixture of *next model constraints*, *absence operator*, and disabling the *next* constraint (*loose constraint*). Between these abilities he was able to sculpt a master

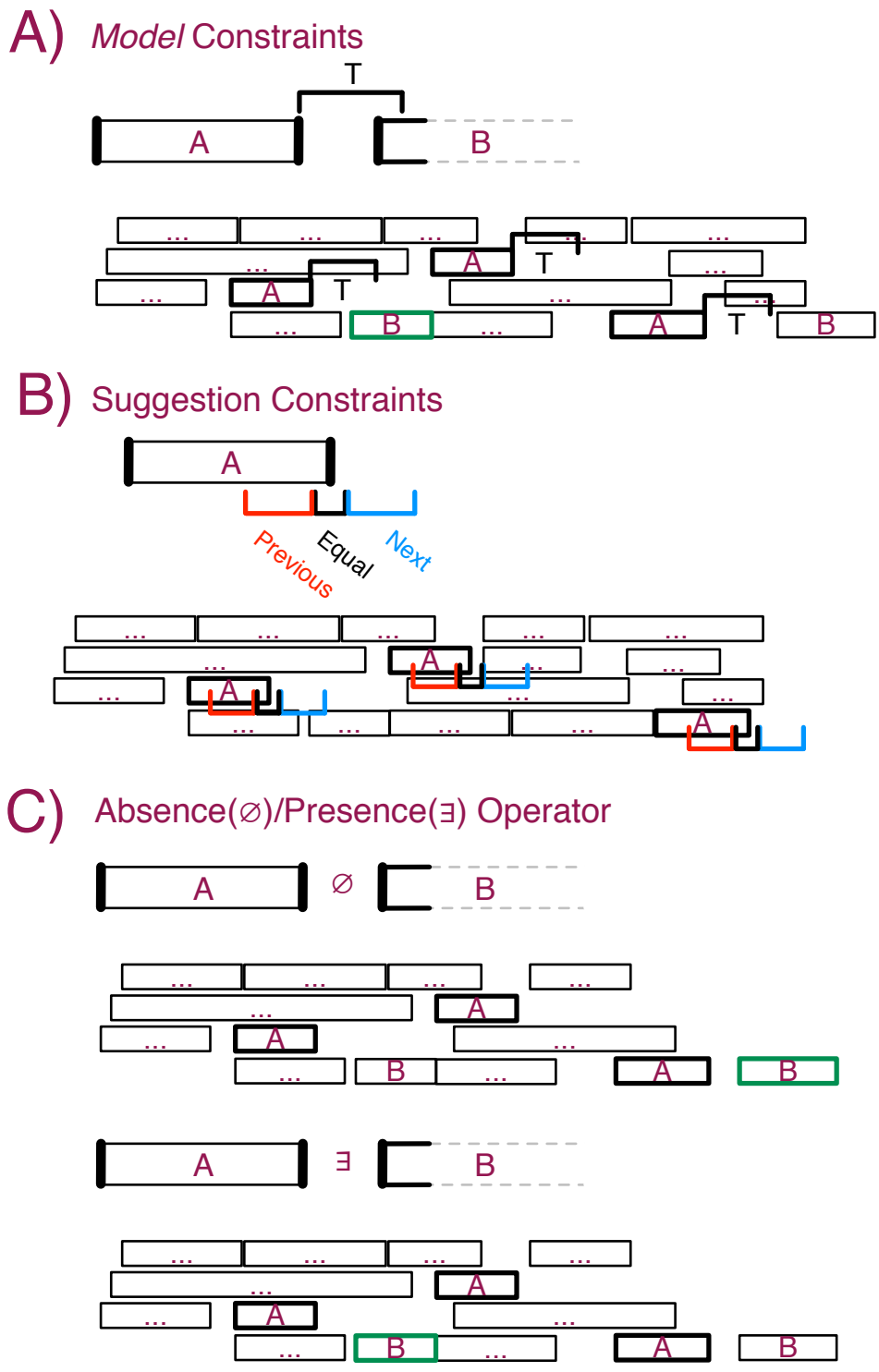


Figure 5.15: A) Example of a *Model* Constraint. The instance with the proper match according to the constraint is in green. B) Illustration of Suggestion Constraints where the events that are within certain temporal windows for each matching instance are presented as suggestions. C) Example use of Absence and Presence operator. The Absence operator identifies matches with no other events between the specified semi-intervals and the Presence operator does the opposite. The respective proper matches are in green.

A) Capability is useful

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P1	5	5	5	5	5	5	5	5
P3	4	5	4.5	4	5	5	5	4
		TS Total Mean	4.75				IS Total Mean	4.88
							Total Mean	4.81

B) Improves ability to do research

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	I	I	I	I	IS Mean
P1	5	5	5	5	5	5	5	5
P3	4	4	4	4	5	5	5	4.75
		TS Total Mean	4.5				IS Total Mean	4.88
							Total Mean	4.69

Figure 5.16: 5-point Likert scale results for the Temporal Constraints.

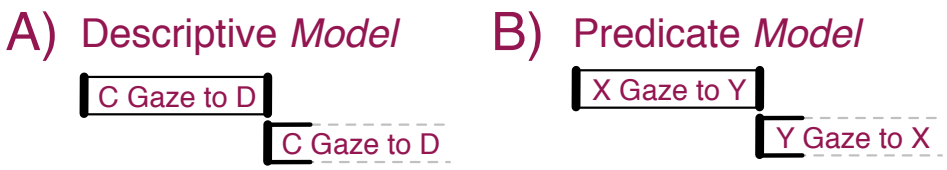


Figure 5.17: A) Example of a descriptive *model* with specific actors C and D. B) Example of the same *model* as a predicate *model* with variables X and Y that could have several bindings including C and D.

model that reflected his interests (Figure 5.3). However, he still wanted some more detailed control over these constraints, such as absence/presence operators for specific kinds of events. This need for P1 gave us ideas for future temporal constraints. P3 tried several forms of temporal constraints until she settled on using the *model constraints* and *context constraints*. The participants' rankings of Temporal Constraints can be seen in Figure 5.16.

Descriptive *Models*: This feature was purposed for describing a *model* using specific

A) Capability is useful

	Training Sessions			Independent Sessions				
Participant	TS	TS	TS Mean	I	I	I	IS 4	IS Mean
P2	5	5	5	5	5			5
		TS Total Mean	5				IS Total Mean	5
							Total Mean	5

B) Improves ability to do research

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P2	4	4	4	4	4			4
		TS Total Mean	4				IS Total Mean	4
							Total Mean	4

Figure 5.18: 5-point Likert scale results for Descriptive *Models*.

values, e.g., Mary looks at Bob. This allowed the participants to describe a *model* with specific values of interest as illustrated in Figure 5.17 A. Here actors C and D are specified with specific event type *gaze*. Only P2 used Descriptive *Models*. He did this since he was interested in looking for specific *models* containing specific values. P2's rankings of Descriptive *Models* can be seen in Figure 5.18.

Predicate *Models*: This feature was purposed for describing a *model* using variables, e.g., X looks at Y, and wildcards, e.g., '*' from regular expressions, along with specific values. This is illustrated in Figure 5.17 B where variables X and Y are given for the involved actors and the specific event type *gaze*. This allowed the participants to describe a *model* with variables and wildcards (i.e., abstract *model*) and then see what specific values bind to the variables and wildcards. For reference, more details and figures can be seen in Section 6.3.2. P1 and P3 used this feature and found it invaluable. Each had different *models* they

A) Capability is useful

	Training Sessions			Independent Sessions				
Participant	TS 2	T	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P1	5	5	5	5	5	5	5	5
P3	5	5	5	5	5	5	N/A	5
		TS Total Mean	5				IS Total Mean	5
							Total Mean	5

B) Improves ability to do research

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS	IS 2	IS 3	IS 4	IS Mean
P1	5	5	5	5	5	5	5	5
P3	4	5	4.5	5	5	5	5	5
		TS Total Mean	4.75				IS Total Mean	5
							Total Mean	4.88

Figure 5.19: 5-point Likert scale results for the Predicate *Models*.

were interesting in identifying in their data but they were unsure what specific values would match. Hence, using Predicate *Models* allowed them to define *models* in predicate and see what matches occur in their data. The participants' rankings of Predicate *Models* can be seen in Figure 5.19.

Abstract Segmentation: This feature was purposed to provide an organization to aid the participants in viewing suggestions. This is discussed in Section 4.2.7 and is repeated for ease in Figure 5.20. The main difference in Figure 5.20 is the added Time boxes (unique instances). These represent each instance the selected Actor-Description combination occurred. Also, for our purposes the *Actor X* is a tuple of actor and event type, i.e., the *Actor X*'s are all the permutations of actor and event type combinations such as for actor *Bob* and event types *Gaze* and *Speech*, there will be *Bob-Gaze* and *Bob-Speech*. P3 was the only

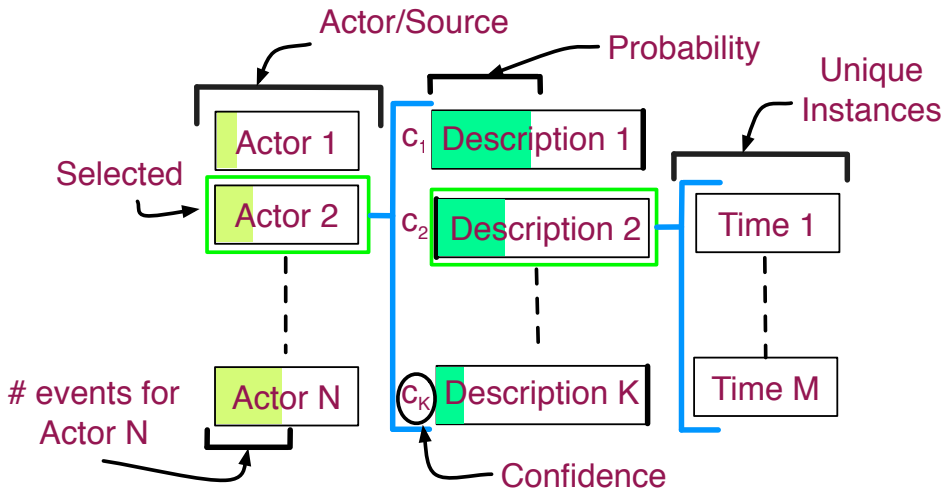


Figure 5.20: Graphical organizational structure for suggestions.

participant who used this feature. As mentioned above in the Aggregate View discussion, P3 found this useful to get a feel for the results of her *models* and then count the number of times different patterns occur. At one point P3 was interested in particular *models* consisting of two semi-intervals but was unsure how they existed in the data. Hence, she defined a *model* consisting of a single predicate semi-interval and then look at the suggestions using the Abstract Segmentation in Aggregate View. Through observing the suggestions, she would fill in the second semi-interval that was of interest according to the suggestions provide. This aided her in defining the *models* she used later in her analysis.

P3 also made several positive comments about the information scent of Abstract Segmentation. Information scent, originally presented in [116], is the idea of providing visual/graphical elements in a visualization to inform the user of its contents. The P3's rankings of Abstract Segmentation can be seen in Figure 5.21.

A) Capability is useful

	Training Sessions			Independent Sessions				
Participant	TS 2	T	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P3	N/A	N/A	N/A	5	5	5	N/A	5
		TS Total Mean	N/A				IS Total Mean	5
							Total Mean	5

B) Improves ability to do research

	Training Sessions			Independent Sessions				
Participant	TS 2	T	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P3	N/A	N/A	N/A	5	5	5	N/A	5
		TS Total Mean	N/A				IS Total Mean	5
							Total Mean	5

Figure 5.21: 5-point Likert scale results for Abstract Segmentation.

Video View: This view was purposed to allow the participants to look at the original data from which their annotated datasets originated. For reference, details and figures can be seen in Section 6.3.4. This served as a means for the participants to verify the relevance/legitimacy of identified instances. P1 and P3 were the only participants with video, however, P2 really liked the feature and saw the value in it and wished he could have used it as well. P1 used this feature extensively in order to verify if the instances matching his *model* were relevant to his analysis. Sometimes this lead to him finding miss-codings or coding errors in his data. P3 used it similarly. She would check the video to get the context of identified *model* occurrences, check occurrences for their relevance, and check for miss-codings (or lack of annotations). P3 discovered that her data was missing some annotations either due to coder error or the users in the video talking too low to be heard leading to nothing being coded. The participants' rankings of the Video View can be seen in Figure 5.22.

A) Capability is useful

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P1	4	5	5	5	5	5	5	5
P3	5	5	5	N/A	5	3	N/A	4
		TS Total Mean	5				IS Total Mean	4.5
							Total Mean	4.75

B) Improves ability to do research

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P1	2	5	3.5	5	5	5	5	5
P3	5	5	5	N/A	5	3	N/A	4
		TS Total Mean	4.25				IS Total Mean	4.5
							Total Mean	4.38

Figure 5.22: 5-point Likert scale results for the Video View.

Time Scaling: This feature was developed to aid in viewing datasets in the ESO with long timeframes (e.g,. years). For reference, details and figures can be seen in Section 6.3.3. It allows the user to define a base time unit (the time unit the data is stored in such as seconds or minutes) and then define a viewing time unit. For example, if the events are stored at the minutes level, one can view the data at an hour time scale, in essence zooming out. P3 was the only participant who used this feature. Her data was stored in seconds, but she found it useful to view the data at the minute timescale. This was especially helpful for her since each of her datasets were two hour sessions and so viewing in minutes gave her an easy way to see what part of her data instances occurred. It also allowed her to zoom to a better viewing level in the ESO. P3's rankings of Time Scaling can be seen in Figure 5.23.

Clustering: This was an offline feature where the participant's data could be clustered

A) Capability is useful

	Training Sessions			Independent Sessions				
Participant	TS 2	TS 3	TS Mean	IS 1	IS 2	IS 3	IS 4	IS Mean
P3	5	5	5	5	5	5	N/A	5
		TS Total Mean	5				IS Total Mean	5
							Total Mean	5

B) Improves ability to do research

	Training Sessions			Independent Sessions				
Participant	TS 2	T	TS Mean	IS 1	IS 2	IS 3	IS	IS Mean
P3	N/A	5	5	4	5	5	N/A	4.66667
		TS Total Mean	5				IS Total Mean	4.66667
							Total Mean	4.83333

Figure 5.23: 5-point Likert scale results for Time Scaling.

within channels. This was useful when a channel of the participants data had a string of event that he or she wanted to be coalesced into one event, or clustered together. This is illustrated in Figure 5.24. A problem we ran into for P1 was the threshold chosen for P1's data worked in most cases. Hence, the missed *model* discussed above that P1 found through visual scan of the ESO. P1 was the only participant who used this feature. In his data, each turn of his participants had several events, however, P1 just wanted each turn to be seen as one event. Therefore, we applied clustering for each of his participants to achieve this. Some cases were not captured as P1 was afraid of making the clustering threshold too loose resulting in skewing the turn-taking structure. Further discussion is below in Section 5.2.7. P1 only actively used Clustering during one session (session TS3, the session where usage was tested and successful), hence, his 5-point Likert ranking only consists of that session in which he gave it a 5. P1 was very pleased with the results Clustering had in aiding his

Figure 5.24: A) Section of P1's data before clustering, and B) the same section after clustering.

analysis. After TS3, P1 only used the Clustered form of his data.

5.2.11 Problems, Challenges, and Criticisms

During the course of our use-cases, there were a number of problems/challenges P1 and P3 encountered. P1 encountered two different problems/challenges. The *first* is related to his data. He discovered some miss-coded data during his sessions which lead to false positives or sections of data that required closer review to identify relevant instances. His coding scheme reflected when participants contributed but not necessarily when they took turns. He found that some turns taken by participants did not contribute to the story, hence, nothing was coded for those participants during such turns. Therefore, his original assumption about contributions reflecting turn-taking order was not completely accurate all the time, however,

most times, it was. P1 also discovered that the scheme used to convert his coded transcripts into event intervals was not completely accurate. For each of the coded features in his transcripts, the start time of the section of text the features came from was known along with the start time of the next text section with the next feature. Hence, some interpolation of the start and end times of the feature segment within a text section was performed based on the number of characters in the feature segment with respect to the text section and the time length of the text section. This provided a good estimated time interval for feature events and preserved relative temporal ordering. However, this caused problems with some matches and produces irrelevant matches. The *second* problem/challenge is related to a desired feature from the system. Since P1 was looking at turn-taking orders (and disruptions of such order), he wanted the system to automatically detect the normal turn-taking order of a session and report any anomalies. Since this was not a current feature of the system, he was faced with how to accomplish this with the features given.

P3 encountered three different problems/challenges. *First*, P3 discovered that her data was coded with less detail than she would have preferred. This caused her to watch more session videos so as to identify instances and explore the data more to discover why her data seemed more sparse than expected. The *second* was due to a connection error experienced with the network drive that stored her session videos. This kept her from verifying the relevance of some identified instances. The *last* was meaningless results presented when she disabled the next constraint. Many of the results were temporally too far apart to be meaningful to her. However, this did demonstrate that the *models* she was interested in existed as temporally

tight occurrences in her data.

Overall, most of the problems and challenges encountered by my participants stemmed from errors in their data. We also saw how our approach and view of their data aided them in identifying miss-codings and when their data contained less detailed annotations than expected. Despite these challenges and problems, our participants were still able to analyze their and identify support for their respective hypotheses.

During the use-cases, our participants were very positive about the system and approach. However, we should point out the possibility that the participants did not want to offend us during the sessions since the moderator was present. This is known as a *demand characteristic* [108]. Despite this concern and potential evidence for such, we did observe that the participants were not hesitant to share criticisms. They may have been lenient in some ways but still shared constructive critiques. In Figure 5.25 are a number of excerpts from the interview transcripts in which we can see some of these critiques. Sections are embolded for emphasis and we provide an explanation and context of the quote. In some of the explanations we provide a contextual label for the quote (in parenthesis at the end). Here we can see that our participants still provided constructive criticism which was insightful for us for providing future improvements. Despite the criticisms, our participants provided many praises and positive comments on our system and approach. These excerpts can be seen in Figure 5.26. This contrast provides a nice rounded view of our participants feedback of the system and approach. Overall, our participants did share more positive comments and praises than critiques.

Critiques

Participant	Quote	Explanation and Context
P1	"What **could be improved**, [pauses and thinks] I don't know, maybe it might be nice to have the ESO, kind of, whenever **I pick one of the instances automatically zoom in or out to make sure the whole thing [instance] is visible in the window**. And then maybe, **from the ESO being able to say play this instance and have it automatically populate the start and end range with some threshold on either side or something**."	P1 talking about user interface (UI) improvements that would be beneficial to the process (feature request).
P1	" **I don't really care about the temporal constraint.** That's an artifact of the current implementation "	P1 was not finding the temporal constraints in its current form useful (during TS2). However, further improvements implemented for later sessions proved useful for P1. (negative comment)
P1	**"Sometimes there is some false positives, something does match our query,..."**	P1 during IS4 commenting on what was not helpful.
P1	**"...it seems like when the matches are very far apart its hard to get the pattern to find them. So that could be improved."**	P1 talking about a potential improvement during IS2. He was having trouble molding his pattern to find instances where the semi-intervals were temporally far apart.
P2	**"...just visualize simultaneously multiple views**, like timelines of different files. perhaps like **do operations on many many files instead of one at a time**, its one thing. **Maybe search and do stuff on multiple files at the same time.** I am pretty sure it's going to be useful for others as well."	P2 explaining the desire to be able to open and search multiple files. This was the main point of criticism for him during his sessions. Whenever asked about improvements or what is not helpful, most of the time he would mention this. (feature request)
P3	"Maybe **being able to select 2 types of events** would be cool"	P3 wanting further control over search criteria. (feature request)
P3	"That would be the fact that **you can't do the time scaling and still flip through the instances with the linked situated suggestions and ESO**."	P3 saying what was not helpful in one of her training sessions. We were able to add this feature request so she could use it in future sessions. (feature request)
P3	" Ideally **not having to manually create the patterns** each time"	P3 wanted to be able to create the patterns within the UI more easily. (improvement)
P3	"We talked about the **potential drag and drop feature** [drag the semi-intervals around and they would snap-to-grid]. I think that that would really speed up the analysis, **which I already think is pretty fast**."	P3 is discussing an improvement to the UI that would speed up the analysis. The describe improvement is something already in-the-works and is partially working. P3 also commented at the end how fast the approach is as-is.
P3	"I say strongly agree and then agree, mainly **because I have more sparse data so its not as useful for pairing stuff down** but **it is helpful in eliminating the ginormous amount of data that exists if they didn't have constraints.**"	P3 commenting on the usefulness of temporal constraints during training session three. Since she has sparse data, temporal constraints are not as useful but still she found them helpful in certain ways. (negative comment)
P3	I'll give those 4's just because **sometimes I wasn't entirely sure why somethings showed up and why some others didn't for the aggregate model. I think I just don't understand it as much**, but it's definitely useful in constraining what's actually shown to keep it to useful intervals.	P3 commenting on usefulness of temporal constraints and their ability to improve their ability to perform research during IS1(question 2 of semi-structured interview). She had trouble with understanding the factors dictating the temporal constraints of the model and its context. However, from what she did understand, she still found the constraints useful. (negative comment)
P3	"We **found some that matched up but some that were too far apart** [through disabling next constraint] **to be meaningful**."	P3 commenting on relevance of matches (question 7 of semi-structured interview) during TS2. Relaxing the temporal constraints resulted in meaningless results for her analysis goal.

Figure 5.25: Excerpts from participant transcripts highlighting critiques of the system and approach.

Praises

Participant	Quote	Explanation and Context
P1	"... **the fact that it has held out for all these other sessions is pretty interesting, that we can get most of the way there with just a simple pattern."**	P1 commenting in IS3 on how the approach was still able to work well despite the coded events not always representing turn taking.
P1	**"Being able to specify these kinds of patterns** like we are searching for is really good."	P1 in IS4 commenting on the expressiveness of the patterns. This also shows support for structural modification of patterns.
P1	"...**I am able to translate some of my qualitative interests into these patterns we are searching for...**"	In his final session, P1 comments on how he was able to translate his interests to patterns.
P2	"Ultimately what it does is not new, but the way it does it is pretty new"	This is P2 commenting in TS2 on descriptive models and whether it is a new capability. He noted the novelty of our modeling schema.
P2	"The visual part, **the ease of seeing where things are that you are looking for** and where are the things that surround the things you found [**context**] and **building models visually**. **Seems like it is a pretty versatile tool**, It can crack all sorts of data. Also, the video capability, which I could not use, but it's very promising. I really like that."	In TS2, when asked what was provided that was not previously provided (question 3 from semi-structured interview) P2 positively commented on the use of context and visual model building among other aspects. He also noted the versatility of the approach.
P2	"...**it allowed me to gather information that I needed for my hypothesis testing"**	Part of P2's final comments in his last session. He noted how the approach allowed him to collect what he needed to test his hypothesis.
P3	"I especially **liked creating the patterns and adjusting them to try to get the best fit**"	P3 positively commenting in TS2 about structurally adjusting patterns to find the best fit.
P3	"I really like being able to **get a range of patterns returned**"	P3 positively commenting in TS3 about the model matches provided through use of predicate models.
P3	"Just really being **able to look at all of these different combinations of patterns** that's going on to try to find out what the user behavior is quantified."	In TS3, when asked what was provided that was not previously provided (question 3 from semi-structured interview), P3 positively comments on the process as a whole in its ability to view patterns of various structures to better understand the user behavior in the data and explore the data.
P3	"Being **able to start off with just one little predicate model and all of a sudden seeing all of this stuff that is going on in the data** and how it all works together."	P3 positively commenting in IS1 on what was useful in the approach. She notes the benefit of predicate models in viewing the possibilities in the data.
P3	"**The entire pattern creation and exploration.**"	P3 positively commenting in IS3 on what was helpful about the approach. She found the pattern creation of the approach and the exploration facilitated by it was useful.
P3	"I also liked being **able to change the models pretty simply**. Especially the variables."	P3 positively commenting in IS4 about predicate models. She like the simple ability to change the models including the support for variables.
P3	"I'd say the **smoothness in which I could change the query into the data**. For example u**sing the *'s versus variable X and variable Y to get different meaningful results**[predicate]. Like, I was looking at did this pattern happen at all and then did this pattern happen when an action was taken by one person and then an action was taken by the other person [using *'s versus using variables]. I liked being able to do that. And **crank through everything quickly**."	In IS4, when asked what was provided that was not previously provided (question 3 from semi-structured interview), P3 positively commented on the ease at which to change the query especially with predicate support.
P3	" **Letting the system do some of the thinking for me**, that I didn't have to do a lot of manual counting or trying to differentiate between S1 and S1 or S1 then S2 patterns [talking about binding to vars to get an actor order]. **I just kind of let it [the system] do it for me**. That was nice."	In IS4, when P3 was asked what was helpful, she positively commented on how the approach was able to take some of the cognitive load from her and let her focus on the analysis.

Figure 5.26: Excerpts from participant transcripts highlighting praises of the system and approach.

5.2.12 Discussion

Overall, our use-cases have provided support that our approach does aid behavior analysis. Each participant was able to successfully utilized our approach to facilitate their analysis goals. Through this we were able to satisfy the requirements of phase 2 and 3 of our evaluation. We have tested our approach on multiple datasets (phase 2) and gathered positive feedback from users on the relevance of identified *models* (phase 3). We were also able to address the three points at the beginning of this section (Section 5.2): 1) provide further support for TEBA as an analysis approach, 2) that experts not as familiar with TEBA can benefit from it, and 3) our support for TEBA aids in analysis. The results of our use-cases have shown that we were able to provide support for TEBA (1) through our interactive, *model* evolution strategy. Our approach (IRSM) was able to aid our participants in performing TEBA even though they were not familiar with TEBA (2). Our support for performing TEBA was able to aid our participants in their analysis (3).

Another interesting note is that we observed our participants using our system in unexpected yet beneficial ways. Our participants mainly used IRSM as a hypothesis tester. They would provide an initial hypothesis (*model*), see how it matched in the data, learning about the hypothesis and/or learn about the data through the matches, then updated the hypothesis based on what was found. Despite this unexpected use, it was beneficial to the participants and lends itself as evidence to the versatility of our approach.

Another observation was that our initial idea of *model* growth did not match the goals of our

participants as much as we expected or at least, in the way we expected. Our participants still "grew" *models* but through learning about their data and refining their view of how the behavior they were looking for existed in their data (identifying variations). Growing was mainly performed through re-specification of their initial *model* iteratively (as mentioned above) until they achieved the desired matches (i.e., *model* formulation that matched relevant occurrences). Each iteration was used as a probe into the data to inform the creation of the next iteration.

In terms of how our participants searched their data, we observed how each participant, each faced with different challenging characteristics in their data, were able to successfully search their data and identify relevant instances to their respective analysis goals. Through these observations we can make some recommendations of how to approach searching data with different temporal characteristics.

We first will discuss what was successful. If the data is dense (e.g., temporally tight) either throughout or in clusters, *model* and *context constraints* can aid in navigating the dense data and not provide overwhelming results. If the data is sparse, *loose constraints* help in finding matches. For clustered data, *model constraints* perform well in staying focused on results within the clusters whereas *loose constraints* can be used to identify events between two or more clusters. For data that exhibit more than one of these temporal characteristics, the strategies applied for each characteristic can be used in combination. However, the success of this depends on the search goal, hence, testing several permutations of strategies may be needed.

We observed that *normalization* performed well for aiding in viewing the order of events by removing any temporal variability. *Filtering* was an excellent approach to removing any distracting event or unnecessary information. This allowed our participants that used it to focus on the characteristics of their data that were important to them.

As far as what was unsuccessful or could cause problems, *loose constraints* may not work well by themselves as the results may have events that are temporally too far apart to be useful (assuming this is not desired). However, using *loose constraints* with the absence operator performed well (also assuming this kind of relationship is desired). Overall, caution is merited in using *loose constraints* by themselves. *Model Constraints* do not perform well in sparse data, especially if the data has a wide temporal variability. However, *model constraints* can be used successfully in sparse data to identify events that are more temporally close than others. *Normalization* removes any relative timing information for events in terms of event length and relative timing between events. Removal of this information can skew the view of the data and misinform the observer if not careful.

We would like to conclude this section with a summary of what our participants learned about their data, what was meaningful to them, and how they plan to present their findings. The presented information here came predominately from the follow-up questions (Figure C.7) asked each participant after their sessions were completed.

P1 learned that "...the turn-taking structure [of his data] did reflect evidence of collaboration." This occurred many times in his data. However, he also learned that the contributions that were coded in his transcripts, and in turn the contents of the datasets used in conjunction

with our system, "...does not contain all the information necessary to recognize collaboration...". For further validation, he had to reference the original transcripts and video to gain any extra needed information. He discovered many cases of mis-codings in his data, which he found helpful. He also remarked that there is something to the simple *models* he used but they are not enough. There is a need "...to look at them in conjunction with the video." Hence, he found merit in the *models* he was working with (even if simple) but a coupling with the original data (e.g., video) is needed. As far as what was meaningful to P1, he stated "[k]nowing when a participant's contribution started or ended in relation to when another person started or ended, [and] also ... the ability to say that there was nothing between 2 events [absence operator or exclusion criteria]". P1 does plan to present the results of his analysis in a conference venue through providing background information of our approach and the analysis process but is unsure how to formulate the argument at this time. He said there is more work and analysis to be done before that can be solidified. Overall, he noted that since he had "lots of data, ... having a tool to narrow down where interesting cases might be, and quickly determine those places (quick search)" was very useful.

P2 learned there was no correlation where he hypothesized one to potentially exist and in order to find a correlation, he would need more users. He also observed the variety and unpredictability of human subjects which can cause challenges for verifying hypotheses, and hence, lead to new hypotheses. Since P2 was unable to verify the hypothesized correlation, he is not pursuing presentation of his results since he found his hypothesis to be inconclusive. However, the results did provide valuable information to him for how to prepare for future

experiments.

P3 stated that she "[s]tarting out ... thinking that [she] knew what [the structure of her data] was, but [she] ended up letting the data say what the structure was, [and the] tool was good is showing this." She learned about the "underlining structure of how the participants communicated with each other and the display". This aided her in finding evidence supporting her hypothesis. However, she noted that "not all of [her data] was properly coded and documented, [which was a] little discouraging, [and] makes it more difficult to support or refute the hypothesis". During the analysis what was meaningful to her was "extracting the *pattern* instances across participants to actually be able to see the little units of common understanding". The result of her exploration in the first three independent sessions was defining eleven simple *models* that represented "...the little units of common understanding" that aided her in finding evidence supporting her hypothesis. Even though she found good evidence supporting her hypothesis, she notes that "there is more to look at" and her current completed analysis with our approach "definitely prompted the need for further analysis, [and has] been promising so far in supporting the hypothesis". P3 also plans to present her analysis results either in a conference or journal venue but has not "thought about how to present it, [as she] need[s] to gather more data".

5.3 Phase 4 Evaluation

The final phase is to compare the results of our approach to that of users given the same task. A dataset will be chosen in which certain *behavior models* will be presented for identification within the dataset. We will compare the results of our approach to that of participants who identify the *models* independent of our process. This will allow us to provide precision/recall of our process with the baseline being human participants. This evaluation is beyond this dissertation and will be conducted after its completion.

Chapter 6

Software Versions

In this chapter we discuss the details of Temporal Relation Viewer (TRP), the tool that was developed to test the approach of this dissertation. We begin with the *prototype* which helped pioneer our ideas and was used in [95, 92]. After which, we discuss the development of *version 1* stemming from our prototype. *Version 1* was used for the experiments in several publications [93, 91]. We continued developing *version 1* to produce *version 2* (current version). This version incorporated many features that are very beneficial to supporting behavior analysis as was seen in our use-cases.

6.1 Prototype

Initially, we wrote a library in Python based on the n-gram mathematics described in [12]. This library implemented our first version of our comparison metrics described in Section

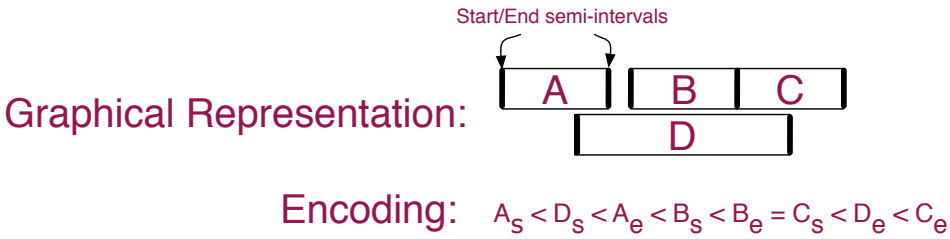


Figure 6.1: Example encoding from multi-channel data to linear sequence using semi-intervals.

4.1. The library is initialized with a set of event sequences and queried with a semi-interval relational sequence similar to the encoding in Figure 6.1 and seen in Figure 6.2. This query results in probabilistic relationships to potential *previous*/*next* events. This library provides forward and backward probabilistic viewing since a query cannot be assumed to always describe the beginning of a sequence.

The database used in our *prototype* was very simple. It consisted of a linear sequence of event semi-intervals separated by comparison operators ('<' or '='), see Figure 6.1. Using this encoding, we were able to develop algorithms to identify *related* semi-intervals to a specific query (Figure 6.2). Given identified instances of a query, we performed relational reasoning using the comparison operators in the linear sequence to provide what semi-intervals are *related* to the query *model* and how they relate (e.g., *previous*, *next*, and *current*, which was added during later development).

As more information was required to be stored and associated with each event semi-interval, a simple protocol was developed to encode this information. Our datasets were stored as a comma separated value (CSV) file with each entry between cammas consisting of a

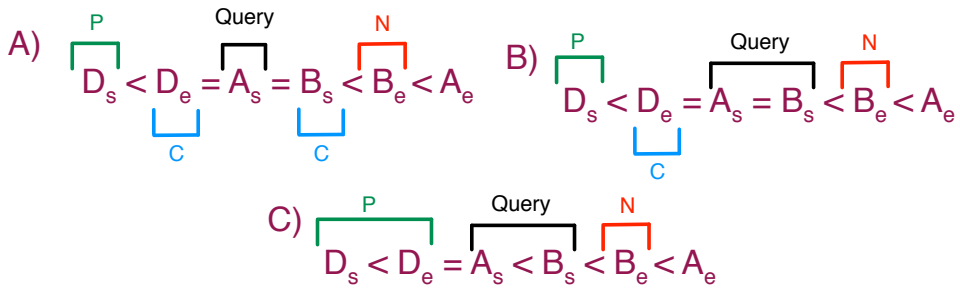


Figure 6.2: A) Example *previous* (P), *current* (C), and *next* (N) categorical relations of semi-intervals with respect to instance of query A_s. B) Query containing only equality. C) Query containing inequality.

colon separated sequence of values: description:position:time. Description was as free-form text describing the event, the position was either 's' or 'e' for a start or end semi-interval, respectively, and the time is the occurrence time of the event semi-interval. This allowed us to process a dataset and explore and identify query matches with limited capability as it was dependent on the descriptions' consistency when the dataset was originally created.

With this encoding and data representations, we performed linear searches through the linear sequence. Through this process we realized the concept of multiple instances of a *model* within the linear sequence. This lead to the realization of a situated view (looking at each instance *in situ*) and aggregate view (looking at results from all instances at once). During this time, we wanted a graphical query system. Hence, we took the linear processing describe above and provided a graphical front end to realize this. An overview screenshot can be seen in Figure 6.3. Here we present the Aggregate and Situated view each with graphical interaction with a *model/pattern* where parts of a *model* (e.g., an interval) could be selected and suggestion lists can be selected to view the respective suggestions. In the Situated

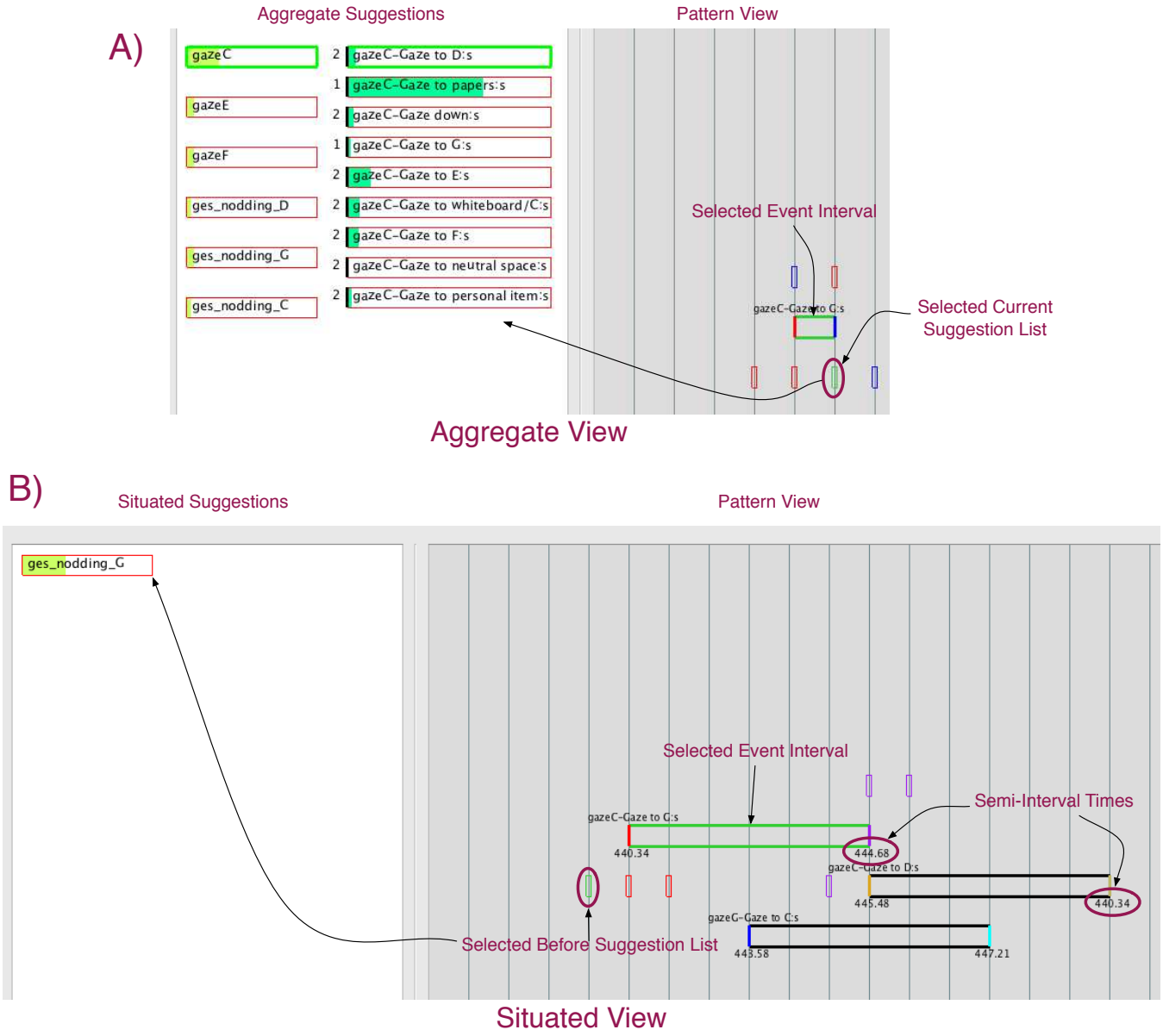


Figure 6.3: A) Overview of Aggregate View. B) Overview of Situated View.

View, the semi-interval times of the current situated instance being viewed is displayed. This version is also where we initially implemented the visualization for our suggestion representation, aggregate and situated suggestions (see Figure 4.7 for reference).

This first version of our system was used to complete the experiments of [95] and the demonstrations of [92].

6.2 Version 1

We performed continual development on our *prototype* for roughly a year. It provided a platform to test our ideas. During that time, we came to a better understanding of the problem we were trying to address, the data we were working with, and the needs of the analysis process. Hence, we took a step back and redesigned our *prototype* and began to start from the ground up using C++. Our *prototype* was written in Python, which is great for quick prototyping, but as a interpreted language, it is slow for any intense processing. Hence, switching to C++ to gain speed in processing.

This redesign lead to development of our *version 1*. In *version 1*, we separated the database and query processing side from that of the visualization. In this way we created a processing library, Temporal Relation Processing (TRP), and a visual from end, Temporal Relation Viewer (TRV). We will discuss the details of each in the following subsections.

Events Table			
eventID	**time**	**descriptionID**	**matchingEventID**
integer primary key	number	integer	integer

Description Table					
descriptionID	**description**	**actor**	**type**	**position**	**matchingDescID**
integer primary key	varchar2(2000)	varchar2(2000)	varchar2(2000)	char(1)	integer

Figure 6.4: SQLite Schema representing events of a corpus. Column names in bold with data type below.

6.2.1 Temporal Relation Processing Library

The purpose of the TRP library was to create a separate library that another program could link to in order to take advantage of the processing support of our approach. TRP provides support for querying a database (flexible support for multiple formats, currently SQLite is used) and performing different processing on the query results. The current processing support is ngrams with $N = 2, 3$ and the whole *model*. An extensive data structure was created to support the representation of *models* in our work.

With this new version, we retired our simple linear sequence database format for a more sophisticated representation. For this we turned to use a SQLite database (http://www.sqlite.org/). Through working with the data during our *prototype* stage, we learned what characteristics of an event seemed important to the expert as per the literature that motivated out work. Hence, we associated four characteristics with an event: description, actor, event type, and position. The description is a free form text description of the event. The actor is the source of the event, i.e., who performed the event. The event type is the kind of event, such as a gaze or speech event. Lastly, the position is whether the event is a start or end semi-interval.

Hence, given these characteristics of events, we created a SQLite schema (Figure 6.4). The schema consists of two tables. The first table lists all event semi-intervals ordered by time. Each semi-interval has a unique event identification number (**eventID**), the occurrence time (**time**), the unique identification number of its description (**descriptionID** - discussed shortly), and the unique identification number of the matching semi-interval in the table (**matchingEventID**). This table allows queries for events in time.

The second is a list of event descriptions where each interval event has two entries, one for the start and one for the end. Here a description consists of its unique identification number (**descriptionID**), a textual description (**description**), the actor involved (**actor**), the type of event (**type**) - e.g., modality, whether the start or end, 's', or 'e', respectively, (**position**), and the unique identification number (**matchingDescID**) of the matching description (start or end). These descriptive aspects of events are a subset of event aspects in [153], except for **position**. This table allows queries based on events' characteristics. These tables are also discussed in [91] and Section 4.3.3.

Version 1 was where we implemented our *pocket* representation discussed in [91] and Section 4.3.3. For ease of the reader, the pertinent figure is repeated in Figure 6.5 with an added illustration of how we iterate through the *model*. As can be seen in this figure, the implicit (ordered) relationships amongst the r_i's. For example, there is no need to explicitly store (remember) that r_3 and r_4 are equal or that r_2 occurs before r_6. Our implementation treats the semi-intervals of each pocket as a list. New semi-intervals added to a pocket are appended to the end. One can iterate through a *model* to view each semi-interval in-turn.

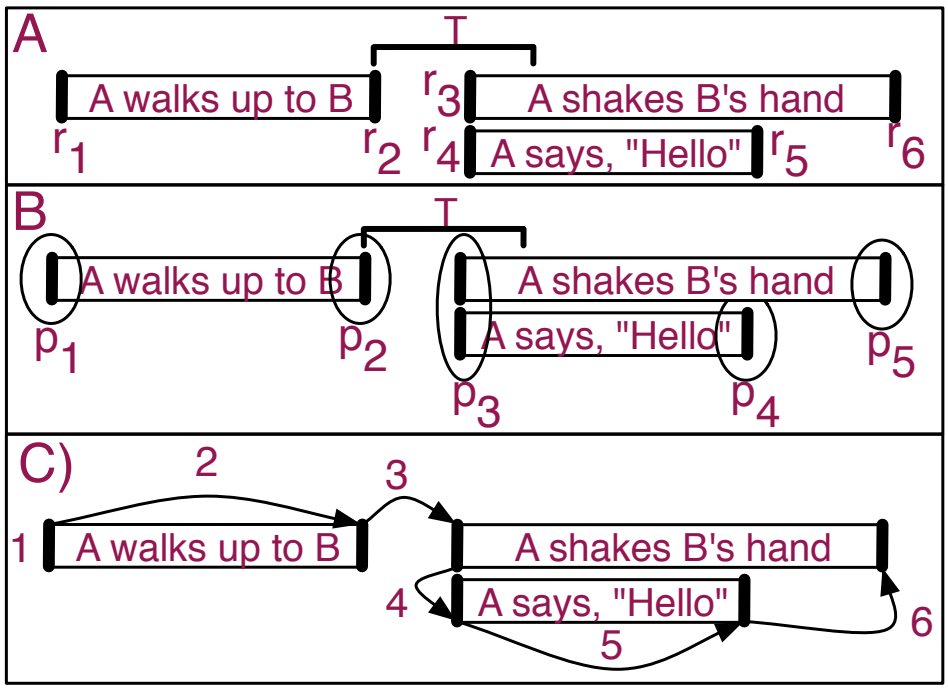


Figure 6.5: A) Example structure of a *model*. Note the temporal constraint between r_2, r_3, and r_4. B) Segmentation into *pockets* of equality. C) Example iteration path for visiting each semi-interval of a *model*.

The iteration path can be seen in Figure 4.9 C. The iteration algorithm pseudocode can be seen in Algorithm 2.

Algorithm 2 Pseudocode for *Model* Iteration

```
Given model R with P pockets
for i = 1 → P do
  Select p_i ∈ R
  for î ∈ |p_i| do
    Select r_î ∈ p_i
    Access r_î
  end for
end for
```

With the given iteration algorithm and SQLite database, we can iterate through a *model* to create a SQL query reflecting the *model's* ordered relational structure. As can be seen by the iteration path, a linear representation is established from the *model*. In *version 1*, we still use this idea of a linear representation (as in our *prototype*) but in a more sophisticated manner. Due to the nature of the events we process, viewing the semi-intervals as a linear sequence is natural and simplifies the problem as we look at the sequence of semi-intervals as they occur (whether in inequality or equality). We realize this linear representation is limited, however, even though simplistic, the results are powerful. Improvement upon this representation is part of future work. The procedure for creation of this query can be seen in Algorithm 3.

Overall, we look up instances of the first two semi-intervals (if only one, then instances of that one are identified) in the specified order and with the given *timing constraints*. Then we systematically perform successive joins for each additional semi-interval with respective constraints being applied at each join.

Algorithm 3 Pseudocode for Query Creation

Given *model* R with size $|R|$
Using Algorithm 2, iterate through R:
for $i = 1 \rightarrow |R|$ **do**
  Create a query q_i that locates all instances of r_i in database.
end for
With the created queries, compose a query to reflect the structure of R:
if $|R| = 1$ **then**
  return results of q_1
else if $|R| = 2$ **then**
  Perform join of q_1 and q_2 where ${r_1}^{t_1} < {r_2}^{t_2}$ and *timing constraint* $\hat{t}_1$ and *other* constraint c_1.
else
  Let $Q \leftarrow q_1$ join q_2 where ${r_1}^{t_1} < {r_2}^{t_2}$ and *timing constraint* $\hat{t}_1$ and *other* constraint c_1.
  for $j = 3 \rightarrow |R|$ **do**
    $Q \leftarrow Q$ join q_j where ${r_{j-1}}^{t_{j-1}} < {r_j}^{t_j}$ and *timing constraint* $\hat{t}_{j-1}$ and *other* constraint c_{j-1}.
  end for
end if
Composed query Q reflects the structure of R.

We use a nested join structure starting with the first two semi-intervals (if at least two), and successively process a join level for each semi-interval afterwards. Each level has *timing constraints* applied specifically with how the semi-interval relates with the prior. These constraints are stored separately from the *model's* structure, however, integration into the structure (e.g., Figure 4.9A) is part of future work. Another constraint is applied (c_i) which represents other special constraints. Currently, the special constraint we apply is to ensure that if two consecutive semi-intervals form a complete event (both start and end), then the occurrences matched are of the same event instance. However, due to the complexities of query generation, performing this constraint for semi-intervals that are non-consecutive (e.g., overlapping intervals) is done at the results level, i.e., filtering the results of Q. In order to

A) q_1 = SELECT events.eventID AS eventID1,
events.time AS time1,
events.descriptionID AS dID1,
events.matchingEventID AS maID1
FROM events WHERE descriptionID = r_1 ORDER BY time ASC;

B)
```
SELECT * FROM
  (SELECT * FROM
    (SELECT * FROM
      (SELECT * FROM
        (SELECT * FROM (q1 JOIN q2) WITH C1)
         JOIN q3 WITH C2)
       JOIN q4 with C3)
     JOIN q5 with C4)
  JOIN q6 WITH C5
```

C)

eventID1	time1	dID1	meID1	...	eventID6	time6	dID6	meID6
7805	8.66	335	7806	...	7808	12.79	342	7807
7811	109.72	335	7812	...	7814	114.46	342	7813
7817	298.45	335	7818	...	7820	303.61	342	7819
7823	476.06	335	7824	...	7826	480.38	342	7825

Figure 6.6: Example query construction for *model* in Figure 4.9. A) q_1 construction. B) Complete nested query. C) Subset of results.

accomplish this dynamic query building, we had to discover a modular query structure in SQL.

An example query using the *model* from Figure 4.9 can be seen in Figure 6.6. In Figure 6.6A we have the construction of q_1 in which each column of the table entry is given a unique identifier (renaming) which are used as reference in creating constraints. This renaming acts as a global reference throughout the query so constraints at one level can incorporate information from another. In Figure 6.6B we have the complete construction of the query with C_i being the associated constraints used in a particular SELECT. As can be seen, the query constructed has a nested structure with a level per semi-interval in the *model*. A subset

of results can be seen in Figure 6.6 C where matches of the first and last semi-intervals (r_1 and r_6 respectively) of four occurrence matches are shown.

6.2.2 Temporal Relation Viewer

The purpose of our Temporal Relation Viewer (TRV) is to provide an interface to define a *model*, query for it, and adjust/update according to the query results. An overview of the interface can be seen in Figure 6.7. Here, one can see the same pieces as in the prototype interface. The user is provided a *Pattern View*/*Model View* where they can graphically describe a *model* to query for. This view also allows them to manipulate a *model* in various ways, such as add via the aggregate/situated suggestions. Temporal constraints amongst the *model* semi-intervals can also be adjusted but such adjustment is done through hard-coded values (interface adjustments are added in *version 2*). The basic functionality of the Aggregate View (and *Aggregate Suggestions*) and the Situated View (and *Situated Suggestions*) remains the same. Most aspects of the *prototype* interface were copied over for *version 1*'s interface. Hence, this provided an interface with which an expert can still guide the formulation of a *model* based on identified occurrences in the data using either an aggregate or situated view.

This second version of our system was used to complete the experiments of [93] and [91].

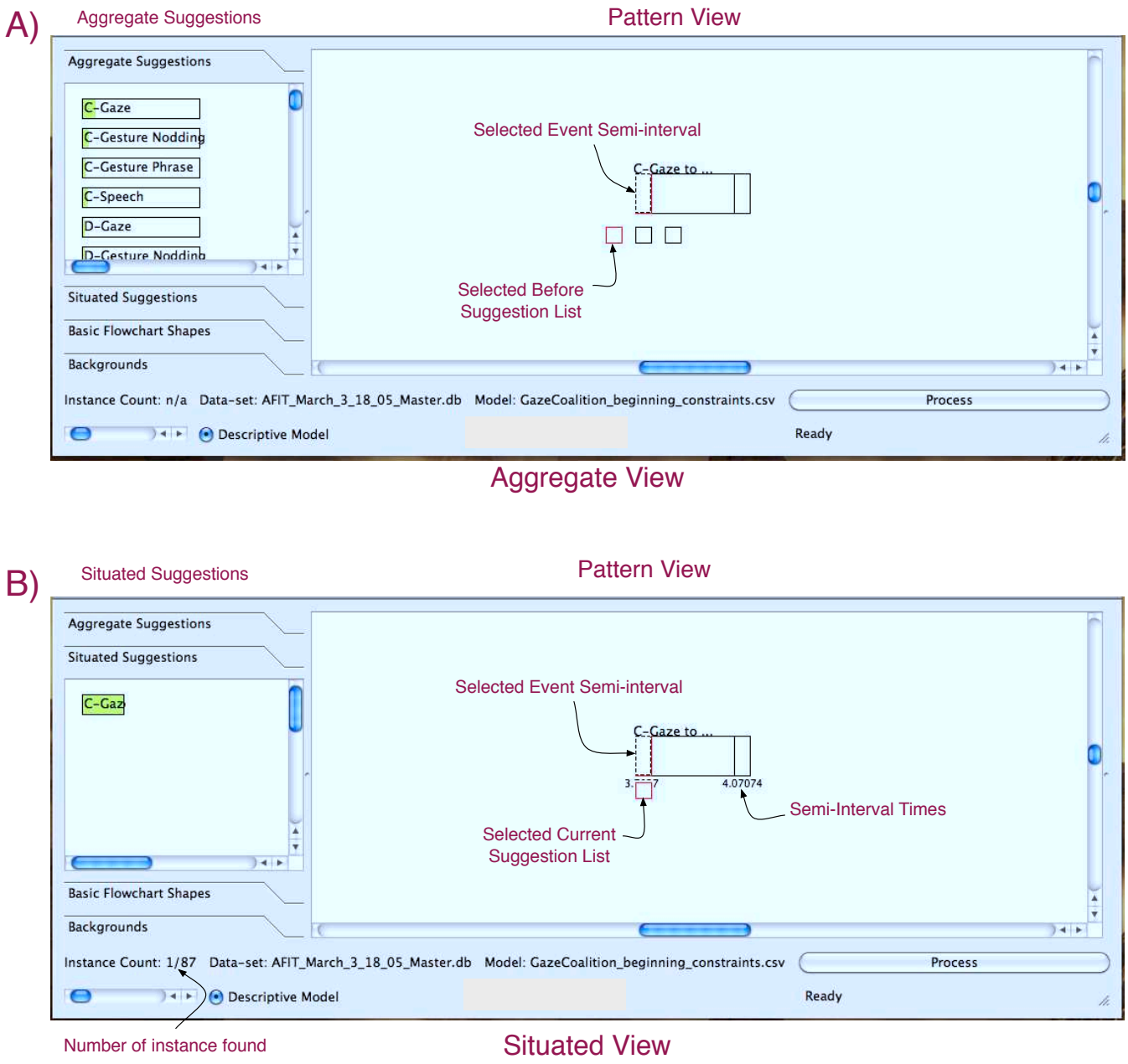


Figure 6.7: A) Overview of Aggregate View. B) Overview of Situated View.

6.3 Version 2

After the successful completion and experimentation of *version 1*, we began another phase of development to add more functionality that appeared necessary for supporting experts in analysis. This next phase brought our system to *version 2*. The main functionalities and features added were updates to the schema, greater interface support for adjustments to details of the *model*, predicate logic support, a simple timeline view of the database, linking to the video the events were extracted from (if applicable), and several other features requested by our participants in our use-cases. We will discuss each of these in turn.

6.3.1 Database and Parameter Adjustment Improvements

We realized that not only knowing the actor (source) of an event is important but who/what is the recipient of the event (if any). Hence, we added a *recipient* column to our *description* table of our SQLite schema. We also added a dialog (Figure 6.8) that can be opened for each semi-interval in the *Pattern View* where the user can change the now five characteristics of an event (description, actor, recipient, event type, and position).

Through this dialog, the user can also adjust various temporal constraints that dictate the matching of the *model* to instances in the database. These include adjusting what it means for semi-intervals to be in quality and the timing constraint between two consecutive semi-intervals (i.e., next constraint). There were several other temporal constraint controls added, but they were features requested by our participants, which are discussed below.

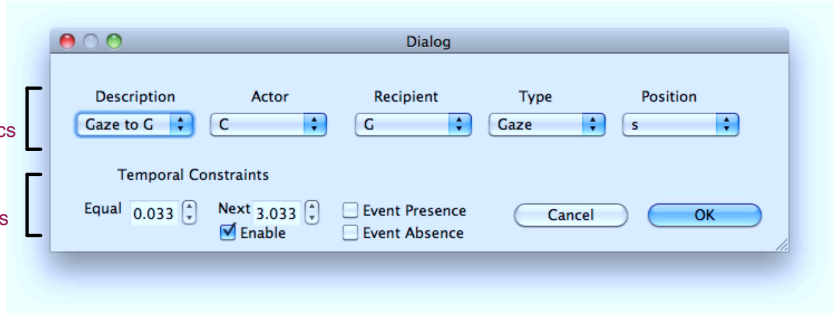


Figure 6.8: Edit dialog for each semi-interval in a patter. Here, the user can change any of the five characteristics of an event semi-interval, plus adjust the various temporal constraints.

6.3.2 Predicate Mode

Up until this point, we had been working with descriptive *models* with exact values. In this way we would specify specifically we wanted to find when Mary looks at Bob. However, it is also important for behavior analysts to define a generic structure without specific actors (or other characteristics) defined and see how the data fits the generic structure. In other words, using variables and wildcards instead of specific values. This is known as predicate logic and variable binding. Hence, we added the capability for the user to define variables for actors/recipients and wildcards (e.g., '*') for all characteristics. In Figure 6.9 we can see the added predicate mode. Here, we have a *model* defining a gaze exchange between actor X and actor Y. On the right are the different variable bindings (or *model* matches). These are the matches found in the data given the predicate *model*.

In Figure 6.10 is an example of selecting a *Model* Match. Once selected, the *model* is bound to those values of the match and can be treated as if a Descriptive *Model*. In other words, once a *Model* Match is selected, the user can view the Aggregate and Situated Suggestions

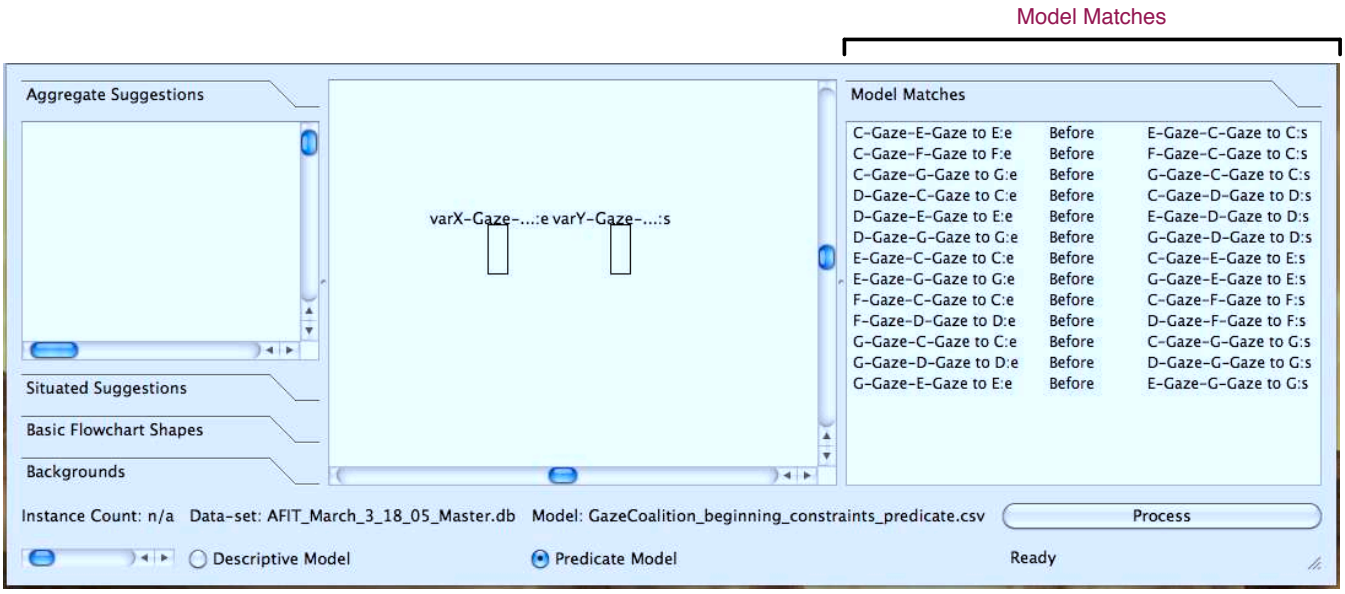
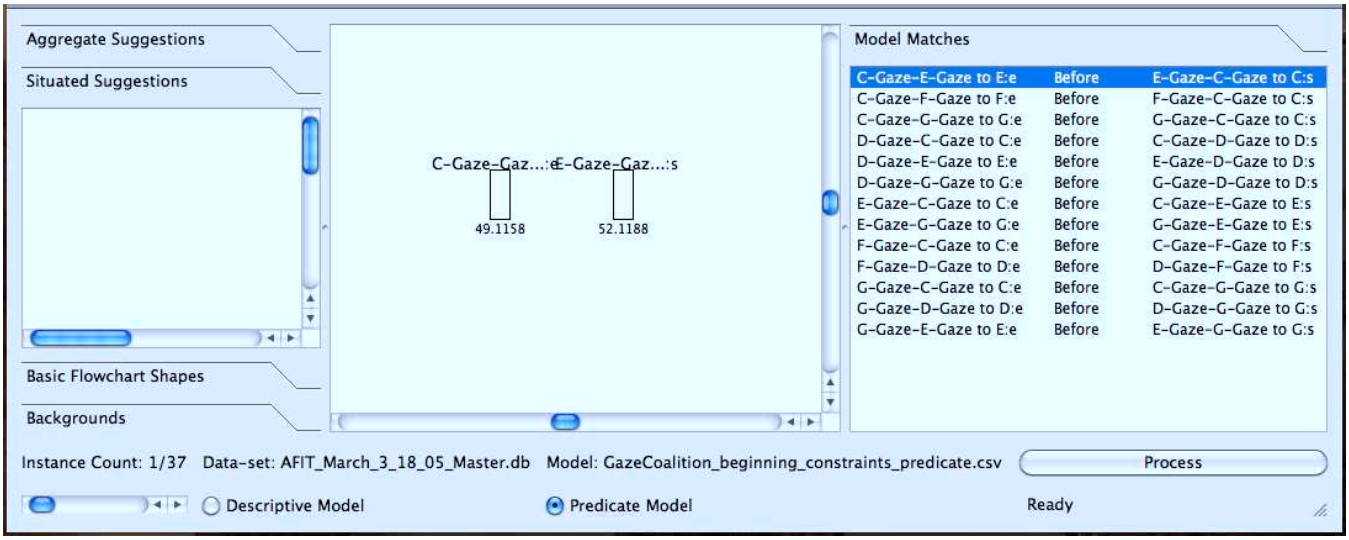


Figure 6.9: Screenshot of updated interface with included predicate mode. *Model* shown is for actor X (varX) ends a gaze fixation to actor Y (varY) and then actor Y starts a return gaze. The variable bindings (*model* matches) are listed on the right.

just as before. The figure shows the Situated View for the particular selected *Model* Match. When a semi-interval is added to the *model* in Predicate Mode, it is not a predicate since what is added is a semi-interval with specific values seen in the data. However, the user can make it predicate through the edit dialog for the semi-interval, as can be seen in Figure 6.11. Currently, variable binding is supported for actors and recipients and wildcards are supported for all five characteristics.

On of the challenges to variable binding is the state-space explosion problem characteristic to binding values to the variables. Fortunately, we were able to leverage SQLite in such a way as to let SQLite perform the variable binding for us. We were able to do this through adding constraints in the SQL query based on the defined variables and wildcards for different event



Predicate Mode

Figure 6.10: Example of a selected *Model* Match. Once selected, the *model* is bound to those values of the match. In this case, Situated View is selected for the selected *Model* Match.

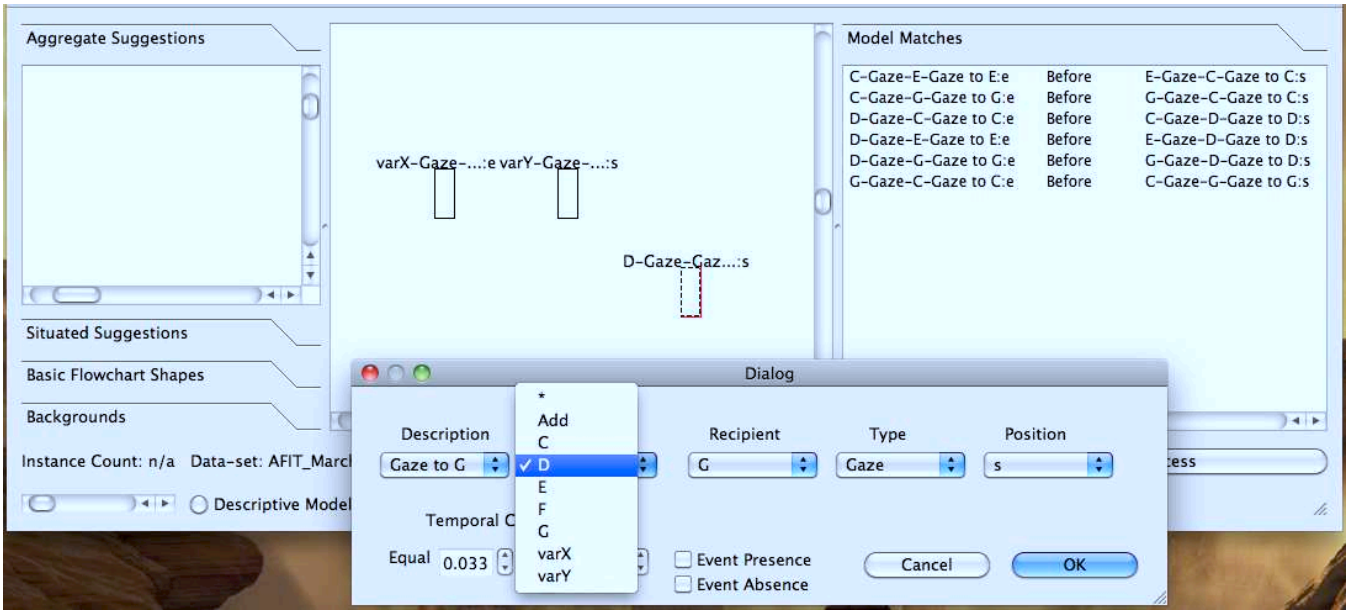


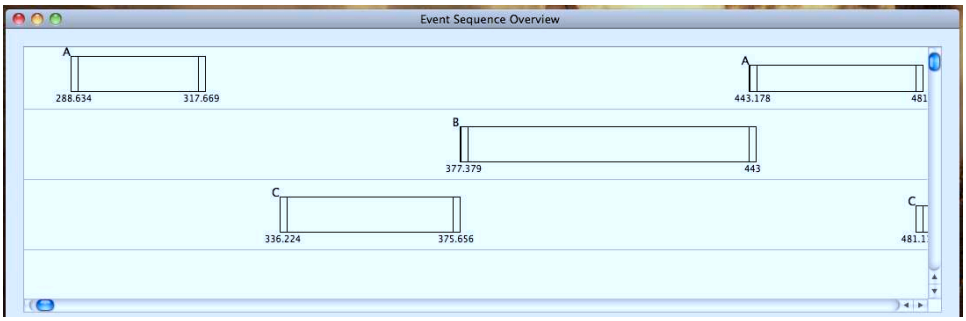
Figure 6.11: Example of the options available for adjusting a semi-interval in a *model* to be a predicate (i.e., change from descriptive to predicate). The user can choose an existing variable (here varX or varY), add a new variable, or choose the wildcard '*' to match anything for the particular event characteristic.

characteristics.

6.3.3 Event Sequence Overview

Initially this began as a simple means to view a database which we thought would be helpful to our use-case participants. It ended up being extremely beneficial. The Event Sequence Overview (ESO) is a simple visualization of the database in a "music score" fashion as illustrated in Figure 6.12. The database being viewed is one of Participant 1's datasets. The ESO allowed the users of our use-cases to view an overview of their data and, in some cases, perform visual scans of the data looking for particular *models*. Along with viewing the database, we added a link between the Situated View and the ESO. As seen in Figure 6.13, when a user is looking at specific situated instances of their *model*, the view of the ESO jumps to the semi-intervals of the current instance and are also highlighted. This allowed our participants to view the greater context of their situated instances.

One feature we added in anticipation of our use-case participants' data is changing the viewing timescale of the ESO. This initially came to mind when we were working with another dataset that consisted of events occurring over years. We realized that we may want to change the timescale to view the data at different time frames. For example, if the base time (i.e., the timescale of the recorded data) is in seconds, we may want to step back and view it in minutes. Hence, changing the timescale was added to the ESO interface, as seen in Figure 6.14.



Event Sequence Overview

Figure 6.12: Event Sequence Overview that displays the database in a "music score" fashion.

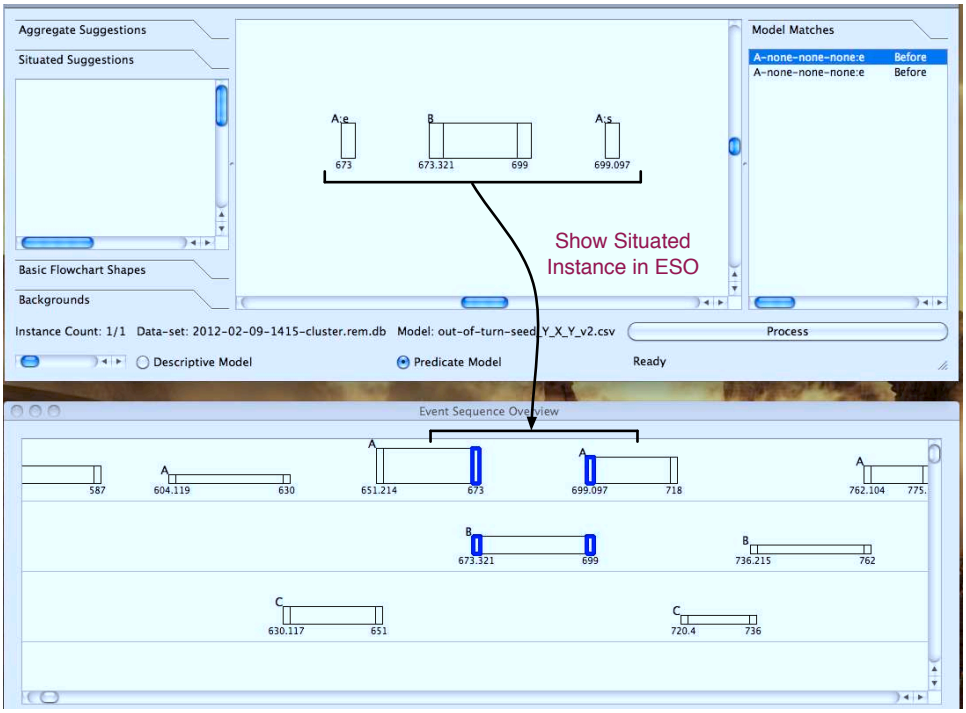


Figure 6.13: Example situated instance of a *model* occurrence being shown (linked to) in the ESO. What is shown is one of P1's datasets and the results of one of the predicate *models* used by the participant.

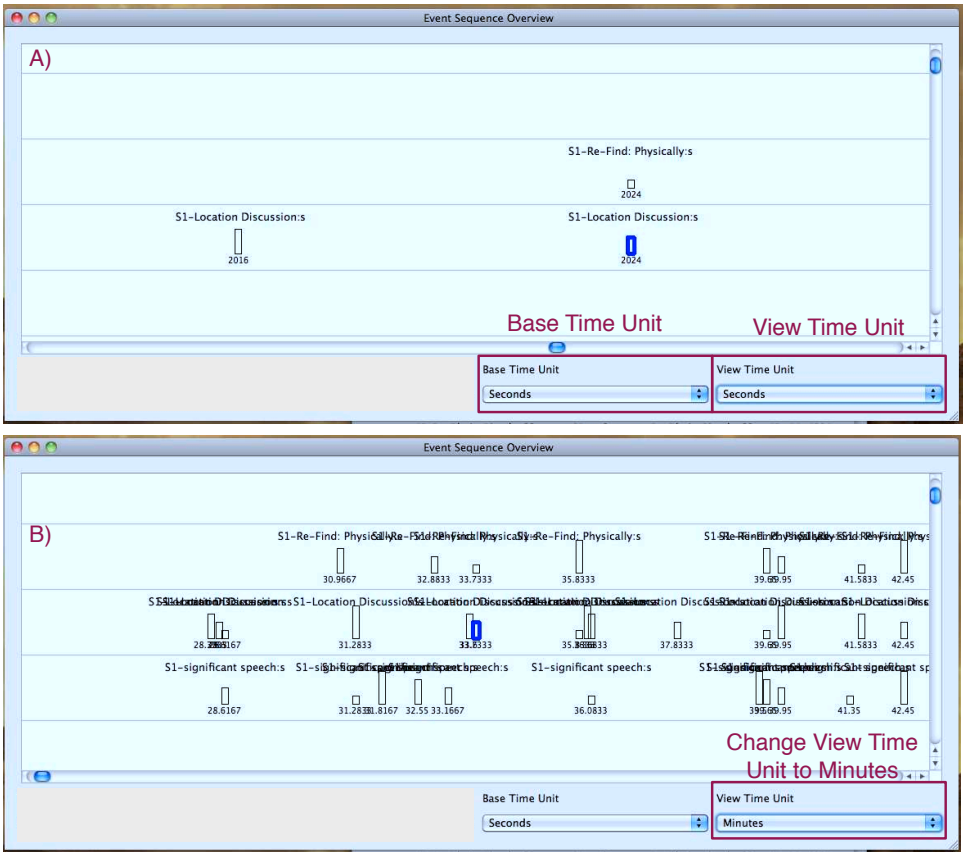


Figure 6.14: A) Timescale controls in ESO interface. B) Changing View Time Unit from seconds to minutes adjusts the view scale. The result of doing so is zooming out.

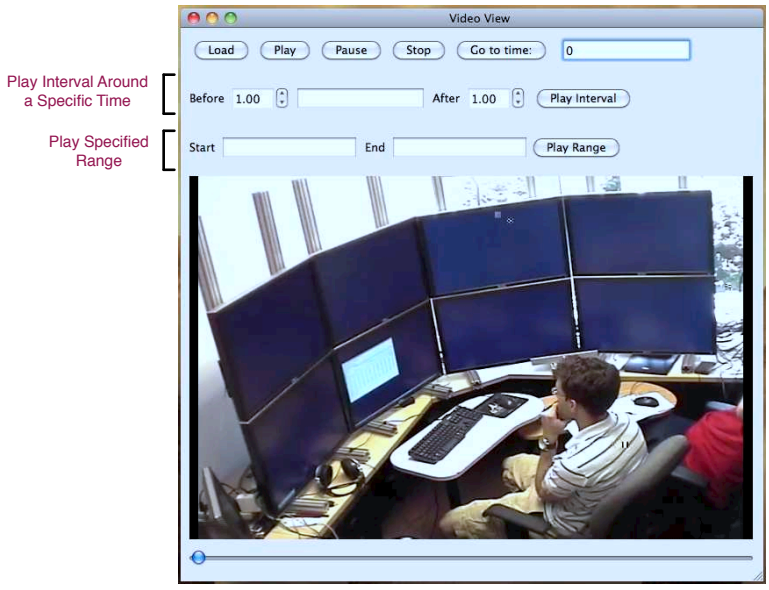


Figure 6.15: Screenshot of Video View. Highlighted are the controls for playing an interval or a range.

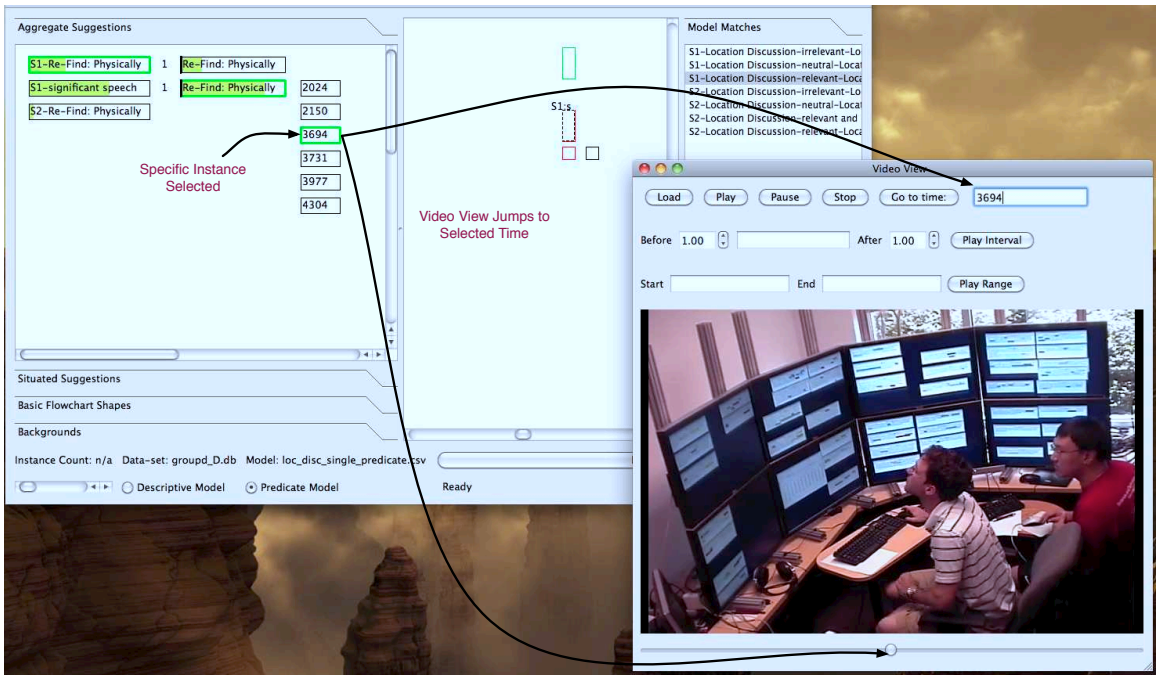


Figure 6.16: Example of looking through the Aggregate Suggestions and clicking on a specific instance time jumps the video to that time.

6.3.4 Video View

From the initial inception of our work, we knew that being able to link to and view any original video data associated with the current dataset being viewed would be very beneficial. Such a view is common amongst the many multimodal analysis tools, for reference see Section 2.1. A screenshot of our video view can be seen in Figure 6.15. We currently support the user to jump to a specific time and play an interval around that time or play a specific range in the video. We do allow jumping to a specific time when the user is looking at Aggregate/Situated suggestions and they click on a time box (Figure 6.16). Our participants requested greater linking between the ESO and the video view, especially for any shown situated instances that are highlighted in the ESO. These are part of our future work.

6.3.5 Features Added during Use-Cases

During our use-cases, our participants made a number of suggestions and feature requests which we were able to add in order to support their analysis. The first were a few different temporal constraints. This included support for a absence/presence operator which allowed the defining of the absence (or presence) of any event between two specified semi-intervals of the *model*. Participant 1 was very interested in the absence as he was interested in *models* where sections of the *model* were not interrupted by any other semi-intervals in his dataset. Hence, we were able to successfully test and utilize the absence operator. The presence operator works in the same way as the absence operator, however, since it has

not be required by the participants (so far) it has not been completed. The controls for absence/presence can be seen in the semi-interval edit dialog, Figure 6.8.

Next, while working with Participant 1, the *next* temporal constraint was too stringent as the events in his data were sometimes temporal close and sometimes not. Hence, he wanted a way to "relax" this constraint. In order to achieve this, we provided the ability to disable this constraint. However, this caused too many results, many of which overlapped with each other. Hence, we successfully applied the concept of non-overlapping support from FEM [115] and also discussed in Section 4.3. This was also added to the semi-interval edit dialog, Figure 6.8. Later, Participant 1 also suggested the addition of playing a range of a video instead of an interval around a specified time (seen in Figure 6.15).

Participant 3 required more control over the temporal constraints of the suggestions returned by TRP. Before this request, the participants could adjust the temporal constraints for matching their *model* in the database but not the temporal window for *previous*, *current*, and *next* aggregate/situated suggestions. Hence we added to the interface to provide such control as can be seen in Figure 6.17.

As the reader would notice, Participant 1 was the driving force behind most requested changes. This is due to his need to define more complex *models* than our system was originally capable of at the beginning of our use-cases. The subsequent changes performed due to his needs allowed the other participants to reap the benefits and possibly not require requesting such changes (as they were already made).

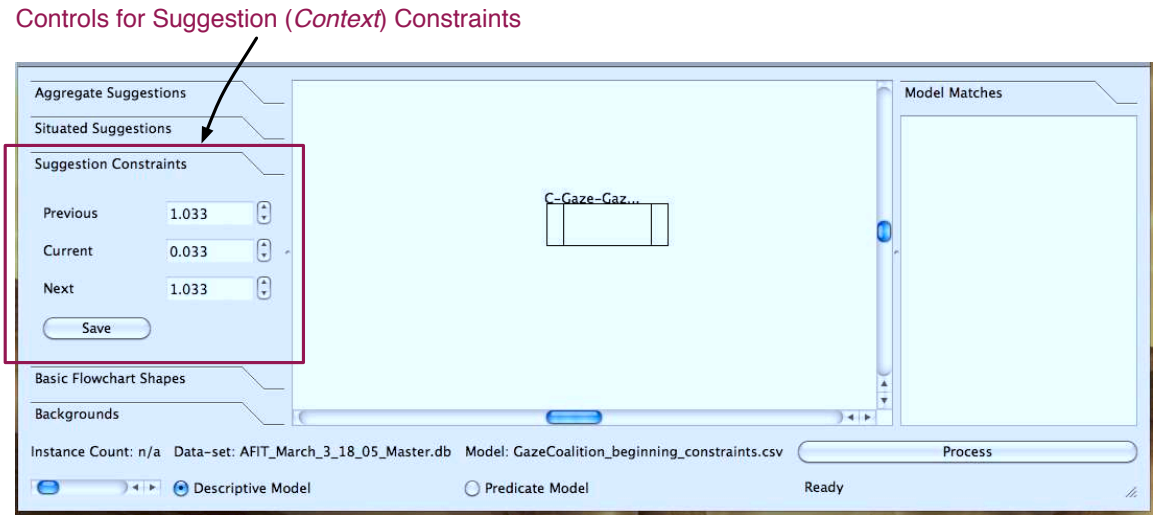


Figure 6.17: A new tab was added to the interface to accommodate the controls for suggestion constraints (*context constraints*).

Chapter 7

Conclusions

In this chapter we highlight the contributions of this dissertation, future work is discussed, and closing conclusions are given.

7.1 Contributions

In this section we highlight the contributions of this dissertation.

1. We have defined a *model* representation and a procedure to interactively refine the *model* structurally to a desired formulation given a starting point (initial *model* guess or assumption/approximation).

2. For the problem domain, we have demonstrated the feasibility of an interval-based representation to describe events in media streams and process them at the semi-

interval level.

3. Our approach identifies *models* that are difficult to identify using other approaches.

4. We demonstrated how to use our approach to support structural learning in the domain of temporal event analysis.

7.2 Addressing Research Questions

In this section we will revisit our research questions from Section 1.4 and provide how they have been addressed.

1. *How does one* ***explore and identify relevant models*** *within multimodal data?* Through our use-cases, we have observed how our approach has supported the participants in exploring their data and identifying relevant *models* within their multimodal data. This was made possible with providing a method to flexibly define *models* that reflected their interest and present situated results within context. In essence, the approach we developed discussed in Chapter 4 addresses this research question.

2. *How can one* ***represent*** *multimodal data to support and facilitate exploration and model identification?* The schema we developed that categorize event characteristics (time, actor, recipient, description, event type) and handle events at a semi-interval level resulted in an effective representation of multimodal data. This provided our participants with enough flexibility to define a *model* based on what is relevant to

them and search their data based on these event characteristics and temporal order. We were able to demonstrate how well this representation can perform through use-cases.

3. *How does one* ***seed such exploration****?* Through our use-cases, we observed our participants seeding their exploration by defining a relevant core of a *model*. This "core" represented a general structure or a piece of a *model* that represented a potential cue or flag where identified instances may signal the identification of something relevant. Overall, the participants would define sub-structures of relevant *behavior models*. Such defining was based on what the participants were looking for and what was coded in the data.

4. *How does one* ***formulate a model*** *that matches relevant instances in the data and can this formulation* ***support insight discovery****?* The formulation of a *model* was successfully done by our participants through defining *models* based on semi-interval events and their event characteristics. Participants did this with either descriptive or predicate *models*. With this support, they were able to formulate a *model* that did match relevant instances. We observed that the participants, with this formulation, were able to learn about their data through exploration. This allowed them to discover behavior existence (or not) leading to insight and understanding of their data.

7.3 Future Work

The successful completion of the body of work presented in this dissertation has provided many venues of continued pursuit. In this section we present our future work. The creation of a structural learning approach for temporal event data has opened many doors for further research. Besides addressing our phase 4 of evaluation, we have many directions for future work. We will discuss our current foci of future work in turn.

7.3.1 Continuing Use-Cases

Two of our participants (P1 and P3) are interested in continuing their analyzing using IRSM. Hence, for future work we will be continuing to study how these participants further search and discovery relevant *models* in their data. During this future study, we will be continually providing any other strategies requested, required, or discovered by the participants and hope to report on these at a future date.

Separate from this, four other student within Human-Computer Interaction at Virginia Tech are interested in using IRSM for analyzing their data. Two of these students want help analyzing data for their masters thesis while the other two have user study data they need analyzed for publication. Two of these students were recruited during a personal inquiry to their analysis needs while the other two heard of our work and expressed interest to using IRSM. We see this as building a foundation of work supporting our approach as a viable method for analysis. We are excited to see the impact of our research as a new analysis

method.

7.3.2 Journals

The successful publication of our approach in [95, 92, 93, 91] has provided a nice body of work which is suitable for a journal. We plan to prepare a journal paper on the technical details and experiments of our approach in more detail than is currently published. The target journal is ACM Transactions on Multimedia Computing, Communications and Applications (ACM TOMCCAP). The preparations of this dissertation has aided in organizing and preparing the material for this submission.

Through our investigations of temporal event analysis, we have observed in the literature how different domains handle, represent, and process temporal events. This observation provides a great opportunity to write a survey paper on how temporal event analysis is conducted in many domains. The target for this is ACM Computing Surveys.

7.3.3 Unified Representation

During our investigations, we realized that central to our research is a need to describe what the expert is looking for. Hence, an artifact of our research emerged as a set of temporal relationship principles. They are currently being developed for describing simple and complex behavior interactions within the context of discrete event sequences. A paper describing this is currently being prepared. This was a collaborative effort with our contribution being the

intricacies of *temporal constraints* (as discussed in Section 5.2.7), the methods using them, and the ideas (e.g., looping and interrupts found in behavior) leading to defining *semantic temporal constraints*. A *semantic temporal constraint* being a semantic inter-event relationship that constrains the temporal relationship of the events (e.g, prerequisite, interrupt, etc.).

7.3.4 Further Research Pursuits

Successful creation of an approach for structural learning in temporal event data has opened many doors fur further pursuit. Here we will briefly describe future research directions based on our findings.

Situated evolution: To date, our research has focused on human-in-the-loop interaction at each step. We now are investigating if there is a way to apply what we have learned to an automated process (e.g., data mining). Currently, one approach has been discovered and investigations are underway. Preliminary results are promising and we plan to run experiments comparing to Theme [146] as it is the only other approach known to us that is similar.

***Modeling* and finding *behavior models* of varying length:** Investigation into how different algorithms function with *behavior models* of varying lengths. We would like to compare the different parameter tunings of each algorithm and see the quality of the resulting *models*.

Structurally exploring linear and multi-channel media data-sets: An investigation focussed on how structural search and exploration can aid in identifying and discovering interesting *models*. This will be on multiple datasets with varying characteristics. The main focus here is how differing characteristics can affect structural search. These characteristics range from temporal characteristics (already explored in Section 5.2.7) to the characteristics goverened by the kind of data such as linear, several channels, and many channels.

Structurally learning new rules from a finite set: Investigation into structural machine learning where we will see if something new can be created from a set of known rules. Starting with a finite set of rules (e.g., the rules exhibited in several plays in a baseball game) we will see if we can discover new rules based on the starting set (e.g., create new legitimate baseball plays based on the initial rule set). Analogous to this is starting with the basics of mathematics (e.g., '+', '-', '*', and '/') and derive more complex mathematical rules. However, our focus will be on discovering new *behavior models*.

Event structure investigations:

- **Taxonomy of temporal structures:** Investigate the different characteristics that describe event datasets (part of this we have already explored as described in Section 5.2.7). Through this we hope to provide a way to categorize datasets based on their temporal structures, plus provide a subset of our relationship principles, in our representation discussed above, necessary to describe *models* in a particular category.

- **Aggregated recombination of multimodal events for discovery of new, rel-**

evant *models*: One observation in our work has been the ability to create a *model* based on the aggregate view [92, 93] that does not exist in the current dataset being viewed. However, said created *model* may be valuable as a new formulation of the *model* which may be useful for another dataset.

- **Organizational bins for mutimodal event content:** Organizing event information for choosing a suitable *model* extension is a challenging task. We have being using one method for such organization we call abstract segmentation where bins are based on labeled aspects of the events (e.g., actor, event type,...). We would like to investigate how well this organizational method works and seek out others.

Visualizations and interactions:

- **Interactive interface:** User studies are needed to help develop the interface.
- **Suggestions:** Deeper investigation into how to visualize and present suggestions to the user.
- **Graphical representations of *models* and *model* characteristics:** User studies and investigations are needed to best visually present and convey characteristics of *models*, e.g., show relationships and temporal constraints.

7.4 Conclusions

The work of this dissertation has presented an approach to aid behavior analysis. Our results provided evidence that our approach does in fact facilitate behavior analysis. This has been shown through successfully addressing our evaluation phases 1, 2, and 3. Addressing our challenging problem resulted in developing a new way of searching through temporal event data. We have developed an approach that aids in refining a *model* to match how occurrences actually exist in the data. This process can grow *models* to an expert's desired formulation based on the data and also test hypotheses about the data contents.

We presented a comparison of our approach against several pattern mining algorithms. Through this comparison we showed the viability of our approach's ability to accurately identify *model* occurrences. We then presented and discussed our longitudinal use-cases that resulted in our participants developing two analysis strategies that were different from the original intended use. This displayed the versatility of our approach. We also realized that we may have created a new breed of analysis approaches which we are excited to investigate further. The discovery and development of our approach has great potential to have a positive impact on the research community. Already at the finishing of this document, several fellow student researchers want to apply our analysis technique to their data.

Overall, we see the results of our work as impacting the multimedia search, multimodal analysis, and behavior analysis communities through presenting a newly developed analysis approach and detailed results of a longitudinal, hands-on experience with researchers ana-

lyzing temporal event data. We hope this spurs on further study of strategies and techniques for searching and identifying relevant *models* in temporal event data.

Appendix A

Theorem 1 Proof

Theorem 1 can be proven through induction:

Proof. Let our base cases be $\hat{T} = |\hat{G}| = 1$ where $2\hat{T} + 1 = 3$, and $\bar{T} = |\bar{G}| = 2$ where $2\bar{T} + 1 = 5$. Then $2\bar{T} + 1 = (2\hat{T} + 1) + 2$.

For the induction step, assume true for $T = |G| = n$ and $T - 1 = |G - 1| = n - 1$. Show true for $T + 1$. We know

$$2T + 1 = (2(T - 1) + 1) + 2$$

Then,

$$\begin{aligned} 2T+1 &= (2(T-1)+1)+2 \\ 2T+1 &= 2T-2+1+2 \\ 2T+1+2 &= 2T+1+2 \\ 2(T+1)+1 &= (2T+1)+2 \\ 2(T+1)+1 &= (2((T+1)-1)+1)+2 \end{aligned}$$

which is the same as $2T+1 = (2(T-1)+1)+2$ but for $T+1$ and $T \Rightarrow$ true for $T+1$

Hence, $\forall\ T \leq |G|$, $2T+1 = (2(T-1)+1)+2$, then the number of possible positions x for Y is $2T+1$. Assuming all the *S-I's* in G do not match Y as a beginning or end and since there are only three relative places to place Y for each *S-*I_t, $0 \leq t < T$, $2T+1$ is an upper bound. If Y's matching *S-I* is already in the *model*, then $x < 2T+1$ since an ending *S-I* cannot go before its matching beginning *S-I*. □

Appendix B

Model Occurrence Likelihood

There is a low probability that a *model* will be randomly generated within the data-set. Most *behavior models* in behavior analysis are temporally tight. Hence, we simplify proving this claim through focusing on small temporal windows within the data-set (i.e., small window along the timeline of the events in the data-set).

Given $\hat{T}$ number of unique time slots for a *S-I* to be placed in a timeline, let w be a fixed sized temporal window being $|w|$ time slots wide and x be the set of unique *S-I's* that may occur at each time slot (at least one per time slot), cardinality of $|x|$. Then, for a *model* m of size $|m|$, an estimated probability for m to be present is:

$$\frac{(\hat{T} - |w| + 1)(\frac{1}{|x|^{|m|}})\binom{|w|}{|m|}}{|w| - |m| + 1}$$

Proof. For each window, the number of ways to choose $|m|$ amongst w's time slots is $\binom{|w|}{|m|}$.

We are looking for a specific sequence chosen from x. Note that an *S-I* in x can only be chosen once per time slot due to the nature of the data despite multiple overlapping channels (e.g., cannot have two occurrences of person A's gaze to person B at the same time). Hence, each *S-I* in x has a $\frac{1}{|x|}$ probability of occurring at each time slot.

Since there are $|m|$ *S-I's* in the *model*, then there is a $\frac{1}{|x|^{|m|}}$ probability of choosing the *model S-I's* and $\binom{|w|}{|m|}$ ways the *model* can appear in the window. Therefore, $\Pr(model \text{ in } w) = \frac{1}{x^{|m|}}\binom{|w|}{|m|}$. However, this is for one window and we need to account for all windows, including overlap between them. Taking a sliding window approach, start with w_0 (first window in the timeline), then slide w_0 to the right one time slot producing w_1. Continuing this process produces $\hat{T} - |w| + 1$ windows. Hence,

$$Pr(m \text{ in timeline}) = (\hat{T} - |w| + 1)(\frac{1}{|x|^{|m|}})\binom{|w|}{|m|}$$

Since we are using a sliding window, there is a chance for double counting an instance of m. Hence, after w_0, we are only interested in *models* that end in the last time slot of w_i, $0 < i < \hat{T} - |w| + 1$. The probability of a *model's* ending *S-I* occurring in the last time slot is $\frac{1}{|w|-|m|+1}$. Therefore, the probability of a *model* m being generated is approximately:

$$Pr(m \text{ in timeline}) \approx \frac{(\hat{T} - |w| + 1)(\frac{1}{|x|^{|m|}})\binom{|w|}{|m|}}{|w| - |m| + 1}$$

This is an approximate probability as the complexity of the generated data-set leads to many variables to consider. An exhaustive proof is unnecessary as it can be seen from the approximation that the likelihood of a specific *model* being generated is very low. For

example, our generated data-set has $\hat{T} = 21,855$ and $|x| = 240$. Give the smallest *model* inserted where $|m| = 4$, and $|w| = 10$, the $Pr(m \text{ in timeline}) \approx 0.000197618$, hence for all 10 is 0.00197618. □

Appendix C

Use-Case Documents

VirginiaTech

Office of Research Compliance
Institutational Review Board
2000 Kraft Drive, Suite 2000 (0497)
Blacksburg, VA 24060
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: March 29, 2013

TO: Francis Quek, Chreston Allen Miller

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires May 31, 2014)

PROTOCOL TITLE: Support for Expert Data Analysis

IRB NUMBER: **12-1005**

Effective March 29, 2013, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the Amendment request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 6,7**
Protocol Approval Date:	**December 4, 2012**
Protocol Expiration Date:	**December 3, 2013**
Continuing Review Due Date*:	**November 19, 2013**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:

Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Figure C.1: The IRB approval letter for our use-cases (page 1).

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
12/04/2012	09223904	National Science Foundation	Compared on 12/04/2012

* Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

Figure C.2: The IRB approval letter for our use-cases (page 2).

Background Questionnaire

Participant: _____

What kind of data analysis have you previously conducted?

What kind of data analysis/visualization software have you used?

How many years have you conducted research?

Academic Status: Masters PhD

Sex: Male Female

Age: _____

Figure C.3: The background questionnaire given to each participant.

This is the script to be used for training the participants in the functionality of the system.

- Go over informed consent
- ▼ Introducing the system and showing summary list
  - Operates on temporal point and interval event data to identify patterns based on some user driven starting point.
- ▼ Procedure for training
  - *Open participant dataset.*
  - *Load model 1.*
  - ▼ Show/Describe Pattern view
    - Displays the current model/pattern that is the query into the dataset.
    - Each semi-interval can be selected revealing up to 3 links to suggestions of semi-intervals that can added to the model.
    - These suggestions can occur before, concurrent, or after the selected semi-interval.
  - ▼ Explain Aggregate Suggestion View
    - *Select end semi-interval. Select link for after suggestions.*
    - Selection displays aggregate suggestions organized in bins.
    - Explain bin structure, fill of 1st and 2nd level, confidence, bold lines, and time boxes.
  - ▼ Explain Situated Suggestion View
    - Look at the suggestions for a particular instance
    - ▼ Show linking to Event Sequence Overview
      - Also shows context
      - Zoom in/out
  - ▼ Show adding to model from Aggregate Suggestion view (can also be done in Situated Suggestion View)
    - *Add semi-interval to end of model.*
  - ▼ Show deletion from the model
    - *Removed added semi-interval of previous step.*
  - ▼ Explain Video View
    - *Load a video if applicable*
    - show linking to video
  - ▼ Show edit dialog for each semi-interval
    - Can change any of the 5 aspects of an event
    - ▼ Can adjust temporal constraints
      - Explain these are used for matching in dataset and not for suggestions
  - ▼ Explain Predicate mode
    - *Load model 2*
    - Explain how one can use a variable instead of a set value
    - Explain wildcards - match anything
    - Not all semi-intervals need to have variables/wildcards. Values can be set for any or all semi-intervals.
    - ▼ Explain Model Matches
      - The different variable bindings found in the dataset.
      - Show selecting different matches updates the pattern to that binding.
      - Once a match is selected, it can be treated as if it were a descriptive model.
      - The only difference is that if a suggestion is added to the model, it is not a predicate but has set values (which can be edited if wanted)
  - ▼ Save results
    - Save all situated results
    - Save the current one being viewed
  - ▼ Bonus features - features added either by request of participants for by investigators after the start of the study
    - Directly add a semi-interval to beginning/end and fill in according to something you want/see in the data
    - Absence/presence of
    - Other: more may or may not be added dependent on participants requests for features
- ▼ Q and A
  - Give participant a chance to ask questions
  - Give participant a chance to try anything they are interested in.

Figure C.4: The training script used during the first training session.

Feature list:

- Load a dataset
- Load a model/pattern
- Press "Process" button anytime to query for the current pattern shown

▼If descriptive pattern, you can do any of the following

- ▼Go to aggregate view (default)
  - Select any semi-interval and look at before, current to, or after suggestions
  - Select a suggestion to add to the pattern
- ▼Go to situated view
  - ▼For each situated instance
    - ▼Select any semi-interval and look at before, current to, and after suggestions
      - Select a suggestion to add to the pattern
    - View in Event Sequence Overview

▼If predicate pattern

- Select a Model match
- For each selected model match, treat as a descriptive pattern

▼Editing a semi-interval

- ▼The event aspects of every semi-interval can be manually changed
  - Description, actor, recipient, event type, position, temporal constraints

- Add a semi-interval not seen as a suggestion to the start/end of the pattern
- Delete a semi-interval from the pattern
- View a section of video by time stamp (if applicable)

▼Event Sequence Overview

- Zoom in/out

- Save current model

▼Save results

- All situated instances
- Current one being viewed

Figure C.5: Feature list for participants' reference during sessions.

Session Interview Questions

Participant: _____

1) Describe your experience in this session using a model growth/exploration strategy?

2) Does each of the following allow you to do something that you have never been able to do before or does it replace/augment a prior tool/process: New, Augment/Replace

Aggregate View, Situated view, Event Sequence Overview, Temporal Constraints

Descriptive Models, Predicate Models, Abstract Segmentation, Video View,

Other _____

For each:

New capability or Augment/replace

What tool/process does it augment/replace?

This capability is useful.

[strongly agree, agree, neutral, disagree, strongly disagree]

This capability improves my ability to do research:

[strongly agree, agree, neutral, disagree, strongly disagree]

3) What was provided that was not previously provided by other systems/approaches?

4) Did the model growth/exploration strategy help the process of discovery?

[strongly agree, agree, neutral, disagree, strongly disagree]

5) Overall, what was helpful?

What was not?

What could be improved?

6) Other comments?

7) For pattern(s)/model(s) that were either used or identified, the instances matching those pattern(s)/models(s) were relevant.

[strongly agree, agree, neutral, disagree, strongly disagree]

Figure C.6: Semi-structured interview Questions

Follow-up Interview Questions

These will be presented in a semi-structured interview fashion (just like the session interviews).

Participant: _____

1) What was your hypothesis?

    Was it supported? Why/Why not?

2) What did you learn from the results of this analysis?

3) What was meaningful to you?

4) What did you learn about your data?

5) How do you plan to present what you learned?

6) Does the described analysis strategy represent/describe how you conducted your analysis?
   (for each participant, I will describe the analysis strategy observed for them individually and then asked this question)

Figure C.7: Follow-up Semi-structured interview Questions

Bibliography

[1] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Chapman & Hall/CRC, 2009.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3 –14, Mar. 1995.

[3] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.

[4] C. Andrews, T. Henry, C. Miller, and F. Quek. Cardtable: An embodied tool for analysis of historical information. In *Tabletop 2007*, 2007.

[5] M. Asaari and S. Suandi. Hand gesture tracking system using adaptive kalman filter. In *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*, pages 166 –171, 29 2010-dec. 1 2010.

[6] A. Bastian. Identifying fuzzy models utilizing genetic programming. *Fuzzy Sets and Systems*, 113(3):333 – 350, 2000.

[7] R. Bednarik, S. Eivazi, and M. Hradis. Gaze and conversational engagement in multiparty video conversation: an annotation scheme and classification of high and low levels of engagement. In *Proceedings of the 4th Workshop on Eye Gaze in Intelligent Human Machine Interaction*, Gaze-In '12, pages 10:1–10:6, New York, NY, USA, 2012. ACM.

[8] H.-G. Beyer. *The Theory of Evolution Strategies*. Springer, 2001.

[9] H.-G. Beyer and H.-P. Schwefel. Evolution strategies âa comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[10] O. Brdiczka, N. M. Su, and J. B. Begole. Temporal task footprinting: identifying routine tasks by their temporal patterns. In *Proceedings of the 15th international conference on Intelligent user interfaces*, IUI '10, pages 281–284, New York, NY, USA, 2010. ACM.

[11] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[12] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, 1992.

[13] H. Brugman and A. Russel. Annotating multi-media / multimodal resources with elan. In *In proceedings of LREC*, pages 2065–2068, 2004.

[14] M. C. Buchanan and P. Zellweger. Scheduling multimedia documents using temporal constraints. In *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 237–249, London, UK, UK, 1993. Springer-Verlag.

[15] M. C. Buchanan and P. T. Zellweger. Automatic temporal layout mechanisms. In *Proceedings of the first ACM international conference on Multimedia*, MULTIMEDIA '93, pages 341–350, New York, NY, USA, 1993. ACM.

[16] P. Budhiraja and S. Madhvanath. The blue one to the left: enabling expressive user interaction in a multimodal interface for object selection in virtual 3d environments. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 57–58, New York, NY, USA, 2012. ACM.

[17] R. A. Calix and G. M. Knapp. Affect corpus 2.0: an extension of a corpus for actor level emotion magnitude detection. In *Proceedings of the second annual ACM conference on Multimedia systems*, MMSys '11, pages 129–132, New York, NY, USA, 2011. ACM.

[18] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The ami meeting corpus: A pre-announcement. In S. Renals and S. Bengio, editors, *Machine Learning for Multimodal Interaction*, volume 3869 of *Lecture Notes in Computer Science*, pages 28–39. Springer Berlin / Heidelberg, 2006.

[19] J. Charoensuk, T. Sukvakree, and A. Kawtrakul. Elementary discourse unit segmentation for thai using discourse cues and syntactic information. In *9th National Computer Science and Engineering Conference (NCSEC)*, 2005.

[20] K.-T. Chen, W.-T. Chu, M. Larson, and W. T. Ooi. Acm multimedia 2012 workshop on crowdsourcing for multimedia. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 1505–1506, New York, NY, USA, 2012. ACM.

[21] L. Chen, M. Harper, A. Franklin, T. Rose, I. Kimbara, Z. Huang, and F. Quek. A multimodal analysis of floor control in meetings. In S. Renals, S. Bengio, and J. Fiscus, editors, *Machine Learning for Multimodal Interaction*, volume 4299 of *Lecture Notes in Computer Science*, pages 36–49. Springer Berlin Heidelberg, 2006.

[22] L. Chen, M. Harper, and Z. Huang. Using maximum entropy (me) model to incorporate gesture cues for su detection. In *Proceedings of the 8th international conference on Multimodal interfaces*, ICMI '06, pages 185–192, New York, NY, USA, 2006. ACM.

[23] L. Chen and M. P. Harper. Multimodal floor control shift detection. In *Proceedings of the 2009 international conference on Multimodal interfaces*, ICMI-MLMI '09, pages 15–22, New York, NY, USA, 2009. ACM.

[24] L. Chen, R. Rose, Y. Qiao, I. Kimbara, F. Parrill, H. Welji, T. Han, J. Tu, Z. Huang, M. Harper, F. Quek, Y. Xiong, D. McNeill, R. Tuttle, and T. Huang. Vace multimodal meeting corpus. *MLMI*, pages 40–51, 2006.

[25] V. Cheng, C.-H. Li, J. T. Kwok, and C.-K. Li. Dissimilarity learning for nominal data. *Pattern Recognition*, 37(7):1471 – 1477, 2004.

[26] Chipmunk. *http://code.google.com/p/chipmunk- physics/*, Last Checked: Aug., 2010.

[27] C.-P. Chou and P. M. Bentler. Model modification in structural equation modeling by imposing constraints. *Comput. Stat. Data Anal.*, 41(2):271–287, 2002.

[28] P. Cohen. Fluent learning: Elucidating the structure of episodes. *AIDA*, pages 268–277, 2001.

[29] D. Conrad and M. Hurles. The population genetics of structural variation. In *Nature Genetics*. 2007.

[30] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[31] R. Cowie, E. Douglas-Cowie, S. Savvidou, E. McMahon, M. Sawey, and M. Schröder. 'feeltrace': An instrument for recording perceived emotion in real time. In *SpeechEmotion-2000*, pages 19–24, 2000.

[32] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. SpringerVerlag, 2003.

[33] A. El Ali, J. Kildal, and V. Lantz. Fishing or a z?: investigating the effects of error on mimetic and alphabet device-based gesture interaction. In *Proceedings of the 14th*

ACM international conference on Multimodal interaction, ICMI '12, pages 93–100, New York, NY, USA, 2012. ACM.

[34] EXMARaLDA. *http://www.exmaralda.org*, Last Checked: May, 2012.

[35] G. Flammia. *Discourse Segmentation of Spoken Dialogue: An Emperical Approach*. PhD thesis, MIT, 1998.

[36] D. B. Fogel. *Evolutionary computation: toward a new philosophy of machine intelligence*. IEEE Press, Piscataway, NJ, USA, 1995.

[37] L. J. Fogel, A. Owens, and M. Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley, 1966.

[38] M. E. Foster, A. Gaschler, M. Giuliani, A. Isard, M. Pateraki, and R. P. A. Petrick. Two people walk into a bar: Dynamic multi-party social interaction with a robot agent. In *Proceedings of the ACM International Conference on Multimodal Interaction (ICMI 2012)*, pages 3–10, Santa Monica, California, USA, Oct. 2012.

[39] C. Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1-2):199 – 227, 1992.

[40] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 562–569, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[41] D. M. Gavrila. The visual analysis of human movement: a survey. *Comput. Vis. Image Underst.*, 73(1):82–98, 1999.

[42] O. L. Georgeon, A. Mille, T. Bellet, B. Mathern, and F. E. Ritter. Supporting activity modelling from activity traces. *Expert Systems*, 29(3):261–275, 2012.

[43] E. Giachin. Phrase bigrams for continuous speech recognition. *IEEE ICASSP*, 1:225–228, 1995.

[44] S. Gorga and K. Otsuka. Conversation scene analysis based on dynamic bayesian network and image-based gaze detection. In *ICMI-MLMI '10*, pages 54:1–54:8. ACM, 2010.

[45] J. F. Grafsgaard, R. M. Fulton, K. E. Boyer, E. N. Wiebe, and J. C. Lester. Multimodal analysis of the implicit affective channel in computer-mediated textual communication. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 145–152, New York, NY, USA, 2012. ACM.

[46] J. Gratch, N. Wang, J. Gerten, E. Fast, and R. Duffy. Creating rapport with virtual agents. In C. Pelachaud, J.-C. Martin, E. André, G. Chollet, K. Karpouzis, and D. Pelé, editors, *Intelligent Virtual Agents*, volume 4722 of *Lecture Notes in Computer Science*, pages 125–138. Springer Berlin / Heidelberg, 2007.

[47] G. J. Gray, D. J. Murray-Smith, Y. Li, K. C. Sharman, and T. Weinbrenner. Nonlinear model structure identification using genetic programming. *Control Engineering Practice*, 6(11):1341 – 1352, 1998.

[48] A. Groce, D. Peled, and M. Yannakakis. Adaptive model checking. In J.-P. Katoen and P. Stevens, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 2280 of *LNCS*, pages 269–301. Springer Berlin / Heidelberg, 2002.

[49] V. Gudivada and V. Raghavan. Content based image retrieval systems. *Computer*, 28(9):18 –22, sep 1995.

[50] G. Guimarães and A. Ultsch. A method for temporal knowledge conversion. *AIDA*, pages 369–380, 1999.

[51] J. Hagedorn, J. Hailpern, and K. G. Karahalios. Vcode and vdata: illustrating a new framework for supporting the video annotation workflow. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 317–321, New York, NY, USA, 2008. ACM.

[52] M. Hearst. Texttiling: A quantitative approach to discourse segmentation. Technical report, University of California, Berkeley, 1993.

[53] J. H. Holland. *Adaptation in natural and artificial systems.* MIT Press, Cambridge, MA, USA, 1992.

[54] F. Höppner. *Knowledge Discovery from Sequential Data.* PhD thesis, Technischen Universität Braunschweig Genehmigte, 2003.

[55] Z. Huang, V. Eidelman, and M. Harper. Improving a simple bigram hmm part-of-speech tagger by latent annotation and self-training. In *NAACL '09: Proceedings of*

Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pages 213–216, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[56] L. Jiang, A. G. Hauptmann, and G. Xiang. Leveraging high-level and low-level features for multimedia event detection. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 449–458, New York, NY, USA, 2012. ACM.

[57] Y.-G. Jiang, C.-W. Ngo, and S.-F. Chang. Semantic context transfer across heterogeneous sources for domain adaptive video search. In *Proc. ACM MM*, MM '09, pages 155–164, NY, NY, USA, 2009. ACM.

[58] P. Kam and A. Fu. Discovering temporal patterns for interval-based events. *Data Warehousing and Knowledge Discovery*, pages 317–326, 2000.

[59] D. Kaplan. Evaluating and modifying covariance structure models: A review and recommendation. In *Multivariate Behavioral Research*, volume 25, pages 137–155. Psychology Press, 1990.

[60] A. Kendon. *Conducting Interaction: Patterns of Behavior in Focused Encounters*. Cambridge: Cambridge University Press, 1990.

[61] S. Kim, M. Filippone, F. Valente, and A. Vinciarelli. Predicting the conflict level in television political debates: an approach based on crowdsourcing, nonverbal communi-

cation and gaussian processes. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 793–796, New York, NY, USA, 2012. ACM.

[62] Kinect. http://www.xbox.com/kinect, January 2013.

[63] M. Kipp. *(to appear) Multimedia Annotation, Querying and Analysis in ANVIL*, chapter 19. Multimedia Information Extraction. IEEE Computer Society Press.

[64] M. Kipp. Anvil - a generic annotation tool for multimodal dialogue. In *Eurospeech*, 2001.

[65] M. Kipp. *Gesture Generation by Imitation – From human behavior to computer character animation*. PhD thesis, Saarland University, Saarbruecken, Germany, 2003.

[66] M. Kipp. Spatiotemporal coding in anvil. In *LREC*, 2008.

[67] J. Koza, M. Keane, M. Streeter, W. Mydlowec, J. Yu, and G. Lanza. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Springer, 2003.

[68] J. R. Koza. *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA, USA, 1992.

[69] J. R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, 1994.

[70] J. R. Koza, D. Andre, F. H. Bennett, and M. A. Keane. *Genetic Programming III: Darwinian Invention & Problem Solving*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[71] S. Laxman and P. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 04 2006.

[72] S. Laxman, P. Sastry, and K. Unnikrishnan. Discovering frequent episodes and learning hidden markov models: a formal connection. *Knowledge and Data Engineering, IEEE Transactions on*, 17(11):1505 – 1517, nov. 2005.

[73] G.-A. Levow. Prosodic cues to discourse segment boundaries in human-computer dialogue. In M. Strube and C. Sidner, editors, *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 93–96, Cambridge, Massachusetts, USA, April 30 - May 1 2004. Association for Computational Linguistics.

[74] C. Li and G. Biswas. Unsupervised learning with mixed numeric and nominal data. *IEEE Trans. on Knowl. and Data Eng.*, 14(4):673–690, 2002.

[75] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, DMKD '03, pages 2–11, New York, NY, USA, 2003. ACM.

[76] S.-Y. Lin, C.-K. Shie, S.-C. Chen, and Y.-P. Hung. Action recognition for human-marionette interaction. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 39–48, New York, NY, USA, 2012. ACM.

[77] Y.-C. Lin, M.-C. Hu, W.-H. Cheng, Y.-H. Hsieh, and H.-M. Chen. Human action recognition and retrieval using sole depth information. In *Proceedings of the 20th*

ACM international conference on Multimedia, MM '12, pages 1053–1056, New York, NY, USA, 2012. ACM.

[78] D. J. Litman and R. J. Passonneau. Combining multiple knowledge sources for discourse segmentation. In *Proc. of ACL*, pages 108–115, Morristown, NJ, USA, 1995. Association for Computational Linguistics.

[79] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan. Hi, magic closet, tell me what to wear! In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 619–628, New York, NY, USA, 2012. ACM.

[80] V. Y. Lunin and A. G. Urzhumtsev. Improvement of protein phases by coarse model modification. *Acta Crystallographica Section A*, 40(3):269–277, May 1984.

[81] Z. Ma, Y. Yang, Y. Cai, N. Sebe, and A. G. Hauptmann. Knowledge adaptation for ad hoc multimedia event detection with few exemplars. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 469–478, New York, NY, USA, 2012. ACM.

[82] S. Madhvanath, R. Vennelakanti, A. Subramanian, A. Shekhawat, P. Dey, and A. Rajan. Designing multiuser multimodal gestural interactions for the living room. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 61–62, New York, NY, USA, 2012. ACM.

[83] M. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods*, 32:93–110, 2000. 10.3758/BF03200792.

[84] H. Mannila, H. Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, 1997. 10.1023/A:1009748302351.

[85] S. Martin, J. Liermann, and H. Ney. Algorithms for bigram and trigram word clustering. *Sp. Comm.*, 1998.

[86] G. McKeown, M. Valstar, R. Cowie, and M. Pantic. The semaine corpus of emotionally coloured character interactions. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 1079 –1084, july 2010.

[87] D. McNeill. *Hand and Mind: What gestures reveal about thought.* Chicago: U. of Chicago Press, 1992.

[88] D. McNeill. Gesture, gaze, and ground. In S. Renals and S. Bengio, editors, *Machine Learning for Multimodal Interaction*, volume 3869 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin / Heidelberg, 2006.

[89] D. McNeill, S. Duncan, A. Franklin, J. Goss, I. Kimbara, F. Parrill, H. Welji, L. Chen, M. Harper, F. Quek, T. Rose, and R. Tuttle. Mind-merging. In *Expressing oneself / expressing one's self: Communication, language, cognition, and identity*, 2007.

[90] R. Mihalcea and M. Burzo. Towards multimodal deception detection – step 1: building a collection of deceptive videos. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 189–192, New York, NY, USA, 2012. ACM.

[91] C. Miller, L.-P. Morency, and F. Quek. Structural and temporal inference search (stis): pattern identification in multimodal data. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 101–108, New York, NY, USA, 2012. ACM.

[92] C. Miller and F. Quek. Toward multimodal situated analysis. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pages 239–246, New York, NY, USA, 2011. ACM.

[93] C. Miller and F. Quek. Interactive data-driven discovery of temporal behavior models from events in media streams. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 459–468, New York, NY, USA, 2012. ACM.

[94] C. Miller, A. Robinson, R. Wang, P. Chung, and F. Quek. Interaction techniques for the analysis of complex data on high-resolution displays. In *ICMI '08: Proceedings of the 10th international conference on Multimodal interfaces*, pages 21–28, New York, NY, USA, 2008. ACM.

[95] C. A. Miller, F. Quek, and N. Ramakrishnan. Structuring ordered nominal data for event sequence discovery. In *Proceedings of the international conference on Multimedia*, MM '10, pages 1075–1078, New York, NY, USA, 2010. ACM.

[96] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126, 2006.

[97] C. H. Mooney and J. F. Roddick. Mining relationships between interacting episodes. In *Proceedings of the 4th SIAM International Conference on Data Mining (SDM'04). SIAM*, 2004.

[98] F. Mörchen. Algorithms for time series knowledge mining. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 668–673, New York, NY, USA, 2006. ACM.

[99] F. Mörchen. *Time Series Knowledge Mining*. PhD thesis, Philips-University Marburg, Germany, 2006.

[100] F. Mörchen. Unsupervised pattern mining from symbolic temporal data. *SIGKDD Explor. Newsl.*, 9(1):41–55, 2007.

[101] F. Mörchen and D. Fradkin. Robust mining of time intervals with semi-interval partial order patterns. In *SIAM Conference on Data Mining (SDM)*, 2010.

[102] L.-P. Morency, C. M. Christoudias, and T. Darrell. Recognizing gaze aversion gestures in embodied conversational discourse. In *ICMI '06*, pages 287–294.

[103] L.-P. Morency, I. de Kok, and J. Gratch. Context-based recognition during human interactions: automatic feature selection and encoding dictionary. In *ICMI '08: Proceedings of the 10th international conference on Multimodal interfaces*, pages 181–188, New York, NY, USA, 2008. ACM.

[104] T. Nagamatsu, R. Sugano, Y. Iwamoto, J. Kamahara, and N. Tanaka. User-calibration-free gaze tracking with estimation of the horizontal angles between the visual and the optical axes of both eyes. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pages 251–254, New York, NY, USA, 2010. ACM.

[105] K. Narendra and J. Balakrishnan. Adaptive control using multiple models. *Automatic Control, IEEE Transactions on*, 42(2):171 –187, feb. 1997.

[106] K. Narendra, J. Balakrishnan, and M. Ciliz. Adaptation and learning using multiple models, switching, and tuning. *Control Systems Magazine, IEEE*, 15(3):37 –51, jun. 1995.

[107] F. Oliveira, H. Cowan, B. Fang, and F. Quek. Fun to develop embodied skill: how games help the blind to understand pointing. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '10, pages 16:1–16:8, New York, NY, USA, 2010. ACM.

[108] M. T. Orne. On the social psychology of the psychological experiment: With particular reference to demand characteristics and their implications. *American Psychologist*, 17(11):776–783, Nov 1962.

[109] S. Ortmanns, H. Ney, and A. Eiden. Language-model look-ahead for large vocabulary speech recognition. *Intl. Conf. on Spoken Language Processing*, 1996.

[110] S. Park, J. Gratch, and L.-P. Morency. I already know your answer: using nonverbal behaviors to predict immediate outcomes in a dyadic negotiation. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 19–22, New York, NY, USA, 2012. ACM.

[111] S. Park, G. Mohammadi, R. Artstein, and L.-P. Morency. Crowdsourcing micro-level multimedia annotations: the challenges of evaluation and interface. In *Proceedings of the ACM multimedia 2012 workshop on Crowdsourcing for multimedia*, CrowdMM '12, pages 29–34, New York, NY, USA, 2012. ACM.

[112] R. J. Passoneau and D. J. Litman. Empirical analysis of three dimensions of spoken discourse: Segmentation, coherence and linguistic devices, 1996.

[113] R. J. Passonneau and D. J. Litman. Discourse segmentation by human and automated means. *Comput. Linguist.*, 23:103–139, March 1997.

[114] D. Patnaik, P. Butler, N. Ramakrishnan, L. Parida, B. J. Keller, and D. A. Hanauer. Experiences with mining temporal event sequences from electronic medical records: initial successes and some challenges. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 360–368, New York, NY, USA, 2011. ACM.

[115] D. Patnaik, P. S. Sastry, and K. P. Unnikrishnan. Inferring neuronal network connectivity from spike data: A temporal data mining approach. *Scientific Programming*, 16(1):49–77, January 2007.

[116] P. Pirolli and S. K. Card. Information foraging. *Psychological Review*, 106:643–675, 1999.

[117] Pymunk. *http://code.google.com/p/pymunk/*, Last Checked: Feb., 2011.

[118] Qt. *http://qt.nokia.com/*, Last Checked: Mar., 2012.

[119] F. Quek. The catchment feature model: a device for multimodal fusion and a bridge between signal and sense. *EURASIP J. Appl. Signal Process.*, 2004:1619–1636, Jan. 2004.

[120] F. Quek, R. Bryll, D. McNeill, and M. Harper. Gestural origo and loci-transitions in natural discourse segmentation. In *IEEE Workshop on Cures in Communication*, 2001.

[121] F. Quek, R. Ehrich, and T. Lockhart. As go the feet...: on the estimation of attentional focus from stance. In *Proceedings of the 10th international conference on Multimodal interfaces*, ICMI '08, pages 97–104, New York, NY, USA, 2008. ACM.

[122] F. Quek, D. McNeill, R. Bryll, and M. Harper. Gestural spatialization in natural discourse segmentation. In *Seventh International Conference on Spoken Language Processing*, 2002.

[123] F. Quek, D. McNeill, R. Bryll, C. Kirbas, H. Arslan, K. McCullough, N. Furuyama, and R. Ansari. Gesture, speech, and gaze cues for discourse segmentation. In *CVPR, 2000.*, volume 2, pages 247–254 vol.2, 2000.

[124] F. Quek, T. Rose, and D. McNeill. Multimodal meeting analysis. In *IA*, 2005.

[125] R. T. Rose. Macvissta: A system for multimodal analysis of human communication and interaction. Master's thesis, Virginia Tech, 2007.

[126] R. T. Rose, F. Quek, and Y. Shi. Macvissta: a system for multimodal analysis. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, pages 259–264, New York, NY, USA, 2004. ACM.

[127] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *TCSVT*, 8(5):644–655, Sep 1998.

[128] S. Russell and P. Norvig. *Artificial Intelligence A Modern Approach*. Prentice Hall, 2003.

[129] H. Sacks. *Lectures on Conversation*. Oxford: Blackwell, 1992.

[130] P. S. Sastry and K. P. Unnikrishnan. Conditional probability based significance tests for sequential patterns in multi-neuronal spike trains. 2008.

[131] T. Schmidt. The transcription system exmaralda: An application of the annotation graph formalism as the basis of a database of multilingual spoken discourse. In *Proceedings of the IRCS Workshop On Linguistic Databases, 11-13 December 2001*, pages 219–227, Philadelphia, 2001. Institute for Research in Cognitive Science, University of Pennsylvania. EN.

[132] T. Schmidt, S. Duncan, O. Ehmer, J. Hoyt, M. Kipp, D. Loehr, M. Magnusson, T. Rose, and H. Sloetjes. An exchange format for multimodal annotations. chapter An exchange format for multimodal annotations, pages 207–221. Springer-Verlag, Berlin, Heidelberg, 2009.

[133] T. Schmidt and K. Wörner. Exmaralda – creating, analysing and sharing spoken language corpora for pragmatic research. *Pragmatics*, 19, 2009.

[134] K. Schoeffmann, M. Taschwer, and L. Boeszoermenyi. The video explorer: a tool for navigation and searching within a single video based on fast content analysis. In *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, MMSys '10, pages 247–258, New York, NY, USA, 2010. ACM.

[135] E. Schwalb and L. Vila. Temporal constraints: A survey. *Constraints*, 3(2/3):129–149, 1998.

[136] X. Shen and J. Ye. Adaptive model selection. *JASA*, 97(457):pp. 210–221, 2002.

[137] Z. Shen, S. Arslan Ay, S. H. Kim, and R. Zimmermann. Automatic tag generation and ranking for sensor-rich outdoor videos. In *MM ' 11*, MM '11, pages 93–102, New York, NY, USA, 2011. ACM.

[138] M. Siu and M. Ostendorf. Variable n-grams and extensions for conversational speech language modeling. *Speech and Audio Processing, IEEE Transactions on*, 8(1):63 –75, Jan. 2000.

[139] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(12):1349 –1380, dec 2000.

[140] Y. Song, L.-P. Morency, and R. Davis. Multimodal human behavior analysis: learning correlation and interaction across modalities. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 27–30, New York, NY, USA, 2012. ACM.

[141] I. Spiro, G. Taylor, G. Williams, and C. Bregler. Hands by hand: Crowd-sourced motion tracking for gesture annotation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 17 –24, 2010.

[142] A. Stolcke and S. M. Omohundro. Inducing probabilistic grammars by bayesian model merging. In *ICGI '94: Proceedings of the Second International Colloquium on Grammatical Inference and Applications*, pages 106–118, London, UK, 1994. Springer-Verlag.

[143] X. Sun and T. H. Applebaum. Intonational phrase break prediction using decision tree and n-gram model. In *Proc. of Eurospeech2001*, pages 3–7, 2001.

[144] TDMiner. http://people.cs.vt.edu/patnaik/software, March 2013.

[145] H. L. Thanh, G. Abeysinghe, and C. Huyck. Automated discourse segmentation by syntactic information and cue phrases. In *Proceedings of Artificial Intelligence and Applications*, 2004.

[146] Theme. http://www.noldus.com/human-behavior-research/products/theme, April 2012.

[147] A. Ultsch. Unification-based temporal grammar. Technical Report 37, Philips-University Marburg, Germany, 2004.

[148] A. Utsumi and J. Ohya. Multiple-hand-gesture tracking using multiple cameras. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 2 vol. (xxiii+637+663), 1999.

[149] O. Vinyals, D. Bohus, and R. Caruana. Learning speaker, addressee and overlap detection models from multimodal streams. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 417–424, New York, NY, USA, 2012. ACM.

[150] V. V. Vishnevskiy and D. P. Vetrov. The algorithm for detection of fuzzy behavioral patterns. In *Measuring behavior*, pages 166–170, 2010.

[151] M. Voit and R. Stiefelhagen. 3d user-perspective, voxel-based estimation of visual focus of attention in dynamic meeting scenarios. In *ICMI-MLMI '10*, pages 51:1–51:8. ACM, 2010.

[152] T. D. Wang, K. Wongsuphasawat, C. Plaisant, and B. Shneiderman. Exploratory search over temporal event sequences: Novel requirements, operations, and a process model. *3rd Workshop on Human-Computer Information Retrieval*, 2009.

[153] U. Westermann and R. Jain. Toward a common event model for multimedia applications. *IEEE MultiMedia*, 14:19–29, January 2007.

[154] P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. Elan: a professional framework for multimodality research. In *In Proceedings of Language Resources and Evaluation Conference (LREC)*, 2006.

[155] R. Yang, S. Sarkar, B. Loeding, and A. Karshmer. Efficient generation of large amounts of training data for sign language recognition: A semi-automatic tool. In K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, editors, *Computers Helping People with Special Needs*, volume 4061 of *Lecture Notes in Computer Science*, pages 635–642. Springer Berlin / Heidelberg, 2006.

[156] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.

[157] Y. Yin. A hierarchical approach to continuous gesture analysis for natural multi-modal interaction. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 357–360, New York, NY, USA, 2012. ACM.

[158] L. Zadeh. Soft computing and fuzzy logic. *Software, IEEE*, 11(6):48–56, Nov 1994.

[159] L. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, May 1996.

[160] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pages 279–286, New York, NY, USA, 2011. ACM.

[161] T. Zhang, C. Xu, G. Zhu, S. Liu, and H. Lu. A generic framework for event detection in various video domains. In *Proceedings of the international conference on Multimedia*, MM '10, pages 103–112, New York, NY, USA, 2010. ACM.