Monitoring Commitments in People-Driven Service Engagements

Anup K. Kalia[1], Hamid R. Motahari-Nezhad[2], Claudio Bartolini[2], and Munindar P. Singh[1]
[1]North Carolina State University, Raleigh, NC 27695-8206, USA
[2]Hewlett Packard Laboratories, Palo Alto, CA 94304, USA

***Abstract*—People-driven service engagements involve communication over channels such as chat and email. Such engagements should be understood at the level of the commitments that the participants create and manipulate. Doing so provides a grounding for the communications and yields a business-level accounting of the progress of a service engagement. Existing work on commitment-based service engagements is limited to design-time model creation and verification. In contrast, we present a novel approach for capturing commitment-based engagements that are created dynamically in conversations. We monitor commitments identifying their creation, delegation, completion, or cancellation in the conversations. We have developed a prototype and evaluated it on real-world chat and email datasets. Our prototype captures commitments with a high F-measure of 90% in emails (Enron email corpus) and 80% in chats (HP IT support chat dataset) and provides promising results for capturing additional commitment operations.**

I. Introduction

People-driven service engagements provide a natural model of collaborative business processes that are carried out over communication channels such as email and chat. For example, consider IT incident management. When an IT incident happens, a report in email is sent to helpdesk workers or IT experts. The handling proceeds through team collaboration accomplished via communications over email and chat.

We can model such service engagements effectively by describing how the team members create and manipulate commitments to one another. For example, the resolution of an IT incident involves the completion or termination of the commitments among helpdesk works and IT experts. We consider commitments to monitor human interactions as it provide an easy way to determine correctness, i.e., if a commitment is created and active then it ought to be terminated or completed. In simple terms, a commitment specifies who is responsible to whom for achieving what under what circumstances.

Accordingly, an important research problem is how to capture such commitment-based service engagements from conversations, wherein commitments are created and discharged dynamically. Existing works on commitments in service engagements are limited to static model creation and verification [1], [2], [3]. They help business analysts predefine commitment-based business process models. However, they do not capture engagements from human interactions.

To fill the above gap, we provide an approach to capture people-driven service engagements where commitments are created and completed dynamically in their conversations. Figure 1 describes our overall approach. Our tool sits above the communication infrastructure, such as email and chat, and monitors conversations. It determines the progression of a commitment from those conversations 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.

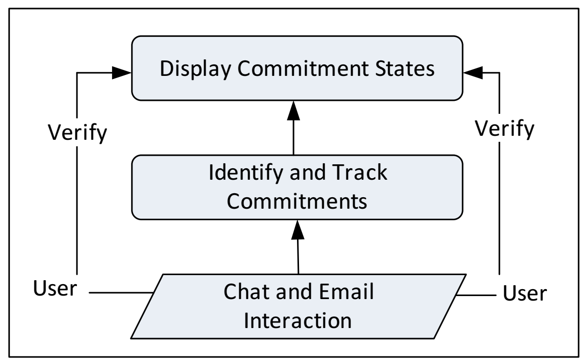

Fig. 1. Monitoring commitments in conversations.

This paper makes the following contributions.

- Define commitments and their lifecycle in the context of people-driven service engagements.
- 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 conversations, 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 [4], [5] 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 II presents the background on commitments. Section III provides a motivating scenario. Section IV defines tasks and commitments in the context of conversations in people-driven service engagements. Section V describes our approach to identifying commitments from email and chat conversations. Section VI explains our dataset, experimentation, and evaluation results. Section VII discusses related work. Section VIII discusses future directions in the current work.

II. Commitments

We adopt Singh's model of commitments [6] 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(DEBTOR, CREDITOR, antecedent, 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.

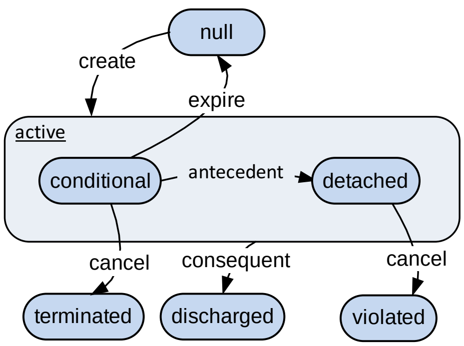

Fig. 2. The lifecycle of a commitment [3].

Telang and Singh [3] present the commitment lifecycle shown in Figure 2. According to Figure 2, a commitment transitions from one state to another due to the following operations: create, detach (antecedent holds), discharge (consequent holds), cancel, and delegate.

- *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.
- *detach(c)* detaches a commitment. A commitment is detached if its antecedent present for a commitment becomes true.
- *discharge(c)* completes a commitment when a debtor executes a committed task.
- *cancel(c)* terminates the commitment c. A commitment can be canceled only by its debtor.
- *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.

III. Running Example

Table I 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.

TABLE I
Sample interaction in an enterprise setting.

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

IV. Understanding Commitments in the Context of People-Driven Processes

We view people-driven service engagements in terms of tasks and commitments derived from the synthetic interactions in Table I. We apply the steps below

- Identify if a message contains a task or an event.
- Check if the task or event indicates commitment creation
- Check if another task delegates, discharges, or cancels the commitment.

A. 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 [7]. We represent a task as T and define it as T(TASK PERFORMER, BENEFICIARY, action). Here, TASK PERFORMER is a business entity that performs the action. BENEFICIARY is a business entity for whom the action is performed. An *action* is a business activity. An action can be a disjunction or a conjunction of subactions. Table II shows messages from Table I that are identified as tasks.

TABLE II
Tasks identified from the interactions of Table I

Task Performer	Beneficiary	Action
AGFIL	John	Inspect car damage
LCS	AGFIL	Inspect car damage
M	LCS	Inspect car damage
M	LCS	Cannot inspect
M1	LCS	Inspect car damage
M1	LCS	Inspected car damage

B. 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 III shows messages from Table I that are identified as C-create and D-create.

TABLE III
COMMITMENT CREATION IDENTIFIED FROM THE INTERACTIONS IN TABLE I.

Debtor	Creditor	Consequent	Operation
AGFIL	John	Inspect car damage	C-create
LCS	AGFIL	Inspect car damage	D-create
M	LCS	Inspect car damage	D-create
M1	LCS	Inspect car damage	D-create

C. Commitment Discharge

A commitment is discharged when the debtor performs the consequent, thereby making it true. Table IV shows messages from Table I 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 IV
DISCHARGE COMMITMENT IDENTIFIED FROM THE INTERACTIONS IN TABLE I.

Debtor	Creditor	Consequent	Operation
M1	LCS	Inspect car damage	D-create
M1	LCS	Inspected car damage	Discharge

D. Subcontracting a Commitment

A commitment $\mathsf{C}(\text{DEBTOR}, \text{CREDITOR}, \top, \text{consequent})$ is subcontracted when its debtor outsources it to a new debtor (debtor'). A new commitment is created $\mathsf{C}(\text{DEBTOR'}, \text{DEBTOR}, \top, \text{consequent})$ but the original commitment remains. Table V shows messages from Table I 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 V
SUBCONTRACT COMMITMENTS IDENTIFIED FROM MESSAGE INTERACTIONS IN TABLE I.

Debtor	Creditor	Consequent	Operation
AGFIL	John	Inspect car damage	C-create
LCS	AGFIL	Inspect car damage	Subcontract
M	LCS	Inspect car damage	Subcontract
M1	LCS	Inspect car damage	Subcontract

E. Commitment Cancellation

A commitment is canceled when its debtor terminates the commitment. Table VI shows messages from Table I, 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 VI
CANCEL COMMITMENT IDENTIFIED FROM MESSAGE INTERACTIONS IN TABLE I.

Debtor	Creditor	Consequent	Operation
M	LCS	Inspect car damage	D-create
M	LCS	Cannot inspect	Cancel

V. APPROACH TO IDENTIFY AND MONITOR COMMITMENTS IN PEOPLE-DRIVEN ENGAGEMENTS

Our process for the identification of commitments from conversations proceeds as shown in Figure 3. First, we preprocess our datasets and extract sentences from the text of conversations. 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.

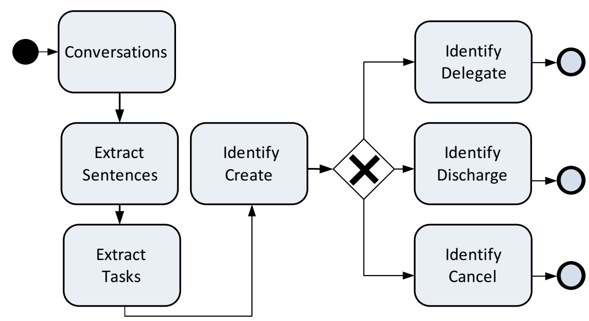

Fig. 3. Process followed to identify commitments.

A. Typed Dependency

To identify a task T from a sentence, we adopt the typed dependency method [8], 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 I. 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 4 shows the triples in a graph format. Below, we explain how we extract tasks from a sentence using these triples.

B. Identifying Conversations 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

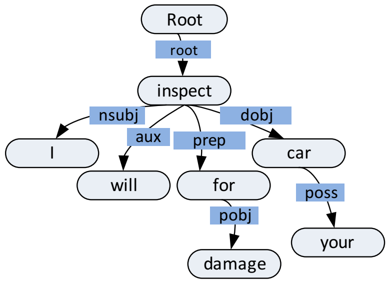

Fig. 4. Typed dependencies derived from a sentence “I will inspect your car for damage” in Table I.

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 conversations as they are already listed chronologically.

C. 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 conversations 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.

D. Identifying Tasks

To identify a task from a sentence, we first extract the features discussed in Section V-C. To obtain the features, we parse a sentence to obtain a *typed dependency array* containing the triples as shown in Figure 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.

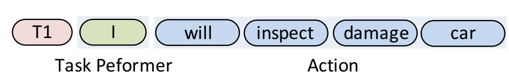

Fig. 5. Steps to identify task from a email sentence “I will inspect your car for damage.”

Figure 5 represents the task structure we obtain after following the above approach on the typed dependency in Figure 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*.

E. 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.

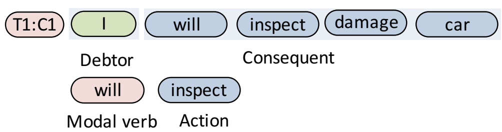

Fig. 6. Steps to identify a commitment from an email sentence "I will inspect your car for damage."

In Figure 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.

F. 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 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* [9] dictionary and the coreference resolution, respectively.

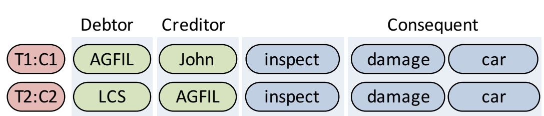

Fig. 7. Steps to identify a subcontract.

G. 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 8. To check if

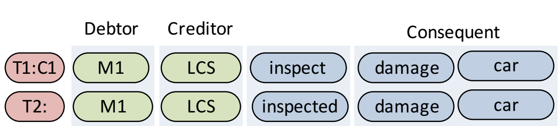

Fig. 8. Steps to identify a discharge.

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 C_1, respectively. If they are the same, we compare their action verbs. We check whether the action verb (inspected) in T_2 is in the past tense (VBD). Then we compare the action verb in T_2 by converting it into its base form (inspect) and trying to match with the action verb (inspect) in C_1. 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 T_2 as discharging C_1.

H. 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 9 describes an example of a commitment C_1 and a

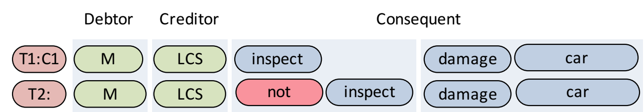

Fig. 9. Steps to identify a cancel.

task T_2. Once we find that the debtor (M) and the creditor (LCS) in C_1 and T_2 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*.

VI. Evaluation and Prototype

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

A. Data

From the Enron email corpus [10], [11], 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 conversations from HP's IT incident management logs comprising 7,154 sentences.

B. 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 IX 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 VII
RESULTS FOR EMAILS, SHOWING PRECISION (P), RECALL (R), AND F-MEASURE (F) FOR DIFFERENT CLASSIFIERS.

	C-create			D-create			Discharge			Cancel			Subcontract			None		
Classifier	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NB	0.84	0.94	0.89	0.81	0.93	0.87	0.21	**0.41**	**0.27**	0.00	0.00	0.00	0.32	0.39	0.35	**0.99**	0.95	0.97
LR	**0.90**	0.95	**0.95**	0.92	0.93	0.94	0.27	0.05	0.08	0.00	0.00	0.00	0.64	0.34	0.48	**0.98**	**0.98**	**0.98**
SVM	0.87	**0.97**	0.92	**0.94**	**0.97**	**0.95**	**1.00**	0.02	0.04	0.00	0.00	0.00	**0.86**	**0.33**	**0.48**	**0.98**	**0.98**	**0.98**

TABLE VIII
RESULTS FOR CHATS, SHOWING PRECISION (P), RECALL (R), AND F-MEASURE (F) FOR DIFFERENT CLASSIFIERS.

	C-create			D-create			Discharge			Cancel			Subcontract			None		
Classifier	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NB	0.73	**0.90**	0.81	**0.85**	0.50	**0.63**	0.60	**0.70**	**0.75**	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.97	0.80
LR	**0.80**	0.85	**0.83**	0.74	0.51	**0.60**	0.64	**0.70**	0.67	**0.22**	**0.13**	**0.16**	0.00	0.00	0.00	**0.97**	**0.98**	**0.97**
SVM	0.79	0.85	0.82	0.73	0.53	**0.61**	**0.63**	**0.71**	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.97	0.97	0.97

C. Results

Tables VII and VIII 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.

$$\text{precision} = \frac{\text{TP}}{\text{TP + FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP + FN}}$$

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

TABLE IX
DISTRIBUTION OF COMMITMENT OPERATIONS IN EMAIL SENTENCES.

Classes	**Email**	**Chat**
Commissive create	342	532
Directive create	162	214
Discharge	38	250
Cancel	7	16
Delegate	12	12
None	3,540	6,130

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 conversations 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 X, 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. For emails, we used SVM and for chats we used LR.

TABLE X
EVALUATION ON INDEPENDENT TEST DATASETS USING SVM FOR EMAIL AND LR FOR CHAT, RESPECTIVELY.

	Email			Chat		
Classifier	**P**	**R**	**F**	**P**	**R**	**F**
C-create	**0.97**	0.84	0.90	**0.90**	0.89	0.89
D-create	**0.94**	0.78	0.86	**0.83**	0.51	0.63
Discharge	0.00	0.00	0.00	**0.66**	0.71	0.69
Cancel	0.00	0.00	0.00	0.00	0.00	0.00
Delegate	**1.00**	0.33	0.98	0.00	0.00	0.00
None	0.96	0.99	0.98	0.96	0.98	0.94

Table XI shows sample examples extracted from the Enron corpus using our approach.

TABLE XI
SAMPLE EXTRACTED FROM THE ENRON CORPUS.

Sentences	**Operation**
Create	
We will expedite materials and installation in an attempt to meet the target date. If any questions, please let me know.	C-create
Please review and send along to your attorney as soon as possible	D-create
Discharge	
I will also check with Alliance Travel Agency to see what may be able to do for us	C-create
I checked with our Travel Agency and they cannot secure cheaper tickets than what we are seeing on the internet	Discharge
Subcontract	
Please take a few moments to review the same and let me know your thoughts	D-create
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	Subcontract
Cancel	
By Wednesday Aug 16 2001, please send all copies of your documentation via interoffice mail to Laura Herrera	D-create
Robbin, please forgive me for not sending this in by Aug 15	Cancel

D. *Prototype Tool*

Figure 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 [12] and Weka libraries [13] 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.

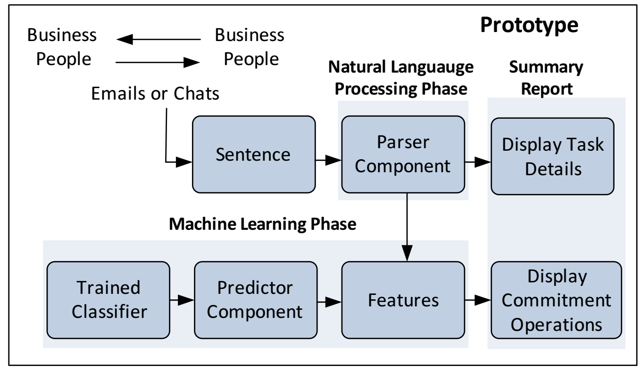

Fig. 10. The architecture of our commitment identification and monitoring tool.

VII. RELATED WORK

Scerri et al. [5] 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. [4] improve over the accuracy results of Scerri et al.'s [5] 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 [14] 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. [15] 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. [16] semantically annotate and extract deontic norms such as obligation, prohibition, and permission from Italian legal texts. De Maat et al. [17] automatically identify different norms from Dutch laws. Savarimuthu et al. [18] 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. [19] 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. [20] extract such workflows from event logs containing workflow enactments. Desai et al. [21] 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. [22] mine changes in logs in adaptive process management systems.

VIII. 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, for improved modeling of service engagements, we will develop enhanced methods for detecting the delegation, discharge, and cancellation of commitments. In particular, 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.

IX. Acknowledgments

This research was partly supported by HP Labs, Palo Alto when Anup Kalia was a summer intern there and partially supported by the Army Research Office under the Science of Security Lablet grant to NC State University.

References

[1] N. Desai, A. K. Chopra, and M. P. Singh, "Amoeba: A methodology for modeling and evolution of cross-organizational business processes," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 19, no. 2, pp. 6:1–6:45, Oct. 2009.

[2] M. P. Singh, A. K. Chopra, and N. Desai, "Commitment-based service-oriented architecture," *IEEE Computer*, vol. 42, no. 11, pp. 72–79, Nov. 2009.

[3] P. R. Telang and M. P. Singh, "Specifying and verifying cross-organizational business models: An agent-oriented approach," *IEEE Transactions on Services Computing*, vol. 5, no. 3, pp. 305–318, Jul. 2012.

[4] A. Lampert, R. Dale, and C. Paris, "Detecting emails containing requests for action," in *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California, 2010, pp. 984–992.

[5] S. Scerri, G. Gossen, B. Davis, and S. Handschuh, "Classifying action items for semantic email," in *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC)*, Valletta, Malta, 2010.

[6] M. P. Singh, "An ontology for commitments in multiagent systems: Toward a unification of normative concepts," *Artificial Intelligence and Law*, vol. 7, no. 1, pp. 97–113, Mar. 1999.

[7] H. R. Motahari-Nezhad, C. Bartolini, S. Graupner, S. Singhal, and S. Spence, "IT support conversation manager: A conversation-centered approach and tool for managing best practice IT processes," in *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference*, ser. EDOC. Washington, DC: IEEE Computer Society, 2010, pp. 247–256.

[8] M. De Marneffe, B. MacCartney, and C. Manning, "Generating typed dependency parses from phrase structure parses," vol. 6, 2006, pp. 449–454.

[9] G. A. Miller, "WordNet: A lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.

[10] A. Fiore and J. Heer, "UC Berkeley Enron email analysis," 2004. [Online]. Available: http://bailando.sims.berkeley.edu/enron_email.html

[11] B. Klimt and Y. Yang, "The Enron corpus: A new dataset for email classification research," in *Proceedings of the 15th European Conference on Machine Learning*, ser. LNCS, vol. 3201, Pisa, 2004, pp. 217–226.

[12] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, 2003, pp. 423–430.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[14] A. Qadir and E. Riloff, "Classifying sentences as speech acts in message board posts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Edinburgh: Association for Computational Linguistics, 2011, pp. 748–758.

[15] J. L. Martínez-Fernández, J. C. González, J. Villena, and P. Martínez, "A preliminary approach to the automatic extraction of business rules from unrestricted text in the banking industry," in *Proceedings of the 13th International Conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems*. London: Springer-Verlag, 2008, pp. 299–310.

[16] R. Bartolini, A. Lenci, S. Montemagni, V. Pirrelli, and C. Soria, "Automatic classification and analysis of provisions in Italian legal texts: A case study," in *Proceedings of the On the Move to Meaningful Internet Systems: OTM Workshops*, ser. Lecture Notes in Computer Science. Springer, 2004, vol. 3292, pp. 593–604.

[17] E. de Maat and R. Winkels, "Automated classification of norms in sources of law," in *Semantic Processing of Legal Texts*, ser. Lecture Notes in Computer Science. Springer, 2010, vol. 6036, pp. 170–191.

[18] B. Savarimuthu, S. Cranefield, M. Purvis, and M. Purvis, "A data mining approach to identify obligation norms in agent societies," in *Proceedings of Agents and Data Mining Interaction*, ser. Lecture Notes in Computer Science, vol. 5980. Springer, 2010, pp. 43–58.

[19] C. Molina-Jiménez, S. Shrivastava, E. Solaiman, and J. Warne, "Run-time monitoring and enforcement of electronic contracts," *Electronic Commerce Research and Applications*, vol. 3, no. 2, pp. 108–125, 2004.

[20] W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. M. M. Weijters, "Workflow mining: A survey of issues and approaches," *Data Knowledge Engineering*, vol. 47, no. 2, pp. 237–267, Nov. 2003.

[21] N. Desai, A. Bhamidipaty, B. Sharma, V. K. Varshneya, M. Vasa, and S. Nagar, "Process trace identification from unstructured execution logs," in *Proceedings of the 8th IEEE Conference on Services Computing*, Miami, 2010, pp. 17–24.

[22] C. W. Günther, S. Rinderle-Ma, M. Reichert, W. M. V. D. Aalst, and J. Recker, "Using process mining to learn from process changes in evolutionary systems," *International Journal of Business Process Integration and Management*, vol. 3, pp. 61–78, 2008.