ABSTRACT

KALIA, ANUP KUMAR. Güven: Estimating Trust from Agents’ Interactions. (Under the direction of Munindar P. Singh.)

How an agent trusts another naturally depends on the outcomes of their interactions? Previous approaches have treated the outcomes in a domain-specific way. We propose Güven, an approach that relates trust to the domain-independent notion of commitments. In particular, Güven incorporates a probabilistic model for trust based on commitment outcomes and show how to train its parameters for each subject based on the subject’s assessments.

To evaluate Güven, first, we provide an approach to extract commitments from interactions. We validate the approach on real-world datasets such as emails and chat messages. The approach yields a high F-measure of 90% for emails (Enron email corpus) and 80% for chat messages (HP IT support chat dataset). Second, we conduct an empirical study where we ask subjects to read emails extracted from the Enron dataset (and augmented with some synthetic emails for completeness) and estimate trust between each pair of communicating participants from the range of 0 to 1. For the evaluation, we consider four models: (1) a baseline that considers fixed trust parameters for all subjects, (2) Güven that learns trust parameters of subjects from their trust assessments, and (3) and (4) the variations of Güven that consider discount window sizes and commitment strengths respectively. Comparing the models, we find that Güven performs better than the baseline model and the variations of Güven perform better than Güven.

To improve the trust estimation using Güven, we hypothesize that there are other factors, apart from commitments, that influence trust. These factors are: (1) outcomes of an agent’s goals, (2) an agent’s moods, and (3) emotions expressed toward an agent. To verify these factors, we conduct an empirical evaluation where we ask subjects to play a variant of the Colored Trails game. We use the data collected to create different static and dynamic Bayesian models that capture the relationships between trust, goals, moods, commitments, and emotions. We train and test these models to predict accuracy results. From the results, we find that the model that consider an agent’s past trust influencing its current trust has higher prediction accuracy than the models that consider its past moods, outcomes of commitments, and emotions expressed toward it, respectively.

Next, we plan to extend Güven to estimate team performance. We hypothesize that team performance can be measured by three different properties: team hierarchy, cohesion, and trust. Therefore, we plan to create approaches to extract these properties from agents’ interactions to estimate team performance.

Our main contribution is to launch a research program into computing trust based on a semantically well-founded account of interpersonal interactions.
Güven: Estimating Trust from Agents’ Interactions

by
Anup Kumar Kalia

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2014

APPROVED BY:

James C. Lester
Jessica K. Jameson

R. Michael Young
Munindar P. Singh
Chair of Advisory Committee
# TABLE OF CONTENTS

## LIST OF TABLES

- v

## LIST OF FIGURES

- vi

## Chapter 1 Introduction

- 1
  - 1.0.1 Challenges
  - 1.0.2 Proposed Approach
  - 1.1 Research Objectives
  - 1.2 Task
  - 1.3 Proposed Plan
  - 1.4 Contributions
  - 1.5 Organization

## Chapter 2 Extraction of Commitments

- 7
  - 2.1 Introduction
  - 2.2 Commitments
  - 2.3 Running Example
  - 2.4 Understanding Commitments in the Context of Agents’ Interactions
    - 2.4.1 Task
    - 2.4.2 Commitment Creation
    - 2.4.3 Commitment Discharge
    - 2.4.4 Subcontracting a Commitment
    - 2.4.5 Commitment Cancellation
  - 2.5 Approach to Identify and Monitor Commitments in Agents’ Interactions
    - 2.5.1 Typed Dependency
    - 2.5.2 Identifying interactions and Sentences
    - 2.5.3 Extracting Features
    - 2.5.4 Identifying Tasks
    - 2.5.5 Identifying Creations
    - 2.5.6 Identifying Subcontracts
    - 2.5.7 Identifying Discharges
    - 2.5.8 Identifying Cancellations
  - 2.6 Evaluation and Prototype
    - 2.6.1 Data
    - 2.6.2 Labeling Data
    - 2.6.3 Results
    - 2.6.4 Prototype Tool
  - 2.7 Related Work
  - 2.8 Conclusions and Future Work

## Chapter 3 Güven: Estimating Trust From Commitments

- 26
  - 3.1 Introduction
  - 3.2 Background on Commitments
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Timeline for different task completions.</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Sample interaction in an enterprise setting.</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>Tasks identified from the interactions of Table 2.1</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Commitment creation identified from the interactions in Table 2.1</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>Discharge commitment identified from the interactions in Table 2.1</td>
<td>11</td>
</tr>
<tr>
<td>2.5</td>
<td>Subcontract commitments identified from message interactions in Table 2.1</td>
<td>12</td>
</tr>
<tr>
<td>2.6</td>
<td>Cancel commitment identified from message interactions in Table 2.1</td>
<td>12</td>
</tr>
<tr>
<td>2.7</td>
<td>Results for emails, showing precision (P), recall (R), and F-measure (F) for different classifiers.</td>
<td>19</td>
</tr>
<tr>
<td>2.8</td>
<td>Results for chats, showing precision (P), recall (R), and F-measure (F) for different classifiers.</td>
<td>19</td>
</tr>
<tr>
<td>2.9</td>
<td>Distribution of commitment operations in email sentences.</td>
<td>20</td>
</tr>
<tr>
<td>2.10</td>
<td>Evaluation on independent test datasets using SVM for email and LR for chat, respectively.</td>
<td>21</td>
</tr>
<tr>
<td>2.11</td>
<td>Sample extracted from the Enron corpus.</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>Examples of email interactions.</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>Different features and trust values from different subjects.</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Statistical test results for $H_1$. ($\mu$-mean of the MAEs)</td>
<td>34</td>
</tr>
<tr>
<td>3.4</td>
<td>Statistical test results for $H_2$.</td>
<td>35</td>
</tr>
<tr>
<td>3.5</td>
<td>Ordering among the feature values obtained from subjects' assessments.</td>
<td>36</td>
</tr>
<tr>
<td>3.6</td>
<td>Statistical test results for $H_3$.</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>Distributions of values of variables in the data.</td>
<td>45</td>
</tr>
<tr>
<td>4.2</td>
<td>Correlations between variables observed in one time slice. (1 in superscript indicates Round 1)</td>
<td>47</td>
</tr>
<tr>
<td>4.3</td>
<td>Correlating variables observed in time slice $t - 1$ with variables observed in time slice $t$. (1 and 2 in superscript indicate Round 1 and 2 respectively)</td>
<td>47</td>
</tr>
<tr>
<td>5.1</td>
<td>Examples of communication patterns from military chat data.</td>
<td>54</td>
</tr>
<tr>
<td>5.2</td>
<td>Ranks of participants selected from the chat rooms.</td>
<td>60</td>
</tr>
<tr>
<td>5.3</td>
<td>Statistically comparing semantic features with social features (sem-semantic, comm-communication, soc-social, avg-average, resp-response, del-delay, msg-messages, hyp-hypotheses, rej-rejected).</td>
<td>64</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Trust updates based on a commitment progression.</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Monitoring commitments in interactions.</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>The lifecycle of a commitment [81].</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Process followed to identify commitments.</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Typed dependencies derived from a sentence “I will inspect your car for damage” in Table 2.1.</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>Steps to identify task from an email sentence “I will inspect your car for damage.”</td>
<td>16</td>
</tr>
<tr>
<td>2.6</td>
<td>Steps to identify a commitment from an email sentence “I will inspect your car for damage.”</td>
<td>17</td>
</tr>
<tr>
<td>2.7</td>
<td>Steps to identify a subcontract.</td>
<td>17</td>
</tr>
<tr>
<td>2.8</td>
<td>Steps to identify a discharge.</td>
<td>17</td>
</tr>
<tr>
<td>2.9</td>
<td>Steps to identify a cancel.</td>
<td>18</td>
</tr>
<tr>
<td>2.10</td>
<td>The architecture of our commitment identification and monitoring tool.</td>
<td>22</td>
</tr>
<tr>
<td>3.1</td>
<td>Commitment lifecycle [81].</td>
<td>27</td>
</tr>
<tr>
<td>3.2</td>
<td>Process for evaluation.</td>
<td>29</td>
</tr>
<tr>
<td>3.3</td>
<td>MAE for predicting trust values.</td>
<td>34</td>
</tr>
<tr>
<td>3.4</td>
<td>Results comparing different window sizes to predict trust values.</td>
<td>35</td>
</tr>
<tr>
<td>3.5</td>
<td>Incorporating commitment weights reduces MAE.</td>
<td>36</td>
</tr>
<tr>
<td>4.1</td>
<td>Two possible enactments in a multiagent scenario.</td>
<td>40</td>
</tr>
<tr>
<td>4.2</td>
<td>A screenshot of the Colored Trails game.</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Timeline of our data acquisition.</td>
<td>45</td>
</tr>
<tr>
<td>4.4</td>
<td>Negated LL, AIC, and BIC scores for Bayesian models (*p&lt;0.05).</td>
<td>49</td>
</tr>
<tr>
<td>5.1</td>
<td>Process followed to compute ranks with respect to directive, question, and informative patterns, and directives and questions combined for participants (P*) from broadcast messages.</td>
<td>56</td>
</tr>
<tr>
<td>5.2</td>
<td>Parse tree derived from “Cos, send all reports up to BN over this net” where Cos is the Chief of Staff position and BN is the Battalion.</td>
<td>57</td>
</tr>
<tr>
<td>5.3</td>
<td>An example of a single link dendrogram with distance d between clusters, applied to estimate rank R (bottom row).</td>
<td>59</td>
</tr>
<tr>
<td>5.5</td>
<td>Percentage accuracy of computing hierarchy using different response patterns directive (Dir), question (Ques), informative (Info), and directive+informative (Dir+Info) via different datasets.</td>
<td>63</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In multiagent systems, trust is one of the most important factors for agents to continue interacting with each other. In such systems, a truster estimates (and continually revises) its trust for a trustee based on their mutual interactions. Trust between them determines the length of their engagements. Understanding such interactions and estimating trust is an interesting and challenging problem. To address the problem, we propose Güven \(^1\), a computational model of trust based on commitments. We evaluate our approach using real-world datasets such as emails and chat messages. We envision that using our approach, users can create different applications such as (1) supporting agents to take important decisions in organizational settings, (2) predicting trustworthiness of agents’ reviews for products and services, and (3) estimating team cohesion or performance.

We briefly describe what does trust mean? According to Gambetta [31] the most important intuition behind trust is that a truster is vulnerable to decisions of a trustee. Further, Gambetta describes that trust indicates a level of subjective probability with which a truster assesses a trustee for performing a particular task. If the trustee’s performance is beneficial to the truster, the truster continues engaging with the trustee. Similarly, Mayer et al. [54] define trust as the willingness of a truster to be vulnerable to a trustee for the completion of a task. The primary motivation behind Mayer et al.’s work is to clarify the confusion between trust and risk. Mayer et al. specify that a truster doesn’t risk anything to trust a trustee. Instead, a truster’s shows willingness to assume a risk. For example, a manager may risk by allowing its employee to perform an important task. If the employee performs, the manager is willing to take further risks with the employee. Rousseau et al. [65] define trust as a psychological state of a trustee comprising of intention to accept vulnerability and mention that trust depends on two necessary conditions. First, is risk where a truster builds an opportunity of trust by taking risks. Second, is interdependence which means that a truster cannot achieve its objectives without relying

\(^1\)From the Turkish word that brings together the concepts of trust and reliance.
upon a trustee. Castelfranchi and Falcone [11] describe that a truster trusts a trustee when the truster has goals to achieve, and believes that the trustee can achieve its goals. A truster’s belief is with respect to how competent and reliable the trustee is, and whether the trustee can perform and achieve the delegated goals.

1.0.1 Challenges

We describe related works that provide approaches to estimate trust from agents’ interactions. Scissors et al. [71] explore the linguistic similarity in chat messages to estimate trust between message senders and receivers. Adalı et al. [1] calculate the relationship strength between two users in Twitter based on social and behavioral aspects such as the number of friends and followers they have, number of messages exchanged between them, and the time delay between the messages exchanged. DuBois et al. [23] provide an algorithm to compute trust and distrust in a social network. Previous approaches to estimate trust are limited to numerical heuristics and are criticized for missing the essential intuitive considerations of trust, e.g., the autonomy of the participants and the vulnerability of the truster to decisions by the trustee [11, 31, 54, 65]. Again, the richer approaches, however, have not lent themselves well to computational techniques that could be applied in practice.

1.0.2 Proposed Approach

We seek to bridge the above gap between theory and practice. Specifically, we propose G"uven, a computational model of trust founded on commitments that supports how people determine trust in others based on their interactions. Commitments are important for trust because they characterize the outcomes of interactions in high-level terms and can be identified from agents’ interactions. We limit G"uven to commitments, although, it can be extended to related concepts such as prohibitions and authorizations.

A commitment $C(\text{debtor, creditor, antecedent, consequent})$ means that the debtor commits to bringing about the consequent for the creditor provided the antecedent holds. For example, $C(\text{Buck, Selia, deliver, pay})$ means that Buck (buyer) commits to Selia (seller) to paying a specified amount provided Selia delivers the goods. When Selia delivers, the commitment is detached. When Buck pays, the commitment is discharged or satisfied. If Selia delivers but Buck does not pay, the commitment is violated. In essence, a commitment describes a social relationship between two persons giving a high-level description of what one expects of the other. As a result, it is natural that commitments (and their satisfaction or violation) can be used as a basis for trust. In the above example, if Buck discharges the commitment, it brings a positive experience to Selia and Selia’s trust for Buck may increase; if Buck violates the commitment, it brings a negative experience to Selia and Selia’s trust for Buck may decrease.
Figure 4.1 shows the intuition graphically.

![Diagram showing the intuition](image)

**Figure 1.1:** Trust updates based on a commitment progression.

Despite the apparent match, few approaches relate trust and commitments. Singh [76] and Chopra et al. [14] relate trust and commitments in terms of logical postulates, from an architectural perspective. In contrast, we understand trust and commitment in probabilistic terms, considering the outcomes of specific commitments and their effect on the trust relationships between the concerned parties.

### 1.1 Research Objectives

The primary research objective ($R_1$) of this dissertation is to create Güven and evaluate it on real-world datasets. The objective splits into three subobjectives.

**$R_{1a}$**. The first subobjective is to build the Güven model and evaluate it on real-world datasets such as emails. To build the model we will adopt Singh’s [75] commitment model and Wang and Singh’s [85] trust model. To evaluate the model, first, we will extract commitments from emails using Kalia et al.’s [41] trained classifier. Second, we will collect subjects’ trust assessments for emails. Once the data is collected, we plan to consider four models for the evaluation: (1) a baseline model that considers fixed trust parameters for all subjects, (2) the proposed model Güven that considers learned trust parameters for each subject, and (3) and (4) two different variations of Güven based on discount window sizes and commitment strengths respectively. Based on the proposed models, first, we will check if Güven estimates trust better than the baseline approach. Then, we will check if the variations improve over the results obtained from Güven. From this objective we will create Güven that estimates trust from agents’ interactions via commitments.
The second subobjective is to improve the Güven model by looking for factors, apart from commitments, that can be included into Güven. For a start, we plan to consider three factors that may influence an agent’s trust: (1) outcomes of an agent’s goals, (2) an agent’s moods, and (3) emotions expressed toward an agent. To evaluate these factors, we will perform an empirical evaluation with subjects where we will ask them to play a variant of Gal et al.’s [30] Colored Trails game. Our variant provides a chat interface, through which subjects can negotiate and exchange tiles and express emotions toward opponents. During the game subjects will record their interactions with their opponents and fill a feedback form with their trust and mood before and after each round in a game. From their interactions, we will manually identify commitments and emotions expressed by subjects. We will used the data collected to create different static and dynamic Bayesian models that capture relationships between trust, moods, goals, and emotions. We will train and test them to predict accuracy results. From, the results we will find factors that influence trust the most. From this objective, we plan to improve the Güven model. Further, we plan to evaluate the improved model on a different dataset collected from Intelligence Advanced Research Projects Activity (IARPA). The dataset is posted as a part of challenge on a third-party website Innocentive. The dataset is created based on the checkmate game [7] where subjects participated and played against each other to assess their trustworthiness for each other.

The third subobjective is to use the Güven model to estimate team performance. To achieve the subobjective, first, we will create approaches to extract properties from agents’ interactions such as team hierarchy, trust, and cohesion that influence team performance. Then, we will create and evaluate methods to verify how these properties influence team performance. From this objective, we want to increase the usability of Güven model from estimating inter-personal trust to estimating team performance.

1.2 Task

Below, we describe the tasks based on the research objectives described above.

T1. In the first task, we will create and evaluate techniques built using natural language processing and machine learning to identify the lifecycle of commitments, i.e., how commitments are created, delegated, canceled, and discharged. We will evaluate techniques using two datasets, one, using the Enron email corpus [26, 47], and two, using the HP’s IT incident management logs.

T2. In the second task, first, we will build and evaluate Güven, a computational model of trust based on commitments. To evaluate the model, we will conduct an empirical evaluation
on emails where we will ask subjects to read emails selected from the Enron corpus, and estimate a trust value ranging from 0 to 1 between the senders and receivers of emails. Using the data from subjects we will evaluate Güven with different approaches. Second, we will evaluate what other factors, apart from commitments, influence trust. For the evaluation, we will conduct an empirical study with subjects where we will ask them to play a variant of Gal et al.’s [30] Colored Trails game. From the data collected we will create different static and dynamic Bayesian models comprising of commitments, goals, trust, moods, and emotions nodes. We will evaluate models based on prediction accuracy to choose the best possible models for estimating trust. Third, we will further evaluate improved Güven based on a different dataset collected from IARPA.

\( T_3 \). In the third task, we will create techniques to extract different team properties from agents’ interactions such as team hierarchy, trust, and cohesion. We will evaluate our techniques using a military exercise dataset and the Enron email corpus. Further, we will create methods to estimate team performance based on team properties extracted.

1.3 Proposed Plan

I plan to complete all the proposed tasks and defend my dissertation before November 2015. Table 1.1 outlines the timelines of different task completions.

Table 1.1: Timeline for different task completions.

<table>
<thead>
<tr>
<th>Task</th>
<th>Research Objectives</th>
<th>Status</th>
<th>Duration</th>
<th>Research Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>T₁</td>
<td>R₁a</td>
<td>Complete</td>
<td></td>
<td>IEEE SCC, 2013 [42]</td>
</tr>
<tr>
<td>T₂</td>
<td>R₁a &amp; R₁b</td>
<td>Partially Complete</td>
<td>Oct 2014–Nov 2014</td>
<td>ECAI, 2014 (Short) [43], AAAI, 2015 (Submitted)[39]</td>
</tr>
<tr>
<td>T₃</td>
<td>R₁c</td>
<td>Partially Complete</td>
<td>Dec 2014–Sept 2015</td>
<td>SocInfo, 2014 [40]</td>
</tr>
</tbody>
</table>

1.4 Contributions

The primary contribution of this dissertation is to create Güven, a model that estimates trust from agents’ interactions. We create this model by considering the intuition that commitment outcomes influence trust. To evaluate the model, first, we extract commitments created, delegated, violated, and discharged from emails using natural language processing and machine
learning techniques. Second, we conduct an empirical study where we ask subjects to read a set of emails and provide trust in the range of 0 to 1 between the senders and receivers of emails. We evaluate Güven against a baseline and variants of Güven. To improve the trust estimation, we hypothesize that there would be other factors that might influence trust apart from outcomes of commitments. These factors are: an agent’s moods, emotions expressed toward an agent, and outcomes of an agent’s goals. To verify our hypothesis, first, we create different Bayesian models that capture the relationships between trust, commitments, goals, moods, and emotions. Then, we conduct an empirical evaluation where we ask subjects to play a variant of Colored Trails game. From the evaluation we collect data and evaluate different Bayesian models in terms of prediction accuracy. Finally, we plan to extend Güven to estimate team performance. To estimate team performance we plan to compute different team properties such as team hierarchy, trust, and cohesion. We envision that using Güven users can build applications that can determine trust between people in an organization from their communications such as emails and chat messages. Also, it can be extended to study team dynamics and performance of team members in an organization.

1.5 Organization

Chapter 1 introduces the problem, challenges, research objectives, tasks, and overall contributions of this dissertation. Chapter 2 illustrates the process of extraction of commitments from emails and chat messages. Chapter 3 proposes and evaluates Güven, a trust model based on commitments. Chapter 4 describes the relationships between trust, goals, commitments, emotions, and moods. Chapter 5 describes process to estimate team hierarchy from broadcast messages.
Chapter 2

Extraction of Commitments

2.1 Introduction

In this work we describe an approach to capture commitments from agents’ interactions. Figure 4.1 describes our overall approach. Our tool sits above the interaction infrastructure, such as email and chat, and monitors agents’ interactions. It determines the progression of a commitment from those interactions and displays its changing state to users to help them carry out a service engagement. The user can choose to accept or reject its suggestions.

![Diagram](image)

Figure 2.1: Monitoring commitments in interactions.

This paper makes the following contributions.

- Define commitments and their lifecycle in the context of agents’ interactions.
• Provide and evaluate an approach involving natural language processing and machine learning to identify commitment creation, delegation, cancellation, and discharge.

• Develop a tool that monitors email and chat interactions, identifies the creation and progression in a commitment, and nonintrusively presents them to users. The users can choose to accept or reject its suggestions.

We have experimentally validated our approach on real-world email and chat datasets. Our approach yields better accuracy than existing work [48, 70] for commitment identification, and performs well on identifying commitment delegation, cancellation, and discharge, which others have not studied.

The paper is structured as follows. Section 2.2 presents the background on commitments. Section 2.3 provides a motivating scenario. Section 2.4 defines tasks and commitments in the context of interactions in people-driven service engagements. Section 2.5 describes our approach to identifying commitments from email and chat interactions. Section 5.4 explains our dataset, experimentation, and evaluation results. Section 2.7 discusses related work. Section 2.8 discusses future directions in the current work.

2.2 Commitments

We adopt Singh’s model of commitments [75] to capture business relationships between any two autonomous entities. Specifically, commitments express business meanings underlying the interactions between these entities. A commitment here is a conditional business relationship directed from a debtor to a creditor, and can be formalized as \( C(\text{DEBTOR}, \text{CREDITOR}, \text{antecedent}, \text{consequent}) \).

The above formula shows that the debtor is committed to bringing about the consequent for the creditor provided the antecedent holds. When a debtor sends an offer to a creditor, a commitment is created and becomes active. When the antecedent is brought about, (including if it is initially true) the commitment is detached. When the consequent holds, the commitment is satisfied. If the antecedent holds and the consequent times out, the commitment is violated. If the antecedent is True, the commitment is unconditional.

Telang and Singh [81] present the commitment lifecycle shown in Figure 3.1. According to Figure 3.1, a commitment transitions from one state to another due to the following operations: create, detach (antecedent holds), discharge (consequent holds), cancel, and delegate.

• \( \text{create}(c) \) forms a commitment. A commitment \( c \) is created when a debtor voluntarily offers to do a task or when the debtor is directed to do a task by a superior.
Figure 2.2: The lifecycle of a commitment [81].

- $\text{detach}(c)$ detaches a commitment. A commitment is detached if its antecedent present for a commitment becomes true.

- $\text{discharge}(c)$ completes a commitment when a debtor executes a committed task.

- $\text{cancel}(c)$ terminates the commitment $c$. A commitment can be canceled only by its debtor.

- $\text{delegate}(c, z)$ replaces $z$ as the $c$’s debtor. The debtor of the commitment $c$ is replaced by $z$ when the original debtor delegates the commitment to $z$.

2.3 Running Example

Table 2.1 provides an insurance scenario where an Insurer (AGFIL, an insurance company) commits to inspecting its customer’s (John Doe) car damage. AGFIL delegates the estimate verification to a company Lee Consulting Services (LCS). LCS hires a mechanic (M) and requests M to do the inspection. In the first case, M denies and, therefore, LCS hires another mechanic M1 that does the job.

2.4 Understanding Commitments in the Context of Agents’ Interactions

We view agents’ interactions in terms of tasks and commitments derived from the synthetic interactions in Table 2.1. We apply the steps below

- Identify if a message contains a task or an event.
Table 2.1: Sample interaction in an enterprise setting.

<table>
<thead>
<tr>
<th>S</th>
<th>R</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGFIL</td>
<td>John</td>
<td>I will inspect your car for damage</td>
</tr>
<tr>
<td>AGFIL</td>
<td>LCS</td>
<td>Can you please inspect the car for damage?</td>
</tr>
<tr>
<td>LCS</td>
<td>M</td>
<td>Please inspect the car for damage</td>
</tr>
<tr>
<td>M</td>
<td>LCS</td>
<td>I cannot inspect it as I am busy with other work</td>
</tr>
<tr>
<td>LCS</td>
<td>M1</td>
<td>Please inspect the car for damage</td>
</tr>
<tr>
<td>M1</td>
<td>LCS</td>
<td>I have inspected the car and here is my report</td>
</tr>
</tbody>
</table>

- Check if the task or event indicates commitment creation
- Check if another task delegates, discharges, or cancels the commitment.

2.4.1 Task

A task is a business activity that is either predefined (part of a best practice process) or created on-the-fly by participants in a conversation [59]. We represent a task as $T$ and define it as $T$($\text{task performer}$, $\text{beneficiary}$, $\text{action}$). Here, $\text{task performer}$ is a business entity that performs the action. $\text{beneficiary}$ is a business entity for whom the action is performed. An $\text{action}$ is a business activity. An action can be a disjunction or a conjunction of subactions. Table 2.2 shows messages from Table 2.1 that are identified as tasks.

Table 2.2: Tasks identified from the interactions of Table 2.1

<table>
<thead>
<tr>
<th>Task Performer</th>
<th>Beneficiary</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGFIL</td>
<td>John</td>
<td>Inspect car damage</td>
</tr>
<tr>
<td>LCS</td>
<td>AGFIL</td>
<td>Inspect car damage</td>
</tr>
<tr>
<td>M</td>
<td>LCS</td>
<td>Inspect car damage</td>
</tr>
<tr>
<td>M</td>
<td>LCS</td>
<td>Cannot inspect</td>
</tr>
<tr>
<td>M1</td>
<td>LCS</td>
<td>Inspect car damage</td>
</tr>
<tr>
<td>M1</td>
<td>LCS</td>
<td>Inspected car damage</td>
</tr>
</tbody>
</table>
2.4.2 Commitment Creation

In business interactions through email or chat, most interactions indicate an unconditional commitment. Therefore, if a message contains a task (T) and indicates an unconditional commitment (C), we conclude the task performer of T is the debtor of C, the beneficiary of T is the creditor of C, and the action of T is consequent of C. An unconditional commitment may be created in two ways. In a commissive create (C-create), the debtor voluntarily offers to perform the consequent for the creditor. In a directive create (D-create), an appropriate party is empowered to direct the debtor. Table 2.3 shows messages from Table 2.1 that are identified as C-create and D-create.

Table 2.3: Commitment creation identified from the interactions in Table 2.1.

<table>
<thead>
<tr>
<th>Debtor</th>
<th>Creditor</th>
<th>Consequent</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGFIL</td>
<td>John</td>
<td>Inspect car damage</td>
<td>C-create</td>
</tr>
<tr>
<td>LCS</td>
<td>AGFIL</td>
<td>Inspect car damage</td>
<td>D-create</td>
</tr>
<tr>
<td>M</td>
<td>LCS</td>
<td>Inspect car damage</td>
<td>D-create</td>
</tr>
<tr>
<td>M1</td>
<td>LCS</td>
<td>Inspect car damage</td>
<td>D-create</td>
</tr>
</tbody>
</table>

2.4.3 Commitment Discharge

A commitment is discharged when the debtor performs the consequent, thereby making it true. Table 2.4 shows messages from Table 2.1 wherein first M1 creates a commitment toward LCS (as directed by LCS) and subsequently M1 discharges the commitment by conveying that he or she inspected the car for damage.

Table 2.4: Discharge commitment identified from the interactions in Table 2.1.

<table>
<thead>
<tr>
<th>Debtor</th>
<th>Creditor</th>
<th>Consequent</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>LCS</td>
<td>Inspect car damage</td>
<td>D-create</td>
</tr>
<tr>
<td>M1</td>
<td>LCS</td>
<td>Inspected car damage</td>
<td>Discharge</td>
</tr>
</tbody>
</table>
2.4.4 Subcontracting a Commitment

A commitment $C(\text{debtor}, \text{creditor}, \top, \text{consequent})$ is subcontracted when its debtor outsources it to a new debtor ($\text{debtor'}$). A new commitment is created $C(\text{debtor'}, \text{debtor}, \top, \text{consequent})$ but the original commitment remains. Table 2.5 shows messages from Table 2.1 wherein first a commitment is created from AGFIL toward JOHN and later the commitment is subcontracted, first from AGFIL to LCS, and then from LCS to M and M1.

Table 2.5: Subcontract commitments identified from message interactions in Table 2.1.

<table>
<thead>
<tr>
<th>Debtor</th>
<th>Creditor</th>
<th>Consequent</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGFIL</td>
<td>John</td>
<td>Inspect car damage</td>
<td>C-create</td>
</tr>
<tr>
<td>LCS</td>
<td>AGFIL</td>
<td>Inspect car damage</td>
<td>Subcontract</td>
</tr>
<tr>
<td>M</td>
<td>LCS</td>
<td>Inspect car damage</td>
<td>Subcontract</td>
</tr>
<tr>
<td>M1</td>
<td>LCS</td>
<td>Inspect car damage</td>
<td>Subcontract</td>
</tr>
</tbody>
</table>

2.4.5 Commitment Cancellation

A commitment is canceled when its debtor terminates the commitment. Table 2.6 shows messages from Table 2.1, wherein LCS first subcontracts its commitment to inspect JOHN’s car to M by sending the message *Please inspect the car for damage*, and next M cancels the commitment by uttering *I cannot inspect as I am busy with other work*.

Table 2.6: Cancel commitment identified from message interactions in Table 2.1.

<table>
<thead>
<tr>
<th>Debtor</th>
<th>Creditor</th>
<th>Consequent</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>LCS</td>
<td>Inspect car damage</td>
<td>D-create</td>
</tr>
<tr>
<td>M</td>
<td>LCS</td>
<td>Cannot inspect</td>
<td>Cancel</td>
</tr>
</tbody>
</table>
2.5 Approach to Identify and Monitor Commitments in Agents’ Interactions

Our process for the identification of commitments from interactions proceeds as shown in Figure 5.1. First, we preprocess our datasets and extract sentences from the text of interactions. Second, using natural language processing and a set of heuristic rules applied on features extracted from the conversation text, we identify tasks and commitments. Third, to overcome the limitations of heuristics, we augment our approach with a supervised machine learning approach for the identification of commitments and their lifecycle. Applying machine learning helps us identify commitments for which their various expressions and forms in the natural language may not be captured in fixed patterns and rules.

![Diagram of process to identify commitments](image)

Figure 2.3: Process followed to identify commitments.

2.5.1 Typed Dependency

To identify a task $T$ from a sentence, we adopt the typed dependency method [20], which outputs the relations between individual words in a sentence. A relation between any two words is a triple of the name of the relation, governor, and dependent. For example, consider the sentence *I will inspect your car for damage* from Table 2.1. Here, the triples are nsubj(inspect, I), aux(inspect, will), root(ROOT, inspect), poss(car, your), dobj(inspect, car), prep(inspect, for), and pobj(for, damage). Figure 2.4 shows the triples in a graph format. Below, we explain how we extract tasks from a sentence using these triples.
2.5.2 Identifying interactions and Sentences

We consider both the Enron email corpus and a proprietary HP IT incident management dataset for the evaluation. We preprocess both datasets to make them suitable for parsing and extracting features. Since the email and chat datasets are differently structured, we follow different steps to preprocess them from both these types. For email, we separate information such as sender, receiver, date, and subject. Then we prepare conversation threads by collecting all the emails either replied or forwarded with the same subject name. Next, we split each email into its constituent sentences and parse each of these sentences to extract its features. Unlike for emails, we do not prepare conversation threads for chat interactions as they are already listed chronologically.

2.5.3 Extracting Features

We perform the following steps to extract features from each sentence in emails and chat messages.

- **Coreference resolution** relates a name with a personal pronoun. For example, in a pair of sentences *Please add Jim Curry to your list. He should be part of the due diligence team*, the coreference resolution helps to relate *Jim Curry* (name) with *He* (personal pronoun). This is important because several interactions start with *you* or *he* or *she* or *they* and it is necessary to resolve these pronouns so that we can identify the debtor and creditor of a commitment.

- **Named entity resolution** (NER), identifies for a noun whether it is a PERSON or an ORGANIZATION. Upon identifying a commitment we check whether the debtor and the
creditor of the commitment is a valid debtor by checking if it is a person or an organization from the resolved name entities.

- **Part-of-speech tags extraction** We extract Part-of-Speech (POS) tags for each word, which help identify the type of personal pronoun for a task performer and the state of the verb associated with the performer so as to identify the debtor of a commitment and the state of a commitment, respectively. The present tense of the verb indicates that a commitment is created whereas the past tense indicates a commitment is discharged.

- **Typed dependencies extraction** As discussed above, a typed dependency relates words in a sentence and indicates its logical structure.

Let us discuss the key features in the features used to train our classifiers. The features are based on properties that help identify a sentence as creating, delegating, discharging, or cancel. The features are:

- A **modal verb** signals the creation of a commitment (e.g., will and shall).
- An **action verb** indicates whether a commitment is present in a sentence (e.g., inspect).
- The **present tense** signals the creation, delegation, or cancellation of a commitment.
- The **past tense** signals the discharge of a commitment (e.g., inspected).
- The **debtor** of a commitment is the task performer.
- The **creditor** of a commitment is the one the debtor commits to.
- A **deadline** indicates a commitment creation or delegation (e.g., by tomorrow, by Monday)
- The prior **creation of a commitment** is a prerequisite for discharge, delegation, and cancellation if the create commitment already exists.
- A **subcontract signal** is identified when a debtor directs a new creditor.
- A **negative verb** indicates the presence of a canceled commitment (e.g., cannot inspect).
- The **type of the personal pronoun** in the subject indicates a commitment being created, canceled, or discharge (first, second, or third person) or delegation (second or third).
- The **bigram of a modal verb and a second person pronoun** indicates a directive creation (e.g., can you).
- The **bigram of a first person pronoun and a modal verb** indicates a commissive (e.g., I will).
• The bigram of “please” and an action verb indicates a directive (e.g., please inspect).

• A question mark in a sentence indicates a directive commitment creation.

2.5.4 Identifying Tasks

To identify a task from a sentence, we first extract the features discussed in Section 2.5.3. To obtain the features, we parse a sentence to obtain a typed dependency array containing the triples as shown in Figure 2.4. In a typed dependency array, first, we look for the nsubject relation and check if the dependent in the relation is a valid subject (personal pronoun, organization, or person) and the governor is a valid action verb (VB, VBD, VBP, VBZ, or VBN). If both the governor and the dependent are valid, we store the dependent as the task performer and the governor as the action for the task performer. We extract the action details using the action verb by finding its dependencies in the array of triples by looking for nouns or verbs associated with the action verb.

Figure 2.5: Steps to identify task from a email sentence “I will inspect your car for damage.”

Figure 2.5 represents the task structure we obtain after following the above approach on the typed dependency in Figure 2.4. In the task structure, as shown, the task performer extracted is I. Since the task performer indicates a first person personal pronoun, the actual performer is the sender of the message and the beneficiary is the receiver of the message. Here, the action is inspect damage car.

2.5.5 Identifying Creations

Once we have extracted a task from a sentence, we check whether the task indicates the creation of a commitment. To identify such tasks, we check whether the action verb in the task is in present tense and has a relationship with a modal verb and the word please. If so, we store the task performer as the debtor and the action as the consequent, respectively, of the commitment.

In Figure 2.6, the action verb inspect is in the present tense (VB) and has a relationship with a modal verb will. Therefore, the task indicates a commitment being created.
2.5.6 Identifying Subcontracts

To identify a subcontract in a sentence, we check if a commitment $C_2$ has been created after commitment $C_1$, as shown in Figure 2.7. Then we check if the debtor (AGFIL) in $C_1$ is the creditor (AGFIL) in $C_2$ and the consequents (inspect damage car) in both $C_1$ and $C_2$ are same. To match the consequents in the commitments, we check whether the action verbs and nouns in both commitments are the same or related using WordNet [55] dictionary and the coreference resolution, respectively.

2.5.7 Identifying Discharges

If an identified task has its action verb in the past tense, then it may signal a discharge commitment, provided a commitment is already created. For clarity, consider the example of a commitment $C_2$ and a task $T_2$ in Figure 2.8. To check if $T_2$ discharges $C_1$, we compare the task performer (M1) and the beneficiary (LCS) in $T_2$ with the debtor (M1) and the creditor (LCS)
in C₁, respectively. If they are the same, we compare their action verbs. We check whether the action verb (inspected) in T₂ is in the past tense (VBD). Then we compare the action verb in T₂ by converting it into its base form (inspect) and trying to match with the action verb (inspect) in C₁. If the verbs are the same or related, we compare the nouns in the two tasks. If they are the same or related, we mark T₂ as discharging C₁.

2.5.8 Identifying Cancellations

For identifying a canceled commitment, we compare a task with the commitments that already exist and check whether there is a relation in the type dependency array where the action verb is associated with a negative word such as not. Figure 2.9 describes an example of a commitment C₁ and a task T₂. Once we find that the debtor (M) and the creditor (LCS) in C₁ and T₂ are the same, we compare their main action verbs and nouns, respectively. If these verbs and nouns are the same, we check whether the action verb is a negative verb. Note that a verb considered as negative if it has a relation with a negative word such as not.

2.6 Evaluation and Prototype

We validate our contribution, via two steps: (1) an automatic labeling of the data (sentences) using the approach of Section 2.5 and (2) manually labeling a subset of the data and using it for training and testing our approach.

2.6.1 Data

From the Enron email corpus [26, 47], we selected 4,161 email sentences that were exchanged between Kimberly Watson, an employee of Enron, and more than 50 people, including her colleagues at Enron, clients, friends, and family members. For the chat data, we selected 271 interactions from HP’s IT incident management logs comprising 7,154 sentences.
2.6.2 Labeling Data

Two annotators (graduate students in computer science) labeled the sentences. We resolved conflicts by allowing the two annotators to discuss their labels for sentences. Table 2.9 shows the distribution of the email and chat sentences as annotated. Next, we ran the Naïve Bayes (NB), Logistic Regression (LR), and Support Vector Machine (SVM) classifiers and used ten-fold cross-validation to produce our results. We use NB, LR, and SVM classifiers as they are among the most popular ones used for text classification.

<table>
<thead>
<tr>
<th>C-create</th>
<th>D-create</th>
<th>Discharge</th>
<th>Cancel</th>
<th>Subcontract</th>
</tr>
</thead>
<tbody>
<tr>
<td>P R F</td>
<td>P R F</td>
<td>P R F</td>
<td>P R F</td>
<td>P R F</td>
</tr>
<tr>
<td>NB 0.84</td>
<td>0.94</td>
<td>0.89</td>
<td>0.81</td>
<td>0.93</td>
</tr>
<tr>
<td>LR 0.90</td>
<td>0.95</td>
<td>0.92</td>
<td>0.93</td>
<td>0.94</td>
</tr>
<tr>
<td>SVM 0.87</td>
<td>0.97</td>
<td>0.92</td>
<td>0.94</td>
<td>0.97</td>
</tr>
</tbody>
</table>

2.6.3 Results

Tables 2.7 and 2.8 present our results for the email and chat data, respectively, using the NB, LR, and SVM classifiers and ten-fold cross validation. We use the following well-known metrics to show our results.

\[
\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}
\]
\[
\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}
\]

\[
\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

Table 2.9: Distribution of commitment operations in email sentences.

<table>
<thead>
<tr>
<th>Classes</th>
<th>Email</th>
<th>Chat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commissive create</td>
<td>342</td>
<td>532</td>
</tr>
<tr>
<td>Directive create</td>
<td>162</td>
<td>214</td>
</tr>
<tr>
<td>Discharge</td>
<td>38</td>
<td>250</td>
</tr>
<tr>
<td>Cancel</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>Delegate</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>None</td>
<td>3,540</td>
<td>6,130</td>
</tr>
</tbody>
</table>

The NB classifier performs well in most cases though it assumes all the attributes are conditionally independent of one another. One major shortcoming of NB is that it needs large datasets. Using NB, for email, our results show high F-measures for commissive creation and directive creation and low F-measures for subcontract, discharge, and cancellation. For chats, we obtain slightly lower F-measures for commissive and directive creation and a higher F-measure for discharge than email.

LR is well-suited to modeling continuous valued functions and can predict accurately even for small datasets. Using LR, for email, we obtain significantly high F-measures for commissives and directives compared to NB, and low F-measures for subcontract, discharge, and cancel. For chat, LR performs better than NB for commissive creation and cancellation, but its results for other classes are lower.

SVM uses nonlinear mapping to convert the training data to a higher dimension, and looks for a linear optimal separating hyperplane. A hyperplane separates one class from another. SVM is useful as it can model complex nonlinear decision boundaries and are less prone to over-fitting. Using SVM, in email, we obtain significantly higher F-measures for commissive creation, directive creation, and subcontract and low F-measure for discharge and cancellation. For chat, using SVM, we obtain lower F-measures than for emails for commissive creation, directive creation, but, high for F-measure for discharge.

We obtain high precision, recall and F-measures for both commissive and directive create for both emails and chats using NB, LR, and SVM. The results for these classes are high because they are independent of each other and occur frequently in both datasets. The results for other
classes are low because they depend on the prior existence of a commitment and it is difficult to find this specific feature automatically. Compared to discharge and subcontract for chat, the result for subcontract is higher for emails because we can easily identify the debtor and creditor of a commitment based on the sender’s and receiver’s information. The results for discharge in email is low because—as it turns out—discharge occurs rarely in emails. In case of chat, the precision for discharge is higher because the distribution of discharge is high and participants in chat messages tend to immediately report their progress. However, the overall percentage is low for both emails and chats because it is difficult to identify the consequent of a commitment across sentences. For emails, we find a high precision using SVM with low recall and F-measure. We attribute the high precision of SVM to some of the sentences in emails that were identified accurately as discharge by our algorithm. For cancel, we obtained 0% F-measure in emails and 16% in chats. This is because, as we said earlier, it is difficult to identify a prior commitment and identify the negative words associated with the action verb.

As shown in Table 2.10, we evaluated our trained model on two test datasets drawn from the Enron and HP corpora and containing 1,326 email and 2,299 chat sentences, respectively. The test datasets are disjoint from our training datasets. The sentences in the datasets were manually labeled by two annotators by resolving their conflicts. For emails, we used SVM and for chats we used LR.

Table 2.10: Evaluation on independent test datasets using SVM for email and LR for chat, respectively.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Email P</th>
<th>Email R</th>
<th>Email F</th>
<th>Chat P</th>
<th>Chat R</th>
<th>Chat F</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-create</td>
<td>0.97</td>
<td>0.84</td>
<td>0.90</td>
<td>0.89</td>
<td>0.89</td>
<td></td>
</tr>
<tr>
<td>D-create</td>
<td>0.94</td>
<td>0.78</td>
<td>0.86</td>
<td>0.83</td>
<td>0.51</td>
<td>0.63</td>
</tr>
<tr>
<td>Discharge</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.66</td>
<td>0.71</td>
<td>0.69</td>
</tr>
<tr>
<td>Cancel</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Delegate</td>
<td>1.00</td>
<td>0.33</td>
<td>0.98</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>0.96</td>
<td>0.99</td>
<td>0.98</td>
<td>0.96</td>
<td>0.98</td>
<td>0.94</td>
</tr>
</tbody>
</table>

Table 2.11 shows sample examples extracted from the Enron corpus using our approach.
2.6.4 Prototype Tool

Figure 2.10 presents the architecture of our interactive tool for commitments identification and tracking. The tool can be potentially plugged into chat and email clients. It identifies commitments in a panel for verification and confirmation. When a person sends a chat or an email message, it parses the sentence using the parser component and extracts potential task details and features. Using the predictor component and a trained classifier, we identify the classes based on the extracted features. The trained classifier used is generic as the features remains the same for all domains. Its parser and predictor components are Java-based (Stanford Parser [45] and Weka libraries [36] for SVM). Our tool displays a summary of all tasks and commitments based on chats. This tool can be effectively used to identify and manage commitments in the context of service engagement platforms such as IT service management domain where chat is the primary means of communication.

![Diagram of the architecture of the commitment identification and monitoring tool.]

Figure 2.10: The architecture of our commitment identification and monitoring tool.

2.7 Related Work

Scerri et al. [70] focus on action items in emails and check whether these action items fall broadly under the request, suggest, assign, and deliver classes. For identifying the classes, they use a rule-based classification model. Scerri et al.’s work is promising as it provides an automatic approach for identifying classes. However, it does not apply machine learning, resulting in low accuracy for identifying creation commitment. Lampert et al. [48] improve over the accuracy results of Scerri et al.’s [70] work by using supervised machine learning combined with email zoning. Lampert et al.’s accuracy result is high (84%). However, it would be difficult to use
their model because they do not focus on identifying task and commitment parameters, which are essential for our work.

Qadir and Riloff [63] classify sentences from message board posts as commissives, directives, expressives, and representatives. Using the SVM classifier, they obtain high accuracy for commissives and directives. Their work is limited to speech acts and does not identify commitment parameters.

Researchers have also worked on extracting policies, rules, and norms from unstructured text such as contracts. Martínez-Fernández et al. [53] provide an approach to extract semantics of business vocabulary and rules language (SBVR) from unrestricted text. Their work is preliminary and they do not provide any accuracy result. Bartolini et al. [6] semantically annotate and extract deontic norms such as obligation, prohibition, and permission from Italian legal texts. De Maat et al. [19] automatically identify different norms from Dutch laws. Savarimuthu et al. [68] propose an architecture to infer the obligation norm in a multiagent society. For our work, we focus on extracting commitments from emails and chats.

Molina-Jiménez et al. [56] propose a way to describe a contract in terms of finite state machines (FSM). To create an FSM, they extract rights and obligations from contract text. Then they execute the FSM to monitor the contract. Molina-Jiménez et al.’s work is limited to static texts like contracts, whereas we focus more on dynamic texts as in email and chat. Moreover, their approach is manual.

Process mining addresses extracting and monitoring orchestrated processes rather than people-driven processes. Van der Aalst et al. [83] extract such workflows from event logs containing workflow enactments. Desai et al. [22] propose an approach to trace processes from unstructured execution logs. They apply their approach to real-world business processes in a service delivery center. Günther et al. [34] mine changes in logs in adaptive process management systems.

2.8 Conclusions and Future Work

Current techniques in service computing focus on automated service engagements. However, the human, ad hoc aspects of service engagements are the most challenging. In particular, many important and expensive service engagements are people driven and traditional techniques simply do not apply on them.

Our approach, realized in a tool, is quite effective at inferring the creation and some other operations on the commitments that arise among the participants in a service engagement. First, our approach promises, with suitable enhancements, a novel means for the monitoring of commitments in people-driven service engagements as a basis for judging whether they are successful. Second, it could also form the basis of an approach for mining ad hoc processes that
underlie people-driven service engagements.

This research opens up some important future directions. In particular, we will develop enhanced methods for detecting the delegation, discharge, and cancellation of commitments. Also, we will investigate unsupervised techniques, which would reduce the burden of manually labeling data. We will study improved models that capture richer patterns as seen in real-life settings. Such models can facilitate both monitoring and mining.
Table 2.11: Sample extracted from the Enron corpus.

<table>
<thead>
<tr>
<th>Sentences</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create</strong></td>
<td></td>
</tr>
<tr>
<td>We will expedite materials and installation in an attempt to meet the target date. If any questions, please let me know.</td>
<td>C-create</td>
</tr>
<tr>
<td>Please review and send along to your attorney as soon as possible</td>
<td>D-create</td>
</tr>
<tr>
<td><strong>Discharge</strong></td>
<td></td>
</tr>
<tr>
<td>I will also check with Alliance Travel Agency to see what may be able to do for us</td>
<td>C-create</td>
</tr>
<tr>
<td>I checked with our Travel Agency and they cannot secure cheaper tickets than what we are seeing on the internet</td>
<td>Discharge</td>
</tr>
<tr>
<td><strong>Subcontract</strong></td>
<td></td>
</tr>
<tr>
<td>Please take a few moments to review the same and let me know your thoughts</td>
<td>D-create</td>
</tr>
<tr>
<td>This appears to be OK and we should be able to sign on however please review the statement and let me know if you see a problem with our support of the PHC statement</td>
<td>Subcontract</td>
</tr>
<tr>
<td><strong>Cancel</strong></td>
<td></td>
</tr>
<tr>
<td>By Wednesday Aug 16 2001, please send all copies of your documentation via interoffice mail to Laura Herrera</td>
<td>D-create</td>
</tr>
<tr>
<td>Robbin, please forgive me for not sending this in by Aug 15</td>
<td>Cancel</td>
</tr>
</tbody>
</table>
Chapter 3

Güven: Estimating Trust From Commitments

3.1 Introduction

We propose Güven a computational model of trust founded on commitments that supports how an agent determine trust in others based on their interactions. Commitments are important for trust because they can be identified from interpersonal interactions and can help us characterize the outcomes of such interactions in high-level terms. We limit Güven to commitments, although, it can be extended to related concepts such as prohibitions and authorizations.

We conduct an empirical evaluation, on emails (automatically analyzed using our previous approach [42]). We show how to train the model parameters so as to capture a user model indicating each user’s propensity to trust given commitment outcomes. Our evaluations yield promising, but imperfect, results on the viability of inferring trust from the commitments arising in interactions, suggesting the need for better extraction techniques. Our main contribution is to show how trust can be computed, not just theorized about, via the domain-independent concept of commitments.

3.2 Background on Commitments

Figure 3.1 presents the commitment lifecycle we adopt. A commitment is created when a debtor either voluntarily creates it (commissive creation) or is directed to do a certain task (directive creation). Given the debtor’s autonomy, the latter presumes a prior commitment on part of the debtor. A commitment is detached if a condition or an antecedent present for a commitment holds true; discharged when a debtor executes a committed task. A commitment is terminated when a debtor cancels the commitment before it is detached or a creditor releases the commit-
ment. A commitment is violated when a debtor cancels the commitment after it is detached or when a consequent timeout occurs. Additionally, a commitment can be delegated and assigned. A commitment is delegated when the debtor of a commitment is replaced by a new debtor and assigned when the creditor of a commitment is replaced by a new creditor. We map interactions between persons to commitment operations.

3.3 Intuitions on Trust and Commitments

We describe some criteria for how trust values may be updated based on operations on their commitments.

3.3.1 Effects of Commitment Operations

We describe the effect of commitment operations on trust. Before we describe the effects, let us consider some situations wherein a commitment exists from a debtor toward a creditor.

**Effect of discharge.** When the commitment is discharged, the trust of the creditor for the debtor, increases.

**Effect of violation.** When the commitment is violated, the creditor’s trust in the debtor decreases.

**Effect of delegation and discharge.** When the commitment is delegated by the original debtor to a new debtor, and the new debtor satisfies it, the creditor’s trust in both the original and the new debtor increases.

**Effect of delegation and violation.** When the commitment is delegated from the original...
debtor to a new debtor and the new debtor violates it, the creditor’s trust in both the original and the new debtor decreases.

**Effect of assignment and discharge.** When the commitment is assigned from the original creditor to a new creditor and the original debtor discharges it, the trust of the original and new creditor for the original debtor increases.

**Effect of assignment and violation.** When the commitment is assigned from the original creditor to the new creditor and the debtor violates it, the trust of the original and the new creditor for the original debtor decreases.

We make the following assumptions regarding the increase or decrease of trust. These are simplifying assumptions and could be relaxed in some settings.

- In our basic approach, the change in trust is the same for all commitments. We additionally provide an approach in which the change in trust depends upon the strength of a commitment.

- We assume commitment discharge and violation to be all or none; in our scenarios, partial success is not easy to infer.

- In case of violation, the trust of the creditor for the debtor decreases irrespective of whether the debtor was truly responsible. A person’s beliefs and goals are private and cannot be identified directly from his or her interactions.

- In case of delegation, the original creditor’s trust in the original and new debtor changes equally, reflecting the idea that the creditor has a positive experience thanks to the two debtors.

- In case of assignment, the new creditor’s trust in the debtor changes as much as the original creditor’s trust in the debtor, reflecting the intuition that both creditors’ expectations are met.

### 3.3.2 Subjectivity, Memory, and Strength

Trust is modulated by features that affect how trusters judge outcomes, such as the satisfaction or violation of a commitment. First, trust assessment is subjective. Trusters differ in how they reward or penalize a trustee when a commitment is discharged or violated, respectively. Second, trust assessment depends on the truster’s memory: trusters with limited memory would tend to forget all but (some varying number of) recent experiences. Recent experiences may turn out to be more predictive of future experiences (that trust is about) than past experiences. Third, the effect on trust of a commitment’s outcome would be greater when the commitment is more important.
3.3.3 Hypotheses and Evaluation Strategy

We now present our research hypotheses in informal terms. These hypotheses motivate our evaluation strategy, computational methods, and study design. The following section on evaluation refines these hypotheses into technical claims.

\( H_1 \) Predicting trust values by learning trust parameters for each subject yields more accurate results than using fixed trust parameters for all subjects.

The details of the trust parameters are given in Section 3.4. Assuming \( H_1 \) holds, we consider learned parameters as the baseline approach. We check if other approaches improve accuracy beyond the baseline.

\( H_2 \) Predicting trust values by learning a specific discount window size for each subject yields more accurate results than the baseline.

\( H_3 \) Predicting trust values by inferring strengths of positive and negative experiences yields more accurate results than the baseline.

3.3.4 Evaluation Strategy

Figure 3.2 summarizes our evaluation process. Our evaluation strategy is to gather data from subjects in the two decision contexts and proceed as follows.

**Step 1.** Build a dataset of interpersonal interactions with trust values. For emails, subjects provide third-party assessments; for games, subjects provide their own trust assessments. Table 3.1 shows the examples of email interactions. Based on the emails exchanged between Kim and Dorothy different subjects assign different trust values in the range of 0
and 1 from Kim toward Dorothy and from Dorothy toward Kim. Similarly, for chats $P_1$ assigns a trust value for $P_4$ and $P_4$ assigns a trust value for $P_1$.

**Step 2.** Identify commitment operations from the interactions. For emails, using Kalia et al.’s [42] trained classifier. Table 3.1 shows example commitment operations identified in emails, respectively.

### Table 3.1: Examples of email interactions.

<table>
<thead>
<tr>
<th>Sender</th>
<th>Receiver</th>
<th>Email Content</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kim</td>
<td>Dorothy</td>
<td><em>I will also check with Alliance Travel Agency ...</em></td>
<td>create($C_1$)</td>
</tr>
<tr>
<td>Kim</td>
<td>Dorothy</td>
<td><em>I checked with our Travel Agency ...</em></td>
<td>discharge($C_1$)</td>
</tr>
<tr>
<td>Rob</td>
<td>Kim</td>
<td><em>By Wednesday Aug 16 2001, please send all copies of your documentation ...</em></td>
<td>create($C_2$)</td>
</tr>
<tr>
<td>Kim</td>
<td>Rob</td>
<td><em>Rob, please forgive me for not sending this in by Aug 15</em></td>
<td>cancel($C_2$)</td>
</tr>
</tbody>
</table>

**Step 3.** Partition the dataset into training and test datasets. Learn model parameters for each subject from the training data.

**Step 4.** Apply the learned model to predict trust in the test data and compute the model’s accuracy.

We repeat the process for all subjects and present our results.

### 3.4 Güven: Model of Trust based on Commitments

We adopt Wang and Singh’s [85] trust model, which represents trust as evidence $\langle r, s \rangle$. Here, $r \geq 0$ and $s \geq 0$ respectively represent the positive and negative experiences the truster has with the trustee. Both $r$ and $s$ are real numbers. Wang and Singh calculate trust as the probability of a positive outcome as $\alpha = \frac{r}{r+s}$. Suppose Buck and Selia transact 10 times and exactly eight transactions succeed from Selia’s perspective. Then Selia’s trust in Buck would be 0.8.

The basic idea is for each truster to maintain evidence $\langle r, s \rangle$ about each trustee. The initial evidence, $\langle r_{in}, s_{in} \rangle$, represents the truster’s bias. An interaction may yield a positive, negative, or a neutral experience. In these cases, the evidence is updated by respectively adding $\langle i_r, 0 \rangle$, $\langle 0, i_s \rangle$, or $\langle i_r, i_s \rangle$. The updated evidence is then used to calculate the updated trust in the next interaction.
\( \langle 0, i_s \rangle \), and \( \langle \lambda_i r, (1 - \lambda)i_s \rangle \), where \( \lambda \in [0, 1] \). In essence, we characterize each truster via five parameters \( (i_r, i_s, r_{in}, s_{in}, \lambda) \).

### 3.4.1 Considering Subjectivity

To evaluate \( H_1 \), we learn a specific truster’s parameters based on positive, negative, and neutral experiences it acquires from trustees and the truster’s actual trust in various trustees. For the \( k \)th trustee, let \( \alpha_k \) represent the truster’s actual (as revealed) and \( \hat{\alpha}_k \) the truster’s predicted trust in \( k \). Let \( E^+_k \), \( E^-_k \), and \( E_k \) represent the numbers of positive, negative, and neutral experiences, respectively. Then,

\[
\hat{\alpha}_k = \frac{r_{in} + E^+_k i_r + \lambda \cdot E_k i_r}{r_{in} + s_{in} + E^-_k i_r + E^-_k i_s + E_k (\lambda i_r + (1 - \lambda)i_s)}
\]  

(3.1)

Via nonlinear least-squares regression technique that uses trust region reflective algorithm [16], we estimate the truster’s parameters to minimize the mean absolute error (MAE) of prediction, \( \sum_{k=1}^n |\hat{\alpha}_k - \alpha_k| \).

### 3.4.2 Considering Memory

We capture the effect of memory by considering a discount window, defined simply as the most recent \( W \) experiences. Let \( n \) be the total number of experiences the truster acquires from the trustee. Let \( t = \min(n, W) \). Let \( E^+_t \), \( E^-_t \), and \( E^-_t \) be the positive, neutral, and negative experiences inferred from the \( t \) transactions. The trust of a truster in the trustee depends on whether \( t \) is less than \( W \). When \( t < W \), the truster’s trust is \( \langle r_{in} + E^+_t i_r + \lambda \cdot E_t i_r, s_{in} + E^-_t i_s + (1 - \lambda)E_t i_s \rangle \); otherwise, it is \( \langle r_{in} + E^+_t i_r + \lambda \cdot E_t i_r, s_{in} + E^-_t i_s + (1 - \lambda)E_t i_s \rangle \). When \( t < W \) we ignore the initial bias as the truster’s trust is based on recent \( W \) experiences, which simply means that the truster has already forgotten its initial bias.

### 3.4.3 Considering Strength

We posit that a truster acquires experiences of varying weights based on commitment outcomes (satisfied or violated). To calculate the weight, we identify the following features in a sentence indicating a commitment creation. Except the feature action verb, we evaluate rest of the features empirically from a subject evaluation. We provide the outcome of our evaluation in Table 3.5.

**Commissive over directive.** A commissive (e.g., “I will . . .”) may carry a greater weight than a directive (e.g., “Could you please . . .”) because it holds even without the presumption of another commitment.
Debtor’s type. A single debtor may carry a greater weight than multiple debtors “we” (“We will follow up”). A single debtor, as in “I will follow up,” has clearer responsibility than multiple debtors.

Creditor’s type. Multiple creditors may carry a greater weight than a single creditor. Multiple creditors arise when a debtor commits to a set, e.g., when a product manager commits to his employees to review a product. The intuition is that having multiple creditors makes the debtor accountable to more parties.

Modal verbs. Some modal verbs (e.g., will or shall) may convey high confidence over others (e.g., can, could, may, would) [60]. The intuition is that “will” indicates that a commitment will be surely satisfied whereas “can” indicates that the commitment may not be satisfied. We learn the weights of different modal verbs based on data obtained from human subjects.

Action verbs. Some action verbs convey a greater level of importance than others. For example, “resolving an issue” may be more important than “reviewing a proposal.” We compute the weights of verbs using Burchardt et al.’s [9] FrameNet tool, which provides weights for words used in different senses, e.g., 1 for resolve and 0.383 for review.

Deadlines. Noun phrases with deadlines [17] may convey more importance than noun phrases without deadlines. For example, an explicit deadline, as in “I will repair the car by Monday,” enhances the importance of the commitment. We assume that merely mentioning a deadline increases the seriousness of a commitment. We defer to future research additional subtleties, such as the duration or urgency of a deadline and the extent to which it may be broken, since in our empirical settings durations and urgency do not arise.

Our evaluation ranks feature values as discussed above. We map the ranks to weights (cardinal numbers, higher for higher ranked features) and sum the weights to compute a value.

3.5 Evaluation

We evaluated Güven via an empirical study with 30 subjects (graduate and undergraduate students from various departments). We conducted the study in two phases. In this phase, subjects read 33 emails selected from the Enron email corpus [26, 47] and provided a trust value ranging from 0 to 1 between the senders and receivers of email. The emails were selected on the basis of containing sentences that indicate commitment creation, satisfaction, or violation—such sentences were identified using Kalia et al.’s [42] method. We augmented the dataset with 28 synthetic sentences indicating commitment satisfaction or violation, which do not occur
frequently in the corpus. Subjects provided trust values based on their intuitions by reading these emails. We did not disclose the commitments identified. We did not provide any additional guidelines that might restrict a subject’s individual perception of trust. Once subjects have provided trust values, we mapped commitment operations to positive, negative, and neutral experiences. Table 3.2 shows an example of two rows created from the first two interactions between Kim and Dorothy given in Table 3.1. S1, . . ., S6 in Table 3.2 represent the subjects who have provided trust values based on the interactions between Kim and Dorothy. Based on the experiences collected from emails and trust values collected from subjects, we created 28 rows of data for each subject. Hence, for 30 subjects we get 28*30 = 840 rows.

Table 3.2: Different features and trust values from different subjects.

<table>
<thead>
<tr>
<th>Trust Pairs</th>
<th>Experiences</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kim→Dorothy</td>
<td>2 Neutral, 0 Positive, 0 Negative</td>
<td>0.3</td>
<td>0.4</td>
<td>0.35</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>Dorothy→Kim</td>
<td>1 Neutral, 1 Positive, 0 Negative</td>
<td>0.7</td>
<td>0.9</td>
<td>0.8</td>
<td>0.7</td>
<td>0.6</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Additionally, subjects were given 10 questions based on different features as discussed in Section 3.4.3. The questions were about ranking these features’ values in order of their perceived importance. Hypotheses $H_1$, $H_2$, and $H_3$ are about evaluating the proposed approaches with respect to their prediction accuracy.

3.5.1 Results

[Verifying $H_1$] We collected the trust values from the subjects from the emails assigned to them. We divided the data collected from subjects into three-fold training and test data and learned trust parameters for each subject ($r_i$, $s_i$, $i_r$, $i_s$, $\lambda$) that minimize MAE between predicted and actual trust values.

For verifying $H_1$, we calculated the MAE for $\lambda$ ranging from 0.1 to 0.9. Then, we calculated the MAE by learning the $\lambda$ (Learned($\lambda$)) itself. Based on the above MAEs, we obtained a customized $\lambda$ (fixed or learned) for each subject. A customized $\lambda$ for a subject refers to the value of $\lambda$ for which the MAE is minimum. We represent the MAEs obtained using customized $\lambda$s for all subjects as Custom($\lambda$) in Figure 3.3. Finally, we arbitrarily assumed some fixed configurations of parameters ($F_1 = \langle 1,1,1,1,0.5 \rangle$, $F_2 = \langle 2,1,1,1,0.5 \rangle$, $F_3 = \langle 1,2,1,1,0.5 \rangle$). $F_1$ indicates no bias in the initial trust perception where as $F_2$ and $F_3$ indicate positive and negative biases respectively. $\lambda=0.5$ in fixed configurations indicates equal trust increments for the neutral experiences. The configurations can be changed by incrementing or decrementing
Table 3.3: Statistical test results for H1. (μ-mean of the MAEs)

<table>
<thead>
<tr>
<th>μC(λ) &lt; μothers</th>
<th>μC(λ)</th>
<th>μothers</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>μC(λ) &lt; μF3</td>
<td>0.18</td>
<td>0.21</td>
<td>0.14</td>
</tr>
<tr>
<td>μC(λ) &lt; μF2</td>
<td>0.18</td>
<td>0.20</td>
<td>0.18</td>
</tr>
<tr>
<td>μC(λ) &lt; μF1</td>
<td>0.18</td>
<td>0.21</td>
<td>0.07</td>
</tr>
<tr>
<td>μC(λ) &lt; μL(λ)</td>
<td>0.18</td>
<td>0.19</td>
<td>0.34</td>
</tr>
<tr>
<td>μC(λ) &lt; μλ=0.9</td>
<td>0.18</td>
<td>0.19</td>
<td>0.36</td>
</tr>
<tr>
<td>μC(λ) &lt; μλ=0.7</td>
<td>0.18</td>
<td>0.19</td>
<td>0.34</td>
</tr>
<tr>
<td>μC(λ) &lt; μλ=0.5</td>
<td>0.18</td>
<td>0.19</td>
<td>0.33</td>
</tr>
<tr>
<td>μC(λ) &lt; μλ=0.3</td>
<td>0.18</td>
<td>0.19</td>
<td>0.34</td>
</tr>
</tbody>
</table>

different parameters. From the results, we observe that the median of Custom(λ) (0.162) is less than the medians of all other approaches. From the one-tailed t-test results shown in Table 3.3, we found that the mean of Custom(λ) is not significantly lower than the means of other approaches. The overall results suggest that although Hypothesis H1 is supported on descriptive grounds (apparent differences in MAE distributions), it cannot be asserted based on statistical significance tests.

Please note that, in general, statistical significance can be improved by using larger subject populations. However, additional studies with relaxed assumptions may lead to superior results: a practical challenge there being to effectively obtain the requisite judgments from subjects.

![Figure 3.3: MAE for predicting trust values.](image)

[Verifying H2] For verifying H2, first, we determined customized window sizes (CW=1, 2,
Table 3.4: Statistical test results for $H_2$.

<table>
<thead>
<tr>
<th>$\mu_{C(\lambda)+CW} &lt; \mu_{\text{others}}$</th>
<th>$\mu_{C(\lambda)+CW}$</th>
<th>$\mu_{\text{others}}$</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{\lambda=0.7+CW=1}$</td>
<td>0.17</td>
<td>0.27</td>
<td>0.00</td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{\lambda=0.9+CW=12}$</td>
<td>0.17</td>
<td>0.20</td>
<td>0.08</td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{C(\lambda)}$</td>
<td>0.17</td>
<td>0.18</td>
<td>0.27</td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{L(\lambda)+CW}$</td>
<td>0.17</td>
<td>0.18</td>
<td>0.32</td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{\lambda=0.9+CW}$</td>
<td>0.17</td>
<td>0.19</td>
<td>0.17</td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{\lambda=0.7+CW}$</td>
<td>0.17</td>
<td>0.18</td>
<td>0.30</td>
</tr>
</tbody>
</table>

... for each subject for all the values of $\lambda$ (0.1, ..., 0.9, learned $\lambda$). A customized window size (CW) for a subject refers to the value of CW for which the MAE is minimum. We found that if we increase the window size further (i.e., greater than 12) the MAEs for the subjects do not decrease. We obtained MAEs for all the subjects based on various values of $\lambda$ and CWs, and represent them as Custom($\lambda$)+CW shown in Figure 3.4. We compared Custom($\lambda$)+CW with Custom($\lambda$) obtained from $H_1$ and MAEs obtained from other approaches. From the results, we found that the median of Custom($\lambda$)+CW (0.153) is less than the median of Custom($\lambda$) (0.162) and other approaches. From the one-tailed t-test results shown in Table 3.4, we found that the mean of Custom($\lambda$)+CW is significantly lower than the means of approaches that consider fixed configurations (e.g., $\lambda=0.7+CW$). However, for others, the t-test results show that the differences are not significant. As for $H_1$, although the descriptive results support Hypothesis $H_2$, it cannot be asserted based on statistical tests at the desired level of significance.

![Figure 3.4: Results comparing different window sizes to predict trust values.](image)

[Verifying $H_3$] From the first phase of our experiment, we obtained subjects’ assessments
Table 3.5: Ordering among the feature values obtained from subjects’ assessments.

<table>
<thead>
<tr>
<th>Features</th>
<th>Ordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Debtor’s type</td>
<td>“I” &gt; “We”</td>
</tr>
<tr>
<td>3. Creditor’s type</td>
<td>multiple creditor &gt; single creditor</td>
</tr>
<tr>
<td>4. Modal Verbs</td>
<td>“must, will” &gt; “would, should, shall” &gt; “can, could” &gt; “may”</td>
</tr>
<tr>
<td>5. Deadlines</td>
<td>messages with “deadline” &gt; messages without “deadline”</td>
</tr>
</tbody>
</table>

Of the weights of commitments. From their assessment we obtained different orderings among the feature values shown in Table 3.5. The ordering among the values for the creditor type shows that our initial assumption about it was incorrect. For the rest of the feature values our assumptions correctly aligned with subjects’ assessments.

From the orderings shown in Table 3.5 we calculated a weight for each commitment (CWT). Based on the weights, we recalculated trust parameters using different $\lambda$s, namely, $\lambda = 0.1, \ldots, 0.9$, learned $\lambda$. For $H_3$, we calculated a customized $\lambda$ (Custom($\lambda$)+CWT) and compared the results with $C(\lambda)$ from $H_1$. From the results shown in Figure 3.5, we found that the median of Custom($\lambda$)+CWT (0.161) is slightly less than the median of Custom($\lambda$) (0.162). CWT in the figure means considering commitment weight. From the one-tailed t-test results shown in Table 3.6, the mean of Custom($\lambda$)+CWT is not significantly lower than the means of other approaches. That is, although Hypothesis $H_3$ is supported on descriptive grounds, it cannot be asserted based on the statistical tests at the desired level of significance.

![Figure 3.5: Incorporating commitment weights reduces MAE.](image-url)
Table 3.6: Statistical test results for H₃.

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Mean</th>
<th>Mean</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_{C(\lambda)+CWT} &lt; \mu_{others}$</td>
<td>0.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{C(\lambda)}$</td>
<td>0.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{L(\lambda)+CWT}$</td>
<td>0.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{\lambda=0.9+CWT}$</td>
<td>0.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{\lambda=0.7+CWT}$</td>
<td>0.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu_{C(\lambda)+CW} &lt; \mu_{\lambda=0.5+CWT}$</td>
<td>0.18</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.6 Discussion

Our main contribution is a computational approach for trust that overlays a domain-independent concept describing the social relationships and outcomes of interpersonal interactions. Previous theoretical approaches, both cognitive [12] and architectural [76, 14], have considered rich concepts but they are not easy to use as bases for computing trust in the field. By contrast, previous computational approaches have largely worked in an ad hoc manner that binds the trust reasoning to a particular domain.

For the email dataset, comparing the means of the MAEs, our approach yields a correlation between subjects’ intuitions regarding trust values and those computationally predicted values. Discounting windows customized for each subject yield improved predictions. Considering commitment weights improves predictions further, though not significantly. The limitation of our results may be due to the following reasons: (1) lack of adequate data and (2) a greater fraction of experiences being judged neutral than positive or negative. Also, we lack an existing approach with which to compare our results.

However, we submit that our contribution is valuable for having launched a new research direction on computational techniques unifying trust and commitments. Publishing imperfect results, as in this submission, might serve as an antidote to a systematic bias in academic research to favor “success stories” over accurate reporting of empirical results, a bias that is increasingly decried in the scientific disciplines, e.g., [44, 69].

3.6.1 Related Work

Burnett and Oren [10] examine the effects of delegation using a probabilistic trust model [38] and propose an approach for weighting trust updates based on shared responsibility. They evaluate their approach using different weighting strategies. We restrict our trust update to delegation chains of length three (the debtor, the new debtor and the creditor). This means if the new debtor delegates the commitment to another debtor (debtor'), trust between the debtor and the new debtor remains unaffected. In Burnett and Oren’s work there is no restriction to
the delegation chain length. However, finding longer chains in a text corpus is rare.

Adali et al. [1] emphasize on different behavioral features such as reciprocity, assortativity, attention, and latency to capture relationships between people whereas we emphasize on commitments created and discharged between people based on the text content exchanged between them.

Scissors et al. [71] performed an empirical evaluation with 62 students and found that different forms of linguistic similarity such as content (e.g., positive emotion words, task-related words), structure (e.g., verb tense, phrasal entrainment), and style (e.g., chat abbreviations, emoticons) reflect different levels of trust between participants. In our work instead of linguistic similarity we consider commitments created between participants to calculate trust between them.

3.6.2 Future Directions

First, our dataset is not large. A challenge we faced was motivating subjects to provide trust values truthfully for a larger dataset. Second, our work is limited to predicting trust updates and ignores certainty, which is crucial to trust but difficult to elicit from subjects. In the future, we plan to address these limitations by adopting an incentive scheme that motivates subjects to provide trust values truthfully. Third, there is no reason to be limited to commitments: indeed, we have begun work on bringing in cognitive aspects such as goals and emotions, suitably elicited from subjects, as a basis for creating commitments and judging commitment outcomes and overall trust.
Chapter 4

A Bayesian Model of Trust, Goals, Moods, Commitments, and Emotions

4.1 Introduction

An agent’s moods, trust, and emotions affect the way the agent interacts with another agent and, specifically, how the agent makes decisions in the interactive context. Depending on whether an agent’s goal is achieved because of a commitment by another agent, the agent’s mood, trust, and emotions may be affected.

For example, consider two agents, Alice and Bob, as shown in Figure 4.1. Alice has a goal to complete a task but she cannot complete it by herself. She asks Bob to complete the task. (She may additionally commit to Bob to pay him for the completed task, though this does not feature in our scenario.) Bob agrees, meaning he commits to Alice to perform the specified task. Now consider two main possibilities. First, as in Figure 4.1 (a), suppose Bob completes the task thereby satisfying his commitment. Now Alice achieves her goal. We may expect that Alice’s mood becomes more positive, her trust in Bob increases, and she displays a positive emotion to Bob. Additionally, Bob’s mood may be influenced by Alice’s positive emotions and satisfaction towards Bob’s commitment. This relationship may encourage Bob and Alice to continue to interact in the future. Conversely, as shown in Figure 4.1(b), Bob may fail to complete the task he committed to perform, thereby violating his commitment to Alice. We would expect to observe negative moods and emotions in this case, and they may be discouraged from interacting in the future.

This example suggests that causal relationships can arise between moods, trust, emotions, goals, and commitments. Understanding these relationships is critical in developing frameworks
that provide insights on patterns of interactions among agents and how the agents make decisions in organizational settings, which can significantly affect team performance. The causal relationships among these concepts, however, have not been studied in the literature [51, 18].

Figure 4.1: Two possible enactments in a multiagent scenario.

In the literature, there have been theoretical studies formalizing the relationships among these concepts, but no empirical studies have been conducted. This work is the first to study the relationships of those concepts based on an empirical evaluation. We find the following common relationships of the five concepts above in the literature: (1) positive emotions (e.g., happiness, gratitude) increase trust, whereas negative emotions (e.g., anger) decrease trust [24, 2]; (2) positive emotions (e.g., gratitude) trigger positive moods [73]; (3) a trustor’s trust in a trustee refers to the trustor’s expectation in the trustee’s commitment towards the trustor [76, 11]; (4) positive or negative moods (e.g., joy) and emotions (e.g., gratitude) are closely related to goal completion or failure respectively [79, 33]; and (5) an agent’s achieved goal may be the consequence of another agent’s commitment towards the agent [82].

The existing work has the following limitations:

- The existing work has not investigated the relationships between the five concepts introduced above and not clearly explained how they may affect each other.

- The existing work has not considered dynamically changing characteristics of relationships.

Relationships can be static or dynamic. A static relationship describes the relationship at
a single time frame whereas a dynamic relationship means the relationship over multiple time frames which can explain evolving relationships upon different interactions over time.

In our work, we conduct an empirical study to answer the following questions:

- Are there any causal relationships between trust, moods, emotions, goals, and commitments? If so, what relationships exist between them?
- Are such causal relationships static or dynamic?

We introduce different Bayesian models that build on existing studies linking moods, trust, emotions, commitments, and goals. We adopt a simpler form of appraisal theory [4, 49] assuming that agents process their moods and emotions by appraising the state changes of their goals and commitments. Specifically, we assume that moods and emotions do not appear continuously but are captured at discrete points that indicate the state changes of goals and commitments. We also adopt a simpler form of the dimensional theory [67], where we consider valence for stating moods and emotions but not arousal.

We use the Colored Trails game by Gal et al. [30] as a decision-making platform in our empirical study. We collect data about subjects’ commitments to each other; outcomes of their commitments; their emotions as displayed to others; their moods and trust in their opponents. We train and evaluate Bayesian models that reflect causal relationships.

In our empirical study, we found the following insightful results: (1) an agent’s trust is mostly influenced by its past trust; (2) its mood is mostly affected by its past goal outcomes; (3) an agent’s expectations towards others’ commitments tend to be affected by the outcome of its past commitments toward others (i.e., if an agent is more likely to commit others’ request, it also expects a same level of commitment from others); (4) an agent’s decision on whether to satisfy its commitments toward others is more likely affected by its trust in others and the outcome of others’ commitments toward it; (5) an agent’s emotion is influenced by the outcome of others’ commitments toward it; and (6) an agent’s goal outcomes is influenced by others’ commitments toward it. Overall the key finding is that positive cycles of relationships between these concepts are reinforced by positive past experience, representing reciprocal relationships, and they can be triggered by initial positive actions taken such as commitment and goal completion.

4.2 Technical Approach

We treat goals, commitments, trust, moods, and emotions as discrete random variables in our Bayesian model.

**Goal G_A.** A goal is a private condition that an agent wants to achieve. A goal motivates an agent to act but is not directly visible to others. A goal G_A can have a binary value, *achieved* (ach) or *failed* (fai).
Commitment $C_{A,B}$. A commitment $C_{A,B}(r,u)$ means that a debtor $A$ commits a creditor $B$ to bring about a consequent $u$ provided an antecedent $r$ holds. A commitment provides grounds for $B$ to expect some actions from $A$ [75]. The outcome of a commitment can be represented as: satisfied (sat) when $u$ holds regardless of whether $r$ does; or violated (vio) when $r$ holds but $u$ fails to hold.

Trust $T_{A,B}$. $A$’s trust in $B$ refers to $A$’s expectation in $B$ to bring about the specified condition, i.e., to satisfy a commitment [11]. We consider the variable $T_{A,B}$ with three possible values: low, medium, or high.

Emotion $E_{A,B}$. An emotion is a (transient) response of an agent to a significant external or an internal event [29, 77]. Considering two agents, $A$ and $B$, $A$ can display its emotion to $B$ about a certain event, notating it as a variable $E_{A,B}$ with two possible values: positive (pos) or negative (neg).

Mood $M_A$. A mood is a low-intensity, long-lasting condition of feeling good or bad [29, 27]. For example, a positive event occurring in the morning may leave someone in a good mood for the entire day. A mood is felt, but it is not necessarily directly displayed. We represent $A$’s mood as $M_A$ with three possible values: negative (neg), neutral (neu), or positive (pos).

4.2.1 Baseline Models

First of all, to understand the relationships between the five concepts, we adopt several different Bayesian models from the existing works as baseline models considering the assumptions used in this work. Second, we check if these models hold for a dataset derived from the observations in our empirical study. If they hold, we consider them for evaluation. In addition, we can learn new models from the data. Finally, we evaluate both the baseline and the new models based on prediction accuracy. To describe baseline models, we consider two agents Alice ($A$) and Bob ($B$) in an interactive context, and target nodes are defined for $B$ in the models. Similar nodes can be defined for $A$ as well. The baseline model is described as follows.

- $B$’s current (curr) trust in $A$: $[M_1]$ outcomes of $A$’s past commitments toward $B$ influence $B$’s current trust in $A$ ($C_{A,B}^{past} \rightarrow T_{B,A}^{curr}$); $[M_2]$ $B$’s past moods influence $B$’s current trust in $A$ ($M_{B}^{past} \rightarrow T_{B,A}^{curr}$); $[M_3]$ outcomes of $B$’s past goals influence $B$’s current trust in $A$ ($G_{B}^{past} \rightarrow T_{B,A}^{curr}$); $[M_4]$ emotions expressed by $A$ toward $B$ in past influences $B$’s current trust in $A$ ($E_{A,B}^{past} \rightarrow T_{B,A}^{curr}$).

- $B$’s current moods: $[M_5]$ outcomes of $B$’s past goals influence $B$’s current moods ($G_{B}^{past} \rightarrow M_{B}^{curr}$); $[M_6]$ outcomes of $A$’s past commitments toward $B$ influence $B$’s current moods ($C_{A,B}^{past} \rightarrow M_{B}^{curr}$); $[M_7]$ past emotions expressed by $A$ toward $B$ influence $B$’s current moods ($E_{A,B}^{past} \rightarrow M_{B}^{curr}$).
• B’s current expectations about A satisfying its commitments toward B: [M_7] outcomes of B’s past commitments toward A influence outcomes of A’s current commitments toward B (C_{B,A}^{past} \rightarrow C_{A,B}^{curr}); [M_8] B’s past trust in A influence outcomes of A’s current commitments toward B (T_{B,A}^{past} \rightarrow C_{A,B}^{curr}).

• B’s current decisions to satisfy its commitments toward A: [M_9] outcomes of A’s past commitments toward B influence outcomes of B’s current commitments toward A (C_{A,B}^{past} \rightarrow C_{B,A}^{curr}); [M_{10}] B’s past trust in A influence B’s current commitments toward A (T_{B,A}^{past} \rightarrow C_{B,A}^{curr}); [M_{11}] B’s past moods influence B’s current commitments toward A (M_{B}^{past} \rightarrow C_{B,A}^{curr}).

• B’s current emotions expressed toward A: [M_{12}] outcomes of A’s past commitments toward B influence B’s current emotions toward A (C_{A,B}^{past} \rightarrow E_{B,A}^{curr}); [M_{13}] B’s past trust in A influence B’s current emotions toward A (T_{B,A}^{past} \rightarrow E_{B,A}^{curr}); [M_{14}] B’s past moods influence B’s current emotions expressed toward A (M_{B}^{past} \rightarrow E_{B,A}^{curr}).

• Outcomes of B’s current goals: [M_{15}] outcomes of A’s commitments toward B influence outcomes of B’s current goals (C_{A,B}^{past} \rightarrow G_{B}^{curr}); [M_{16}] B’s past emotions expressed toward A influence outcomes of B’s goals (E_{B,A} \rightarrow G_{B}).

The metrics we use to check if different Bayesian models hold for data are Log-Likelihood (LL), Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC) scores, which capture the goodness of fit of models to the data. The metric we use to evaluate model accuracy is the area under receiver operating characteristic (ROC) curve (AUC). The curve describes a classifier and it is constructed by plotting sensitivity versus specificity. We obtain an AUC score for each model by averaging three AUC scores obtained by performing three-fold cross-validation respectively on the data for each model. Sometimes it may happen that the data is inadequate and, hence, three-fold validation cannot be performed. In such cases we consider the AUC score obtained from partial cross-validation (less than three). Also sometimes it may happen that we cannot calculate the AUC score at all. In such cases we select models based on LL, AIC, and BIC scores. We perform the one-tailed t-test on different scores obtained to record the significance level at 5%.

4.2.2 Empirical Study Design

To ensure our work empirically grounded, we develop a variant (Figure 4.2) of Gal et al.’s [30] Colored Trails game. Our variant provides a chat interface through which subjects negotiate and exchange tiles and express emotions toward opponents. We map our variables to data gathered from the game play, text chats, and subject surveys. Each game involves two subjects, Alice (A) and Bob (B).
**Rules of the Game.** (1) A subject is required to play three games with different opponents. (2) Each game consists of five rounds. (3) Each round has a common goal position, and different starting positions for each subject. (4) In all rounds, subjects are allocated the same number but a different set (randomly selected) of colored tiles. (5) Subjects can communicate with their opponents via a chat interface, in which they can negotiate to transfer tiles to each other. (6) At the beginning of a game and at the end of each round, each subject fills a survey. (7) Let $n$ be the number of tiles left unused and $u$ be the number of tiles left used. The scoring function for each subject is defined as follows: (a) the score for a subject who does not reach the goal $= n + 1.5 \times u$ and (b) the score for a subject who reaches the goal $= n + 3.0 \times u$.

Now we describe the mapping between the game and concepts for $B$. The mapping for $A$ is the same.

- **Goal.** In the game, we assume that the goal of $B$ is to reach the goal position.

- **Commitment.** During the game, $A$ can create a commitment by agreeing (through chat) to transfer a specified number of tiles to $B$. If $A$ provides the tiles, $A$ satisfies its commitment towards $B$, otherwise $A$ violates it.

- **Emotion.** $B$ may express an emotion toward $A$ via a chat message. We determine whether the emotion is positive or negative by manually analyzing the text.

- **Mood.** $B$’s mood changes as the games progress and may spill over from one game to the next. We determine $B$’s mood from $B$’s survey answers.
• Trust. We determine B’s trust in A from B’s survey answers.

Subjects. We advertised the study to recently admitted (to minimize the threat of prior knowledge of our study) graduate students in Computer Science. We offered a payment of 10–20 USD each, depending on success in the game. We selected 30 subjects (25 male; 5 female) on first-come basis.

Surveys. (1) At the start of each game and at the end of each round, we asked subjects to record their trust for opponents and mood on a five-point scale (very negative, negative, neutral, positive, very positive). We mapped the responses to our trust ($T_{A,B}$) and mood ($M_A$) variables respectively. For analysis, we converted the above five-point scale into a three-point scale by merging very negative and negative responses and very positive and positive responses.

Figure 4.3: Timeline of our data acquisition.

4.2.3 Data Description

We collected 450 rows of data (30 subjects $\times$ 30 games $\times$ 5 rounds each game), including their survey forms, and whatever chat messages they exchanged. From their interactions, we manually analyzed messages to identify commitment ($C_{B,A}$) outcomes and emotions ($E_{A,B}$) displayed. For commitments and emotions, we added subject’s moods ($M_A$), trust ($T_{A,B}$), and goals ($G_A$). Table 4.1 shows the data distribution for each concept along with the missing data.

Table 4.1: Distributions of values of variables in the data.

<table>
<thead>
<tr>
<th>var</th>
<th>value1</th>
<th>value2</th>
<th>value3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commitment</td>
<td>sat (119)</td>
<td>vio (16)</td>
<td></td>
</tr>
<tr>
<td>Goal</td>
<td>ach (231)</td>
<td>fai (219)</td>
<td></td>
</tr>
<tr>
<td>Trust</td>
<td>low (95)</td>
<td>med (224)</td>
<td>high (131)</td>
</tr>
<tr>
<td>Mood</td>
<td>neg (85)</td>
<td>neu (183)</td>
<td>pos (182)</td>
</tr>
<tr>
<td>Emotion</td>
<td>pos (21)</td>
<td>neg (1)</td>
<td></td>
</tr>
</tbody>
</table>
4.2.4 Constructing a Bayesian Model

Background. A Bayesian network [62] is a directed acyclic graph (DAG) in which the nodes represent random discrete variables and edges represent the direct influence of one node on another. A Bayesian model can be represented as a joint probability distribution \( P(X_1, X_2, \ldots, X_n) = \prod_i P(X_i|\text{Parents}(X_i)) \) where \( X_1, X_2, \ldots, X_n \) are the random variables. In a Static Bayesian Network (SBN) the random variables are observed in a single time slice whereas in a Dynamic Bayesian Network (DBN) random variables are observed across multiple time slices with variables in one time slice being influenced by variables in another time slice.

Random Variables. We created a DBN with two time slices, \( t-1 \) and \( t \). To do so, we mapped each round in a game to a time slice. For example, we mapped time slice \( t-1 \) to Round 1 and the next time slice \( t \) to Round 2. In each round, for each player, we observed seven discrete random variables. As shown in Figure 4.3, first, \( B \) records its trust in \( A \) \( T_{B,A} \) and moods \( M_B \). Then, in the game, in each round, the following variables were observed for \( B \): outcomes of \( A \)'s commitments toward \( B \) \( C_{A,B} \), emotions expressed by \( B \) toward \( A \) \( E_{B,A} \), outcomes of \( B \)'s commitments toward \( A \) \( C_{B,A} \), emotions expressed by \( A \) toward \( B \) \( E_{A,B} \), emotions expressed by \( B \) toward \( A \) \( E_{B,A} \), and finally outcomes of \( B \)'s goals \( G_B \).

Correlations. Before we created a DBN, we determined Pearson’s correlation coefficients \( R \) between the random variables observed during the game. Table 4.2 shows the correlations between variables observed in one time slice whereas Table 4.3 shows the correlations between variables observed in time slice \( t-1 \) and the variables observed in time slice \( t \).

Observations. From the correlation coefficients we obtained the overall relationships between random variables. We only considered positive correlations. In Tables 4.2 and 4.3, positive correlations vary from 0.01 to 0.6 with most of the important correlations based on our hypotheses lie on or above 0.1. Therefore, if \( R \) is greater than or equal to 0.1, we considered that the relationships exists. An \( R \) of 0.1 may indicate a weak correlation. However, such correlations are common in psychology [37, 13, 72]. Based on the observed correlations, we created incremental Bayesian models for trust, moods, commitments, emotions, and goals by adding one node at a time as input to a target node \( (T_{B,A}^2, M_B^2, C_{A,B}^2, C_{B,A}^2, G_B^2) \). For each such incremental model, we calculated the LL, AIC, and BIC scores—and show them in Figure 4.4. As discussed in the evaluation section, we eliminated some of the relationships from the models based on the significant results obtained via LL, AIC, and BIC scores.

Expectation Maximization. Based on the gathered data, we performed Expectation Maximization (EM) using the Hugin tool [52] to obtain the conditional probabilities and LL, AIC, and BIC scores of the Bayesian models. We initialized the outcomes of a commitment, goal, and emotion uniformly as 0.5 (binary values) and the outcomes of trust, and mood uniformly as 0.33 (ternary values).
### Table 4.2: Correlations between variables observed in one time slice. (1 in superscript indicates Round 1)

<table>
<thead>
<tr>
<th>R</th>
<th>( T_{B,A}^1 )</th>
<th>( M_B^1 )</th>
<th>( C_{A,B}^1 )</th>
<th>( E_{B,A}^1 )</th>
<th>( G_B^1 )</th>
<th>( C_{B,A}^1 )</th>
<th>( E_{A,B}^1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{B,A}^1 )</td>
<td>1</td>
<td>0.4</td>
<td>0.02</td>
<td>0.4</td>
<td>0.1</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>( M_B^1 )</td>
<td>0.4</td>
<td>1</td>
<td>0.05</td>
<td>0.4</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>( C_{A,B}^1 )</td>
<td>0.02</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
<td>0.2</td>
<td>–</td>
<td>NaN</td>
</tr>
<tr>
<td>( E_{B,A}^1 )</td>
<td>0.4</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0.4</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>( G_B^1 )</td>
<td>0.1</td>
<td>–</td>
<td>0.2</td>
<td>0.4</td>
<td>1</td>
<td>0.1</td>
<td>–</td>
</tr>
<tr>
<td>( C_{B,A}^1 )</td>
<td>0.2</td>
<td>–0.1</td>
<td>NaN</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( E_{A,B}^1 )</td>
<td>0.1</td>
<td>–0.2</td>
<td>NaN</td>
<td>NaN</td>
<td>–0.2</td>
<td>1</td>
<td>NaN</td>
</tr>
</tbody>
</table>

### Table 4.3: Correlating variables observed in time slice \( t - 1 \) with variables observed in time slice \( t \). (1 and 2 in superscript indicate Round 1 and 2 respectively)

<table>
<thead>
<tr>
<th>R</th>
<th>( T_{B,A}^2 )</th>
<th>( M_B^2 )</th>
<th>( C_{A,B}^2 )</th>
<th>( E_{B,A}^2 )</th>
<th>( G_B^2 )</th>
<th>( C_{B,A}^2 )</th>
<th>( E_{A,B}^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{B,A}^2 )</td>
<td>0.6</td>
<td>0.3</td>
<td>0.3</td>
<td>0</td>
<td>0.04</td>
<td>0.2</td>
<td>NaN</td>
</tr>
<tr>
<td>( M_B^2 )</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
<td>0</td>
<td>0.01</td>
<td>0.1</td>
<td>NaN</td>
</tr>
<tr>
<td>( C_{A,B}^2 )</td>
<td>0.2</td>
<td>0.1</td>
<td>0.4</td>
<td>0</td>
<td>–</td>
<td>0.3</td>
<td>NaN</td>
</tr>
<tr>
<td>( E_{B,A}^2 )</td>
<td>0.5</td>
<td>–</td>
<td>0</td>
<td>NaN</td>
<td>0.2</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>( G_B^2 )</td>
<td>0.4</td>
<td>0.5</td>
<td>–</td>
<td>NaN</td>
<td>–0.1</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>( C_{B,A}^2 )</td>
<td>0.2</td>
<td>0.1</td>
<td>0.3</td>
<td>NaN</td>
<td>0.1</td>
<td>0.2</td>
<td>NaN</td>
</tr>
<tr>
<td>( E_{A,B}^2 )</td>
<td>0.5</td>
<td>0.5</td>
<td>NaN</td>
<td>NaN</td>
<td>-0.2</td>
<td>NaN</td>
<td>NaN</td>
</tr>
</tbody>
</table>

### 4.3 Evaluation

**Trust.** To verify if our baseline models hold for trust, we created multiple Bayesian models by incrementing one input node to \( T_{B,A}^2 \) at each time based on correlations. For each such model, we calculated the LL, AIC, and BIC scores. We found that when trust \( T_{B,A}^1 \), moods \( M_B^1 \) and commitments \( C_{A,B}^1 \) were incorporated in the model, the changes in the scores, shown in Figure 4.4(a), were significant with p-values of 0.00, 0.00, and 0.03, respectively. This means that baselines \( M_1: C_{A,B}^1 \rightarrow T_{B,A}^2 \) and \( M_2: M_B^1 \rightarrow T_{B,A}^2 \) hold whereas \( M_3: G_B^1 \rightarrow T_{B,A}^2 \) and \( M_4: E_{A,B}^1 \rightarrow T_{B,A}^2 \) do not hold. In addition, we learned a new model \( M_{17}: T_{B,A}^1 \rightarrow T_{B,A}^2 \). To evaluate \( M_1, M_2, \) and \( M_{17} \), we calculated the AUC score for each model. We found that the AUC score for \( M_{17} (0.74) \) was significantly greater than the AUC score for \( M_2 (0.63) \) (with a p-value of 0.00) and the AUC score for \( M_2 \) was significantly greater than \( M_1 (0.47) \) (with a p-value of
Therefore, we specify the order between models as \( M_{17} > M_2 > M_1 \).

**Mood.** To verify if our baseline models hold for moods, we created incremental models of moods based on correlations. From the LL, AIC, and BIC scores for each such model, we found that when trust \( T_{B,A} \), moods \( M_B \), commitments \( C_{A,B} \), and goals \( G_B \) were incorporated in the model, the changes in scores, shown in Figure 4.4(b) were significant (with p-values of 0.00, 0.00, 0.03, and 0.02, respectively). This means that baselines \( M_5 \): \( G_B \rightarrow M_B \) and \( M_6 \): \( C_{A,B} \rightarrow M_B^2 \) hold whereas \( M_7 \): \( E_{A,B} \rightarrow M_B^2 \) does not hold. In addition, we learned two new models \( M_{18} \): \( M_B^1 \rightarrow M_B^2 \) and \( M_{19} \): \( T_{B,A} \rightarrow M_B^2 \). Comparing \( M_5 \), \( M_6 \), \( M_{18} \), and \( M_{19} \) based on AUC scores, we found that the AUC score for \( M_5 \) (0.69) was significantly greater than \( M_{19} \) (0.59), \( M_{18} \) (0.56), and \( M_6 \) (0.5) (with p-values of 0.04, 0.03, and 0.003, respectively). Similarly, the AUC score for \( M_{19} \) was significantly greater than \( M_6 \) (with a p-value of 0.04) but not significantly greater than \( M_{18} \) (with a p-value of 0.25). The AUC score for \( M_{18} \) was slightly significant than \( M_6 \) (with a p-value of 0.06). Therefore, we specify the order between models as \( M_5 > (M_{19} = M_{18}) > M_6 \).

**Commitments.** To verify if baselines hold for a player’s expectations about its opponent satisfying commitments toward the player, we created incremental models for \( C_{A,B} \) based on correlations. From the LL, AIC, and BIC scores for each such model, we found that when \( T_{B,A}^1 \), \( C_{A,B}^1 \), and \( C_{B,A}^1 \) were incorporated in the model, the changes in scores, shown in Figure 4.4(c), were significant (with p-values of 0.00, 0.00, and 0.00, respectively). This means that baselines \( M_7 \): \( C_{B,A}^1 \rightarrow C_{A,B}^2 \) and \( M_8 \): \( T_{B,A} \rightarrow C_{A,B}^2 \) hold. In additional, we learned a new model \( M_{20} \): \( C_{A,B}^1 \rightarrow C_{A,B}^2 \). We computed the AUC scores for three models \( M_7 \), \( M_8 \), and \( M_{20} \). We found that the AUC score for \( M_7 \) (0.71) was greater than \( M_8 \) (0.69) and \( M_{20} \) (0.68). The t-test results were not measured on AUC scores because it is not possible to perform three-fold cross-validation for each model because of the data for \( M_{20} \) and \( M_7 \) being inadequate. From partial AUC scores, we specify the order between models as \( M_7 > (M_8 = M_{20}) \).

To verify if baselines hold for a player’s decision to satisfy its commitments, we created incremental models for \( C_{B,A}^2 \) based on correlations. From the LL, AIC, and BIC scores, we found that when \( T_{B,A}^1 \), \( C_{A,B}^1 \), \( C_{B,A}^1 \), and \( T_{B,A}^2 \) were incorporated into the model, the changes in scores, shown in Figure 4.4(d), were significant (with p-values of 0.00, 0.00, 0.01, and 0.00, respectively). This means that \( M_9 \): \( C_{A,B}^1 \rightarrow C_{B,A}^2 \) and \( M_{10} \): \( T_{B,A} \rightarrow C_{B,A}^2 \) hold whereas \( M_{11} \): \( M_B^1 \rightarrow C_{B,A}^2 \) does not hold. In addition, we learned two new models \( M_{21} \): \( T_{B,A}^2 \rightarrow C_{B,A}^2 \) and \( M_{22} \): \( C_{B,A}^1 \rightarrow C_{B,A}^2 \). Comparing \( M_9 \), \( M_{10} \), and \( M_{21} \), we found that the AUC score for \( M_9 \) (0.71) and \( M_{21} \) (0.71) were greater than \( M_{10} \) (0.56) and \( M_{22} \) (0.68). Again, the t-test is not performed due to inadequate data for models \( M_9 \), \( M_{21} \), and \( M_{22} \). Based on partial AUC scores, we specify the order between models as \( (M_9 = M_{21}) > M_{22} > M_{10} \).

**Emotions.** To verify if baselines hold for emotions, we created incremental models for emotions based on correlations. From the LL, AIC, BIC scores for each such model, we found that when \( T_{B,A}^2 \), \( M_B^2 \) and \( C_{A,B}^2 \) were incorporated into the model, the changes in scores were
significant (with p-values of 0.00 and 0.01, respectively). This means that baselines do not hold. However, we learned two new models M23: $T_{B,A}^2 \rightarrow E_{B,A}^2$, M24: $M_B^2 \rightarrow E_{B,A}^2$, and M25: $C_{A,B}^2 \rightarrow E_{B,A}^2$. The AUC scores were not computed due to inadequate data for emotions. Based on the LL, AIC, and BIC scores, we specify the order between models as $M_{23} > M_{24} > M_{25}$.

**Goals.** To verify if the baselines hold for goals, we created incremental models for goals based on correlations. From the LL, AIC, BIC scores for each such model, we found that when $C_{A,B}^2$ was added to the model, the change in scores for $C_{A,B}^2$ was significant (with a p-value of 0.01). This means none of the baseline holds. The new model we got was M26: $C_{A,B}^2 \rightarrow G_B^2$. The
average of AUC scores for M_{26} was 0.56.

### 4.4 Discussion and Conclusion

We proposed Bayesian models to describe relationships between commitments, goals, trust, moods, and emotions between agents. We conducted an empirical study to obtain data from human subjects. We evaluated our hypotheses via LL, AIC, BIC, and AUC scores. Our results indicated the following: (1) a player’s current trust is mostly influenced by the player’s past trust. In addition, it is influenced by the player’s past mood and the outcome of opponents’ past commitments created toward the player; (2) a player’s current mood is mostly influenced by the outcome of the player’s past goals. It is also influenced by the outcome of opponents’ past commitments created toward the player, the player’s past trust, and mood; (3) a player’s current expectations about the outcome of opponents’ commitments toward the player is influenced by the outcome of the player’s past commitments toward opponents; (4) a player’s current decision to satisfy its commitments is mostly affected by the player’s current trust and the outcome of opponents’ past commitments created toward the player. In addition, it is influenced by the player’s past commitment outcomes; (5) a player’s current emotions is influenced by the outcome of its opponents’ commitments; (6) a player’s current goal outcomes is influenced by the outcome of opponents’ past commitments created toward the player.

#### 4.4.1 Related Work

De Melo et al. [21] propose a Bayesian model that captures an appraisal-based mechanism for the interpersonal affect of emotion in decision-making. Their work relies solely on emotion displays of an agent collected from past interactions to predict whether the agent will cooperate in the future or not. Our work extends their approach by considering trust and moods along with emotions collected from an agent. Our work additionally establishes general relationships between emotions, trust, and moods.

Antos et al. [2] conduct a series of negotiation games between humans and computational agents, followed by a trust game where humans choose an agent from among several agents to entrust a fraction of their profit earned from the negotiation games to the agent chosen. Unlike their work where agents only express emotions, we built a model that considers both emotions and trust. Using our model, an agent can choose different strategies based on its emotions and trust for an agent.

Antos and Pfeffer [3] provide a methodology for decision-making by agents that leverages a computational concept of emotions. Their results indicate that emotion-based agents outperform other reasonable heuristics and perform closely approaching to near-optimal solutions.
which often require high computational complexity. In contrast, we consider trust and moods to emotions to support agent in decision-making.

Tanguy et al. [80] provide a model Dynamic Emotion Representation (DER) that integrates emotional responses and keeps track of changing emotion intensities over time. Unlike their work, our work deals with multiple emotional processes.

Dunn and Schweitzer [24] conduct experiments to describe the influence of emotional states on trust. They found that emotions with positive valence increase trust whereas emotions with negative valence decrease trust. In contrast, our work considers both the static and dynamic relationship between emotions and trust.

4.4.2 Threats to Validity and Future Work

The topics we studied are complex in social psychology. Inevitably, an empirical study such as ours faces some threats to validity. Our subject population of computer scientists is not representative of typical users though we mitigated this threat by selecting subjects who were not well-versed in the research area studied. The artificial setting of a game would tend to have lower stakes than many real-life interactions and may thus elicit limited emotional responses. To control our experiment, we prevented subjects from (1) knowing their opponent; and (2) projecting or receiving any visual or auditory signals. In real-life, past relationships are significant as is nonverbal communication. Thus, our findings may not propagate well to real-life settings.

In future work, we plan to address the above threats. On the theoretical side, we plan to complete the picture with regard to decision-making, especially accounting for variations in the intensity of mood, trust, and emotions. On the practical side, we plan to exploit and enhance emerging text analysis techniques, e.g., Kalia et al.’s [?] approach for extracting commitments from chats, to help automate the extraction of commitments and emotions, and facilitate studies at a larger scale.
Chapter 5

Determining Team Hierarchy

5.1 Introduction

In an organization, a team is a purposeful social system created to get work done. Therefore, it is important to understand and characterize the degree to which team members coordinate with each other. In most organizations, a team hierarchy exists among the team members wherein a higher ranking team member sets high-level goals, and guides or motivates lower ranking team members, who are expected to carry out such commands. Although team members have clearly delineated roles, it is important to evaluate whether they are performing their jobs well or whether the team needs restructuring. One important factor for evaluating team performance is communication between team members. Eaton [25] provides insight that communication is essential for team members to build their inter-personal relationships which indirectly enhance team performance. Leonard and Frankel [50] describe that for effective teamwork communication is important because it creates predictability and agreement between team members. Resick et al. [64] suggest that information elaboration is important in evolving teams to maintain team performance. Our premise is that we can determine such indicators of organizational effectiveness and team member performance from members’ communications, such as chats and emails, which provide an account of actual behavior while being unobtrusive.

Several works have identified team hierarchies from graphs extracted from online social networks such as Twitter, Flickr, Prison, and Wikivote [35, 57, 58] and text such as emails and short message service (SMS) communications [32, 66, 84]. Gupte et al. [35] and Enys et al. [57, 58] provide hierarchical measures called social agony and global reaching centrality (GRC), respectively, to extract hierarchies from online social networks. Rowe et al. [66] extract an undirected graph from Enron emails [26, 47] based on the number of emails exchanged between Enron employees whereas Wang et al. [84] compute hierarchy from Enron emails as well as from call and SMS data. Gilbert [32] emphasized analyzing text content to extract phrases that
indicate hierarchy. The above works apply when social graphs can be extracted, such as from online social networks and directed messages (emails and SMS). However, these approaches do not apply for broadcast messages, where the receiver is not clear.

Our approach takes in broadcast messages recorded from a multiparty event and produces a team hierarchy among the participants. The basis of our approach is to identify communication patterns from messages that indicate a possible team hierarchy. Broadly, we identify three patterns: directive, question, and informative. We select these patterns based on the existing literature [32, 61] and the fact that they occur frequently in broadcast messages. The overall approach approximates Gilbert [32]. Whereas his approach identifies communication content that indicates power and hierarchy, we additionally compute the ranks and validate our approach versus ground truth. Also, Gilbert’s approach is domain-dependent, whereas our approach is domain-independent and applies to broadcast as well as directed communications.

We analyze semantic, communication, and social features that can be extracted from messages to compute hierarchy. Semantic features include responses to communication patterns and emotions expressed in responses features extracted from text content. Communication features include the average response time delay and messages sent features. Social features include the degree centrality and betweenness centrality features. We hypothesize that semantic features, which capture the meaning of interactions, are better indicators of hierarchy than social features, which merely capture network statistics.

To identify the patterns, we select two chat rooms from a military exercise dataset. We use one chat room to refine our methods to identify patterns and test our method on the second chat room, obtaining an F-measure of 83% for identifying the patterns. From the patterns identified, we collect the features described above. Using these features we determine participants’ ranks computed via hierarchical clustering. We evaluate our results against actual known ranks. In addition, we evaluate the generalizability of our approach to directed communications, as in Enron email corpus. In directed communications, emails exchanged between senders and receivers provide good indicators of hierarchy.

We find that for the chat corpus the accuracy in identifying ranks using the informative pattern is significantly higher than for the directive and question pattern. Additionally, we find that semantic features along with communication features are better indicators of hierarchy than social features. For Enron, we obtain similar results regarding the identification of patterns though we find that social features are better indicators of hierarchy than semantic features, possibly because compared to the military dataset, the Enron corpus is much larger with more participants and messages. And it may be that in such a large corporate organization, the roles, responsibilities, and influence need to be ascertained socially. Also, compared to participants in Enron, participants in military communication networks have well-defined functional roles and prescribed work flows that lead to more structured communication and hence, semantic
features may perform better than social features.

5.2 Communication Patterns in Broadcast Messages

Broadcast messages are sent by participants in a group and hence, everyone in a group can see and respond to messages. Before we infer a hierarchy from broadcast messages it is important to understand what each message means. For example, a message can indicate different illocutions [5] such as directives and commissives. Based on the literature [32, 61] and our preliminary analysis, i.e., manually finding the distributions of meanings of the messages in the military dataset, we hypothesize that hierarchical information can be extracted from messages via three communication patterns: directive, question, and informative. A directive is an order or request; a question is an inquiry; an informative is a report. Directives and questions correlate with the sender having a higher rank than the receiver; informatives the reverse.

An important challenge in dealing with broadcast messages is that the recipient of a message is not clear. To tackle the challenge, we define a window $W$ consisting of two consecutive messages where we assume that the second message $W_{\text{next}}$ is a response to the first message $W_{\text{curr}}$. The two messages must occur in the same chat room and have different senders. A window $W$ is instantiated as a directive, question, or informative pattern if, respectively, $W_{\text{curr}}$ is a directive, question, or informative and correspondingly $W_{\text{next}}$ is an acknowledgment, response, or acknowledgment. Table 5.1 provides examples of these patterns from military data.

<table>
<thead>
<tr>
<th>Window</th>
<th>Sender</th>
<th>Messages</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_{\text{curr}}^a$</td>
<td>8_6i_256_s3</td>
<td>Cos, send all reports up to BN over this net</td>
<td>Directive</td>
</tr>
<tr>
<td>$W_{\text{next}}^a$</td>
<td>8_6i_256_b_cdr</td>
<td>rgr</td>
<td></td>
</tr>
<tr>
<td>$W_{\text{curr}}^b$</td>
<td>8_6i_256_b_cdr</td>
<td>B, what's your status on personnel?</td>
<td>Question</td>
</tr>
<tr>
<td>$W_{\text{next}}^b$</td>
<td>8_6i_256_b_cdr</td>
<td>no casualties</td>
<td></td>
</tr>
<tr>
<td>$W_{\text{curr}}^b$</td>
<td>8_6i_256_b_cdr</td>
<td>have been engaging with SAF and MTRs with no effect</td>
<td>Informative</td>
</tr>
<tr>
<td>$W_{\text{next}}^b$</td>
<td>8_6i_256_cdr</td>
<td>ack, keep me posted</td>
<td></td>
</tr>
</tbody>
</table>
5.3 Process

Figure 5.1 shows the process we follow. In the process, we separately consider the *directive*, *question*, *informative* patterns as well as the combination of *directive* and *question* patterns to compute ranks. Next, we evaluate the accuracy of the ranks computed based on different patterns.

As an illustration, consider computing ranks using *directive* patterns. For each participant $P$ in chat messages we extract the following features. First, we extract *directive* patterns $W$ where $W_{\text{curr}}$ indicates a directive message and $P$ is the sender of $W_{\text{next}}$. From the patterns, we assume that $P$ responds to $W_{\text{curr}}$ and hence, we calculate the total number of such responses to directives for $P$.

Second, we determine whether $W_{\text{next}}$ indicates a positive, neutral, or negative emotion. We extract emotions because we hypothesize that they can be indicators of hierarchy. For example, $P$ may be a team leader and may display positive emotions to motivate subordinates or $P$ may be a subordinate and may express emotions with respect to outcome of his or her actions. We include responses to patterns and emotions within semantic features.

Third, based on the patterns $W$ we find the average response time delay, i.e., the average of the time lags between $W_{\text{curr}}$ and $W_{\text{next}}$ extracted for $P$. Fourth, we find the number of messages that $P$ broadcasts. We include the average response time delay and number of messages broadcast as communication features.

From the patterns $W$ we create a graph that contains directed edges from responders ($P$) to respondees. Using the graph, we compute social features for $P$, i.e., $P$’s degree centrality and betweenness centrality [8, 28]. We aggregate all features—semantic, communication, and social—for $P$. We repeat the feature extraction for all participants $P^*$. Finally, based on $P^*$’s features we compute hierarchical ranks for each $P$. We evaluate computed ranks against the ground truth of actual ranks. We carry out the above process for the informative and question patterns.

Prior works [35, 58, 66] focus primarily on social and communication features to compute ranks whereas we include semantic features based on the intuition that semantic features, being based on the message content, can reveal important hierarchical information. Below, we discuss the extraction of features in detail.

5.3.1 Extracting Semantic Features

To extract semantic features for each participant, first, we identify patterns $W$. To identify patterns, we create a rule-based approach using training data and evaluate it on a test data. Both training and test data consist of broadcast messages labeled *directive*, *question*, or *informative*. To support our rules, for each dataset, we build a domain-specific lexicon of action verbs that
Figure 5.1: Process followed to compute ranks with respect to directive, question, and informative patterns, and directives and questions combined for participants (P*) from broadcast messages.

includes words occurring frequently in the data.

3.1.1 Extracting Responses to Directives

To extract a response to a directive, we determine if a message $W_{\text{curr}}$ in $W$ indicates a directive. To do so, we parse a message $W_{\text{curr}}$ using the Stanford Natural Language Parser [46] and extract a parse tree. Figure 5.2 represents a parse tree for a sample message “Cos, send all reports up to BN over this net.” In the parse tree, first, we look for a verb phrase (VP) indicated by the shading in Figure 5.2. Then, in the VP we look for an action verb (VB). If the action verb matches a verb in our domain-specific lexicon, we extract the rest, i.e., noun (NP) and prepositional phrase (PP), as shown in Figure 5.2. Hence, the words extracted from the example message are “send all reports up to BN over this net” which we identify as a directive. We assume the next message $W_{\text{next}}$ is a response to the directive message.

3.1.2 Extracting Responses to Questions

To extract a response to a question, we determine if a message $W_{\text{curr}}$ in $W$ indicates a question. If a message starts with a word such as what, when, why, has, how, have, and so on and ends with a question mark or if a message starts with a modal verb (MD) such as will, shall, could, would, should, and can followed by the word you, we mark the message as a question. If a message is identified as a question, we assume the next message $W_{\text{next}}$ is a response to the question regardless of its grammar or content.
3.1.3 Extracting Responses to Informative

To extract a response to an informative, we determine if a message $W^{curr}$ in $W$ indicates an informative. If a message begins with the following rgr, Roger, ack, yes, yup, yep, okay, ok, thanks, and so on we tag the message as the informative. Although some of the words (e.g., Roger and ack) are domain-specific, other words (thanks, yes, and okay) are domain independent. Such generic words make this pattern domain-independent. The next message $W^{next}$ we assume is a response to the informative message.

For each participant, we calculate the count of all $W^{next}$ or responses extracted for each pattern.

3.1.4 Extracting Emotions in Responses

For each communication pattern $W$, we determine if the response message $W^{next}$ indicates an emotion, which could be positive, neutral, or negative. We use the Stanford Sentiment Parser [78], which computes the emotion corresponding to a message. For each participant, we compute the sums of the emotion polarities identified from response messages.

5.3.2 Extracting Communication Features

For each participant we extract two communication features. One, the number of messages sent by the participant and second, the average response time delay for a participant based on the messages that indicate responses to a pattern. The number of messages is a network statistic calculated independently of responses to patterns.
5.3.3 Extracting Social Features

To extract social features, we create a graph represented as an adjacency matrix $A_{ij}$. In the matrix, $i$ and $j$ represent the participants. An edge $ij$ in $A$ exists from the sender (responder) of $W_{\text{next}}$ toward the sender (respondee) of $W_{\text{curr}}$, if $W_{\text{next}}$ indicates a response to a pattern, i.e., directive, question, or informative. If an edge $ij$ exists, we mark $A_{i,j} = 1$ else we mark $A_{i,j} = 0$. We also mark $A_{i,j} = 0$ if $i$ equals $j$ because we assume a sender does not respond to itself. We mark $A_{i,j} = 1$ irrespective of one or more responses between $i$ and $j$. From $A_{i,j}$ we can construct a directed graph $G(V, E)$ where $V$ represents the participants and $E$ represents the directed edge between the participants.

Using the directed graph $G(V, E)$ extracted from a pattern, we compute the social features of degree centrality and betweenness centrality. We consider these social features for two reasons. One, they have been used in the literature to interpret Rowe et al.’s [66] hierarchy. Two, we consider chatrooms that contain more intrateam messages than interteam messages, possibly, because we assume graphs derived from intrateam messages may be strongly connected than graphs derived from interteam messages. Our assumption is based on the notion that a chatroom mapping is not one-to-one direct and in general, people subscribe to chatrooms. In that sense the degree distribution is shared widely (observed) by all.

- **Degree centrality** is defined as the degree of a node or the number of edges directed to a node. The degree centrality $dc(v_j)$ of a node $v_j$ equals the number of edges $ij$ directed to $v_j$, i.e., $\sum_i a_{ij}$ [8].

- **Betweenness centrality**, defined as the number of shortest paths passing through a node, is a measure of how important a node is. The betweenness centrality of a node $v_j$ is calculated as $\sum_i \sum_k \delta_{ijk} / \delta_{ik}$ where $\delta_{ijk}$ is the number of shortest paths between $i$ and $k$ that include $j$ and $\delta_{ik}$ is number of shortest paths between $i$ and $k$ [8, 28].

5.3.4 Computing Ranks

We compute ranks based on features extracted for participants. We adopt hierarchical clustering for two reasons. First, it being an unsupervised technique can be applied to datasets of any size. This is useful because we don’t need to create a model from a large dataset and then use the model to produce predictions for a new dataset. Second, we want to infer a hierarchy among team members. The method helps cluster employees with similar rankings.

To compute ranks, we normalize all features extracted for each participant to the interval $[0,100]$. We construct a feature vector for each participant and use the Euclidean distance between them as a basis for hierarchical clustering. We plan to evaluate other distance metrics in future. We adopt the single link algorithm [74], which is a simple and popular technique.
Figure 5.3 shows an example of a hierarchical cluster as a *single link dendrogram*. In Figure 5.3, $d_1, \ldots, d_5$ represent distances between the clusters. We assume that participants in the same cluster have the same rank. Next we provide rules to estimate rank orders between participants in clusters. We derive these rules by checking the consistency in rank outputs by applying the rules on multiple datasets.

**Rank Rule 1** *For the directive and question patterns, increasing distance between clusters from bottom to top indicates decreasing rank.*

**Rank Rule 2** *For the informative pattern, increasing distance between clusters from bottom to top indicates increasing rank.*

Figure 5.3: An example of a single link dendrogram with distance $d$ between clusters, applied to estimate rank $R$ (bottom row).

### 5.4 Evaluation

We evaluate our approach primarily on our military broadcast chat dataset and secondarily on the Enron (directed) email dataset. The evaluation has two steps. First, we evaluate our methods to extract communication patterns, as described in Section 5.3.1. Second, we evaluate our estimation of ranks based on the patterns, as described in Section 5.3.4.

To evaluate the extraction of patterns we use the following metrics: precision, recall, and F-measure. Precision is given by $\frac{true\_positive}{true\_positive + false\_positive}$, recall by $\frac{true\_positive}{true\_positive + false\_negative}$, and F-measure by $\frac{2 \times precision \times recall}{precision + recall}$. The mean absolute error (MAE) of a rank prediction is $\frac{\sum_i |predicted\_rank_i - actual\_rank_i|}{N}$. The accuracy of a rank prediction is $\frac{N - MAE}{N}$, where $N$ is the highest rank.
5.4.1 Data Description

4.1.1 Military

The military dataset was provided by the Mission Command Battle Lab at Fort Leavenworth, Kansas, and the US Army Research Laboratory, Maryland, from an Army simulation experiment (SIMEX). The dataset contains 20 chat rooms, on average, with 42 participants each and 6,998 messages. From the dataset, we consider the following chat rooms: Infantry Brigade Combat Team (IBCT), USMC Maneuver Brigade (MEB), Cavalry (CAV), and Commander (CDR) to evaluate our results. MEB has 546 messages and 50 participants, CAV has 481 messages and 48 participants, CDR has 409 messages and 37 messages, and IBCT Intel has 1027 messages and 64 participants. We consider these chat rooms because, first, they have more messages than the mean number of messages and, second, they have more intrateam messages than interteam messages.

The dataset includes the participants' actual ranks. (Rank 1 is the highest.) Table 5.2 shows the ranks of a few participants who sent more than one broadcast message and belong to a particular military team.

Some participant IDs in the dataset have OCR errors. For example, the ID 8_6i.256_s3 has spurious variants 8_61.256_s3 and 8_6i.256_53 in which i is substituted by 1 and s by 5, respectively. Such errors make it difficult to identify the IDs automatically. To handle such spurious IDs, we select participant IDs with the highest number of messages. For example, if 8_6i.256_s3, 8_61.256_s3, and 8_6i.256.53 have sent 25, 34, and 10 messages respectively, then for our evaluation we consider 8_61.256_s3 with 34 messages.

<table>
<thead>
<tr>
<th>Rank</th>
<th>MEB</th>
<th>CAV</th>
<th>CDR</th>
<th>IBCT Intel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2meb_cdr</td>
<td>8_6i.74_cdr</td>
<td>8_6i.256_cdr</td>
<td>8_6i_s2</td>
</tr>
<tr>
<td>2.</td>
<td>2meb_s2</td>
<td>8_6i.74_s3</td>
<td>8_6i.256_s3</td>
<td>8_6i.156_s2</td>
</tr>
<tr>
<td>3.</td>
<td>2meb_s3</td>
<td>8_6i.74_s2</td>
<td>8_6i.256_s6</td>
<td>8_6i.256_s2</td>
</tr>
<tr>
<td>4.</td>
<td>2meb_fso</td>
<td>8_6i.74_fso</td>
<td>8_6i.256_fso</td>
<td>8_6i.256_s3</td>
</tr>
<tr>
<td>5.</td>
<td>2meb_mech_bn_cdr</td>
<td>8_6i.74_a_cdr</td>
<td>8_6i.256_alo</td>
<td>8_6i.35_s2</td>
</tr>
<tr>
<td>6.</td>
<td>2meb_mech_bn_s3</td>
<td>8_6i.74_b_cdr</td>
<td>8_6i.256_a_cdr</td>
<td>–</td>
</tr>
<tr>
<td>7.</td>
<td>2meb_mech2_bn_cdr</td>
<td>8_6i.74_c_cdr</td>
<td>8_6i.256_b_cdr</td>
<td>–</td>
</tr>
<tr>
<td>8.</td>
<td>2meb_helo_sqdn_cdr</td>
<td>8_6i.74_jtac</td>
<td>8_6i.256_c_cdr</td>
<td>–</td>
</tr>
<tr>
<td>9.</td>
<td>–</td>
<td>–</td>
<td>8_6i.256_wpn_cdr</td>
<td>–</td>
</tr>
</tbody>
</table>
4.1.2 Enron

In the Enron email dataset [26, 47], we arbitrarily consider 62 employees who have sent 38,863 emails with a total of 360,708 email sentences. Prior to the evaluation, we obtain the actual ranks of these 62 employees [32]. The distribution of ranks from 0 to 6 is as follows: 8%, 2%, 29%, 11%, 6%, 36%, and 8%.

5.4.2 Results

We describe the results of our evaluation for extracting patterns and computing ranks on both the military chat dataset and the Enron dataset.

4.2.1 Extracting Communication Patterns

We created the rule-based approach given in Section 5.3.1 using CDR (training data) and evaluated it on CAV (test data). Figure 5.4(a) shows distributions of the communication patterns in these datasets. Notice the high frequency of the informative pattern. Two raters (both graduate students in Computer Science) labeled the data with the various patterns. Their inter-rater agreement (kappa score [15]) was 0.76, which is fairly high. We arbitrarily selected one of the rater’s assigned labels as the ground truth, because we cannot take the average. There are advanced approaches that use Bayesian techniques to estimate a ground truth probability for each classification [?], but this is beyond the current scope and means of the paper.

Based on the training data, we constructed our rules, as described in Section 5.3.1, and evaluated them on the test data. For the training and test data, we found that the F-measures are respectively 0.71 and 0.64 (for the directive pattern), 0.83 and 0.91 (the question pattern), 0.95 for each (for the informative pattern), and 0.84 and 0.83 (overall). Considering the F-measure to identify different patterns as 0.83, we predicted the patterns for the dataset MEB and IBCT Intel.

4.2.1 Computing Ranks via Different Patterns

We used the hierarchical clustering approach described in Section 5.3.4 to compute ranks. Specifically, we considered eight features F₁ to F₈ extracted for each pattern. F₁ represents the counts of responses to patterns, i.e., either directive, question, or informative; F₂, F₃, and F₄ represent the number of negative, neutral, and positive emotions, respectively; F₅ represents the average response time delay; F₆ represents the number of messages sent; F₇ represents the degree centrality; and F₈ represents the betweenness centrality. Since the directive and question patterns have the same relationship, we combined them into the directive+question pattern with the assumption that it would yield improved results over treating them separately.
Using the clustering method, we calculated the percentage accuracies for the four datasets MEB, CAV, CDR, and IBCT Intel for the four patterns respectively. From the mean absolute errors (MAE) we computed the percentage accuracy based on the highest rank $N$ considered for the evaluation. Figure 5.5 describes the overall result. In each panel, the x-axis shows the patterns, i.e., directive, question, informative and directive+question and the y-axis shows the percentage accuracy. From the result, we observed that the percentage accuracy for informative is the highest for all the datasets (73.4%, 76.5%, 69.5%, 68%), which suggests that the informative pattern is a better indicator of hierarchy than other patterns. In addition, we performed one-tailed t-test to check if the accuracy for informative is significantly higher than for directive, question, and directive+question at the significant level of 5%. We find that the accuracy for informative is indeed significantly higher than directive (p=0.03) and directive+question (p=0.002), but not significantly so for question (p=0.06).

### 4.2.2 Evaluating Features

We compared MAEs obtained using only the semantic features with those obtained using only the social features. For the comparison, we performed one-tailed t-tests on the MAEs obtained from the four chat rooms for all patterns.

Table 5.3 summarizes these hypotheses and the results obtained.

In the table, we have stated hypotheses that compare the mean ($\mu$) of the MAEs obtained
using features $F_1$ to $F_8$ for the patterns. When the null hypothesis is rejected we accepted the alternative hypothesis, i.e., one mean is significantly less than the other. Among the features, $F_1$ to $F_4$ represent the semantic features, $F_5$ and $F_6$ represent the communication features, and $F_7$ and $F_8$ represent the social features. We found that the MAEs obtained based on features $F_1$ to $F_4$ were not significantly lower than the MAEs obtained based on features $F_7$ and $F_8$. Recall that $F_5$ is *average response time delay* and $F_6$ is *number of messages*. We found that the MAEs obtained based on features $F_1$ to $F_5$ or obtained based on features $F_1$ to $F_6$ were significantly lower than those obtained based on $F_7$ and $F_8$. When we added $F_5$ and $F_6$ to features $F_7$ and $F_8$, the MAEs obtained were not significantly lower than the MAEs obtained considering features $F_1$ through $F_4$. Similarly, when we added $F_6$ to features $F_7$ and $F_8$, the MAEs obtained were not significantly lower than the MAEs obtained using features $F_1$ through $F_5$. The foregoing suggests that the semantic features are better indicators of hierarchy than the social features.

### 4.2.3 The Enron Dataset

We evaluated our approach on the Enron email dataset [26, 47] as well. A major challenge we faced is to create conversation threads based on a subject or a topic. Whereas in the military dataset we considered the counts of the response messages to *directive*, *question*, and *informative*
Table 5.3: Statistically comparing semantic features with social features (sem-semantic, comm-communication, soc-social, avg-average, resp-response, del-delay, msg-messages, hyp-hypotheses, rej-rejected).

<table>
<thead>
<tr>
<th>#</th>
<th>Alt. Hypotheses</th>
<th>Null hyp. p-val</th>
<th>Null hyp. rej. at 5%?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sem. $(\mu_{F_1toF_4})$ &lt; soc. $(\mu_{F_7toF_8})$</td>
<td>0.23</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>sem. &amp; avg. resp. time del. $(\mu_{F_1toF_5})$ &lt; soc. $(\mu_{F_7toF_8})$</td>
<td>0.04</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>sem. &amp; comm. $(\mu_{F_1toF_6})$ &lt; soc. $(\mu_{F_7toF_8})$</td>
<td>0.00</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>soc. &amp; comm. $(\mu_{F_5toF_8})$ &lt; sem. $(\mu_{F_1toF_4})$</td>
<td>0.08</td>
<td>no</td>
</tr>
<tr>
<td>5</td>
<td>soc. &amp; no. of msg $(\mu_{F_6toF_8})$ &lt; sem. $(\mu_{F_1toF_4})$</td>
<td>0.13</td>
<td>no</td>
</tr>
</tbody>
</table>

messages, for the Enron dataset, we considered the counts of directive and question messages sent by an employee. We considered a message whose subject begins with “RE:” as an informative because it indicates that the message responds to a prior message. To identify patterns we used the rules described in Section 5.3.1. Once the messages were identified, we computed ranks using the rules provided in Section 5.3.4. The features we considered to compute ranks were $F_1$, $F_6$, $F_7$, and $F_8$. We did not consider features $F_2$ to $F_5$ (emotions and average response time delay) because we could not create conversation threads. We constructed $F_7$ and $F_8$ based on the number of messages exchanged between employees.

We found that ranks computed using the informative pattern have higher accuracy (75%) than the directive (74.4%) and question (70.1%) patterns. This results coheres with our finding over the military data. However, unlike the military data, the accuracy from social features (72%) was slightly higher than for the semantic and communication features (71%). We also found that adding semantic features to social and communication features (75%) slightly improved the accuracy over considering only social and communication features (74%). Therefore, along with social and communication features semantic features were important in predicting hierarchy.
5.5 Discussion and Future Work

We provide a novel approach to computing team hierarchy from broadcast messages. To compute the hierarchy, first, we identify three patterns via text mining obtaining F-measures of 80%, 95%, and 60% respectively for question, informative, and directive patterns, and 83% overall. Second, once we identify the patterns, we extract disparate features: semantic, communication, and social. Third, using the features we compute ranks using the hierarchical clustering method. We find that the informative pattern is a better indicator of hierarchy than the other patterns, thus validating our approach. We find that semantic features added with communication features (i.e., the network statistics) are better indicators of ranks than using social features alone. We obtain similar results regarding the usage of patterns to infer hierarchy on the Enron dataset. We also find that semantic features added to social and communication features improve accuracy in predicting hierarchy. However, social features in Enron are better indicators of hierarchy than semantic features. This could be because the Enron dataset is much larger than the military dataset: on average, Enron participants sent more messages than military participants.

Although we consider only two datasets, our study provides some hints as to the differences in how people use chat communications versus email, at least in work-related settings. Email communications would tend to respect predefined organizational relationships (who writes to whom) and thus social features are predictive of hierarchy. In contrast, broadcast communications at the level of connectivity do not respect any predefined relationships. Thus their semantic features are better predictive of hierarchy. In the military setting, the ranks of the participants are well defined. We conjecture that, in settings where ranks are not predefined, such as in collaborations between peers as in open source software development or nascent political movements, broadcast communications would be a way for true hierarchies to emerge.

This work estimates intrateam hierarchy. In future work, we will consider interteam hierarchy. Also, we hope to extend our work on the estimation of a hierarchy to the estimation of team cohesion, trust, and performance. We plan to improve our domain-specific military lexicon to further improve performance. We expect that our results would be stronger on larger datasets where participants communicate more frequently with each other.

5.6 Related Work

There has been a small amount of research on inferring hierarchy from communications. Nishihara and Sunayama [61] compute hierarchy by two measures: based on request actions communicated by a speaker and the number of sentences sent by a speaker. In contrast, instead of identifying requests, we identify patterns such as directive, question, and informative. Moreover, Nishihara and Sunayama do not incorporate features such as emotions, average response time.
delay, or centrality features that can provide important clues to identify hierarchy. Also, they evaluate their work on directed messages but not on broadcast messages.

Gilbert [32] identifies words and phrases from Enron emails [47, 26] that indicate team hierarchy. This work is limited to finding such words and phrases rather than computing a hierarchy. Also, Gilbert’s approach is domain-dependent because it requires words and phrases related to hierarchy. Preparing such lexicons for new datasets can be cumbersome. In contrast, we provide ways to identify patterns that generalizes to different datasets. Also the lexicon we prepare is easy to extract as the verbs are extracted based on their frequencies.

Rowe et al. [66] compute team hierarchy by extracting an undirected graph based on emails exchanged between senders and receivers. They consider centrality measures to compute hierarchy and do not focus on analyzing the content of emails. Hence, Rowe et al.’s contribution does not handle broadcast messages. In contrast, we emphasize understanding the content of the messages to identify the patterns and consider broadcast messages. In addition, we find that patterns and emotions extracted from messages are better indicators of hierarchy than are centrality measures.

Krafft et al. [?] propose a probabilistic model to visualize topic-specific subnetworks in email datasets. In specific, they associate an author-recipient edge (or an email) with different subtopics using K-dimensional topic-specific communication patterns. In our work we take a similar approach where we extract different communication patterns and features from emails and broadcast messages for participants to infer their hierarchy.
REFERENCES


