Identifying Business Tasks and Commitments from Email and Chat Conversations

Anup Kalia, Hamid Reza Motahari Nezhad, Claudio Bartolini, Munindar Singh

HP Laboratories
HPL-2013-4

Keyword(s):

Task Extraction; natural-language processing; Case Management;

Abstract:

Case management applications and, more generally, people-centric processes involve the definition, resolution and communication of commitments for tasks over channels such as chat and email. Identifying and tracking tasks and commitments can help in streamlining the collaborative work in business environments. However, doing so proves challenging due to the syntactical, grammatical, and structural incompleteness of human conversations over chat and email channels. We present a novel approach to automatically identify tasks and commitment creation, delegation, completion, and cancellation in email and chat conversations, based on techniques from natural language processing and machine learning domains. We discover tasks and related parameters from the text of conversations, identify when a commitment to a task emerges and find the state changes of a commitment based features extracted from the text of the conversations. We have developed a prototype and evaluated our approach using real-world chat and email datasets. Our experiments shows high precision for create class i.e., 90% in emails (Enron email corpus) and 80% in a real-world chat dataset and also provides promising results for discharge, delegate, and cancel classes.

External Posting Date: February 06, 2013 [Fulltext] Approved for External Publication
Internal Posting Date: February 06, 2013 [Fulltext]

Identifying Business Tasks and Commitments from Email and Chat Conversations

A. K. Kalia
NC State University
Raleigh NC 27695-8206
akkalia@ncsu.edu

H.R. Motahari-Nezhad, C. Bartolini
Hewlett Packard Laboratories
Palo Alto, CA, United States
hamid.motahari@hp.com

M. P. Singh
NC State University
Raleigh NC 27695-8206
singh@ncsu.edu

ABSTRACT

Case management applications and, more generally, people-centric processes involve the definition, resolution and communication of commitments for tasks over channels such as chat and email. Identifying and tracking tasks and commitments can help in streamlining the collaborative work in business environments. However, doing so proves challenging due to the syntactical, grammatical, and structural incompleteness of human conversations over chat and email channels. We present a novel approach to automatically identify tasks and commitment creation, delegation, completion, and cancellation in email and chat conversations, based on techniques from natural language processing and machine learning domains. We discover tasks and related parameters from the text of conversations, identify when a commitment to a task emerges and find the state changes of a commitment based features extracted from the text of the conversations. We have developed a prototype and evaluated our approach using real-world chat and email datasets. Our experiments shows high precision for create class i.e., 90% in emails (Enron email corpus) and 80% in a real-world chat dataset and also provides promising results for discharge, delegate, and cancel classes.

1. INTRODUCTION

Many processes in organizations are people-driven, and are managed in a collaborative and conversation-oriented manner. Conversations around people-centric processes involve defining and coordinating tasks through commitments among workers over informal channels such as email and chat. Typical examples of such conversations include the handling of insurance claims and IT incidents. For instance, in IT incident management, incident reports are assigned to help desk workers and in some cases to specialized IT experts. The handling usually proceeds through team collaboration and communication over email and chat, in addition to keeping record in specialized systems. Keeping track of all agreed-upon tasks and commitments made during a conversation is a daunting task, and a major source of inefficiencies [4], as the size and number of interactions and processes that a worker is involved in increases.

We investigate the problem of identifying tasks (and related parameters) and commitments from multiparty textual conversations over email and chat. Commitments provide a level of deep modeling that facilitates appropriate progression of tasks. A commitment is created when a worker becomes responsible for a task, whether by volunteering or being directed. Once created, a commitment may be completed, delegated, or canceled. Identifying and monitoring commitments automatically in human conversations is challenging as these conversations are ill-structured, are not necessarily grammatically correct, and contain domain-specific information. For example, such as IP traces, IT-related conversations may contain codes and error logs. In addition, as conversations grows in length and the number of workers involved increases, it becomes difficult to determine the status of the commitments made by any of the workers. Moreover, a single sentence may carry information about several commitments, or several state changes of a given commitment.

The few existing works dealing with tasks or commitments in natural language processing (NLP) [6, 7, 8] consider only part of the problem, i.e., identifying action verb classes in a source such as message board, email corpus, or chat logs. They do not support the identification of commitments, monitoring their progression or lifecycle. In addition, they report low accuracy results.

We introduce a novel approach for the identification of tasks and commitments based on the speech act theory, people-centric processes, and machine learning. We summarize the novel contributions of this paper as follows:

- Define commitments and their lifecycle in the context of people-centric processes and case management applications [5].
- Present an NLP-based algorithm by leveraging typed dependency [1] for identifying a task and its related parameters (owners, deadline, actions) from a chat or email message.
- Determine whether the communication around a task signals the creation, delegation, cancellation, or discharge of a commitment, by extracting selected features from conversations and a classification-based approach. Analyze complex sentence structures to discover cases where a sentence contains multiple task definitions or several lifecycle changes of a commitment such as creation and delegation.
- Develop an automated agent that monitors conversations to identify the tasks and progression in a commitment, and non-intrusively presents them to case workers who determine whether to accept its suggestions.

We have experimentally validated our approach on real-world datasets of email and chat conversations. Our approach yields significantly better accuracy than existing work for task and commit-

ment identification, and performs well for identifying the delegation, cancellation, and discharge of a commitment, which have not been studied by others.

The paper is structured as follows. Section 2 discusses the key background topics on commitments. Section 3 defines tasks and commitments in the context of conversations around people-centric processes. Section 4 describes our approach to identifying commitments from email and chat conversations. Section 6 explains our dataset, experimentation and the evaluation results. Section 7 highlights the limitations of our approach. Section 8 discusses related works. We conclude and discuss future directions in Section 9.

2. BACKGROUND

Several researchers have studied how people converse with each other to create tasks and commitments and collaborate with each other. We review speech acts [9], language/action perspective [14], and commitments [11] as key concepts to build the foundation for the approach presented in this paper. These techniques are foundations for providing automated support for conversation-oriented methods for conducting people-centric processes [4].

2.1 Speech Acts

Searle [9, 10] classified illocutionary acts into five classes: commissives, directives, representatives, expressives, and declarations. A message is classified as a *commissive* if the sender of the message promises to take an action in the future. A message is classified as a *directive* when the speaker intends the receiver to do something. A message is classified as a *representative* if the sender commits to the truthfulness of the message. A message is classified as a *expressive* when the sender expresses his or her psychological state. A message is classified as a *declaration* when the sender of the message brings about a change in status of the referred object or objects. Searle theory provides the basic idea of identifying commissive and directive actions in people's conversations, however, no work is reported on how commissive and directive actions progress in conversation.

2.2 Language/Action Perspective

Winograd [14] extends speech acts to understand human cooperative activity as conversations. In this model, a message in a conversation is identified commissive when it is either *requested*, *offered*, or *counter-offered*, let's say from a party (A) to another party (B). The conversation progresses when B accepts the offer from A and assert A that the conditions are met. Now, if A declares that he or she is satisfied, the conversation reaches a *completion* state.

2.3 Commitments

Unlike Winograd's approach that captures every request as a commitment, Singh's model of commitments [11] capture business relationships between any two entities. These entities can be either employees within a company or the companies themselves. Specifically, commitments denote business meanings underlying the interactions between these business entities. In this model, a commitment is a conditional business relationship directed from a debtor to a creditor, which can be formalized as

$$C(\text{DEBTOR}, \text{CREDITOR}, antecedent, consequent).$$

The formula shows that the debtor is committed to bringing about the consequent for the creditor provided the antecedent holds. When a debtor offers a commitment to a creditor, the commitment is created and becomes active. When the antecedent is brought about, the commitment is detached and 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 and the antecedent may be omitted:

$$C(\text{DEBTOR}, \text{CREDITOR}, \top, consequent).$$

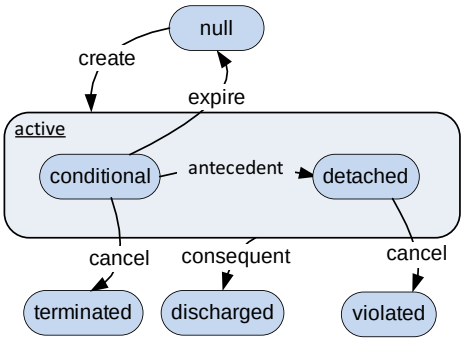

Figure 1: The life cycle of a commitment [13].

Telang and Singh [13] present the commitment life cycle as shown in Figure 1. According to Figure 1, a commitment transits 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 gets created when a debtor voluntarily offers to do a task or when he is assigned to do a task.
- *detach(c)* detaches a commitment. A commitment gets detached if a condition or an antecedent present for a commitment holds 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 debtor delegates his commitment.

3. TASKS AND COMMITMENTS IN TEXT

In this section, we discuss tasks and commitments in connection with textual conversations. To explain the concepts, we use examples from the Enron email corpus and an chat corpus from an HP IT incident management application. The email corpus contains over 250,000 emails sent or received by over 87,000 people. It consists of emails exchanged by Enron employees in the time leading up to the Enron bankruptcy that were revealed as part of the Federal investigation of Enron. The chat corpus contains conversations related to HP IT incident management systems. The corpus is collected from the logs of chat conversations among workers in a customer facing incident management system for big IT accounts. We extracted logs from the months of May and June 2012 and analyzed them for our work.

3.1 Task

A task is a business activity that is either pre-defined (part of a best practice process) or created on-the-fly by people in a conversation [4, 12]. We represent a task as T and define it as

$$T(\text{SUBJECT}, \text{OBJECT}, action)$$

In this definition, a **subject** is a business entity that performs the action. An **object** is a business entity for whom an action is being performed by a subject. An **action** is a business activity performed by a subject. An action can be a disjunction or a conjunction of subactions. And finally, a **deadline** represents the time-out condition for an action to be performed. If the time-out occurs, then we consider the action to be expired. Consider an example from the Enron corpus where Kim sends and email to John: *I will pick it up tonight*. Here, the subject is Kim and the object is John. The action is *pick it up* and the deadline is *tonight*. We can formally represent this task as T1 = T(KIM, JOHN, *pick it up*).

3.2 Create Commitment

We adopt the Singh's model of commitments [11], reviewed in Section 2.3, as an action that is performed by a debtor for a creditor. In our context, we map a task to a commitment as follows:

Definition 1. **(Mapping a task with a commitment)** *If a task T is a commitment C, then subject ∈ T is debtor ∈ C, object ∈ T is creditor ∈ C, action ∈ T is consequent ∈ C, and deadline ∈ T is consequent_timeout ∈ C.*

Consider an example from the Enron dataset where Earl Chanley sends the following email to Jo Williams *We will expedite materials and installation in an attempt to meet the target date*. To extract the structure of a commitment from the sentence, first, we identify a task from it and formally we write it as T1 = T(EARL CHANLEY, JO WILLIAMS, expedite materials $\wedge$ expedite installation). Second, we map T1 to a commitment C1 = C(EARL CHANLEY, JO WILLIAMS, $\top$, expedite materials $\wedge$ expedite installation) as the email indicates a commitment from Earl (as subject) to Jo (as object). By the classification of Searle's illocutionary acts [10] the above examples represents an offer from Earl to Jo. Therefore we define a commissive creates as

Definition 2. **(Commissive create)** *The creation of a commitment C is commissive when the debtor voluntarily offers to perform the consequent for the creditor*

Similar to a commissive create, a commitment can be a directive. We define a directive create as:

Definition 3. **(Directive create)** *The creation of a commitment C is directive when the creditor delegates the consequent to the debtor*

Consider and example of a directive create from the Enron corpus where Steven Schleimer sends the the following email to Kim Watson *Please review and send along to your attorneys as soon as possible*. We formally write the email as C(KIM WATSON, STEVEN SCHLEIMER, $\top$, review $\wedge$ send). In our machine learning approach, we label the sentence as either *ccreate* or *dcreate* based on whether the sentence indicates a commissive or a directive.

3.3 Discharge Commitment

A commitment is discharged when the debtor executes the consequent thereby making it true. An example from the Enron dataset is one where Kim sends an email to Dorothy with the following message, *I will also check with Alliance Travel Agency (formerly Travel Agency in the Park) to see what they may be able to do for us*. The message indicates the creation of a commitment from Kim to Dorothy and can be represented as C1 = C(KIM WATSON, DOROTHY MCCOPPIN, $\top$, check with Travel Agency). In following, Kim sends another email to Dorothy with the following message, *I checked with our Travel Agency and they cannot secure cheaper tickets*. The task T2 from this email T2 = T(KIM, MCCOPPIN, checked with Travel Agency) discharges C1 as Kim checked with the travel agency.

3.4 Delegate Commitment

A commitment is delegated when its debtor outsources it to a new debtor. There can be two cases of delegation. In the first case, the new debtor does not know about the old creditor. We define the delegation as

Definition 4. **(Unknown creditor delegation)** *When a debtor delegates his or her commitment C1 to a debtor', then debtor ∈ C1 is the creditor' ∈ C2 and both the commitments C1 and C2 can be written as*

$$C1 = C(debtor, creditor, \top, consequent)$$

$$C2 = C(debtor', debtor, \top, consequent)$$

Consider an example from the Enron dataset where Gregory sends an email to Glen with the following message, *Please take a few moments to review the same and let me know your thought*. In this email, Gregory delegates a commitment to Glen for reviewing something. We formally represent the commitment as C1 = C(GLEN HASS, GREGORY KLATT, $\top$, review statements). In the follow up, Glen sends another email to Steven with the message, *This appears to be 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*. In this email, Glen delegates his commitment to Steven thereby creating another commitment from Steven to him. This commitment can be formalized as C2 = C$\langle$STEVEN HARRIS, GLEN HASS, $\top$, review statements$\rangle$. As, we can see this commitment has a new debtor (debtor') as Steven while the creditor' Glen is the debtor of the previous commitment.

Now, in the second case of the delegation, the new debtor is committed to the old creditor. We define the delegation as

Definition 5. **(Known creditor delegation)** *When a debtor delegates his or her commitment C1 to a debtor', then creditor ∈ C1 is the creditor' ∈ C2 and both the commitments C1 and C2 can be written as*

$$C1 = C(debtor, creditor, \top, consequent)$$

$$C2 = C(debtor', creditor, \top, consequent)$$

Consider an example from the Enron dataset where Robert sends an email to Drew with the following message, *Please let me know if your business unit has any problem with this course of action*. In this email, Drew creates a commitment with Robert for letting him know something. The commitment is formalized as C(DREW, ROBERT, $\top$, inform about business unit having any problem). In following up, Drew sends another email to Kim; *Please review this message and advise Robert Williams of any relevant information*. In this email, Drew delegates his commitment to Kim thereby creating another commitment from Kim to Robert. This commitment can be formalized as C(KIM, ROBERT, $\top$, provide information). As we can see, this commitment has a new debtor as Kim while the creditor is still the same i.e., Robert.

3.5 Cancel Commitment

A commitment is canceled when the debtor of the commitment terminates the commitment. An example from the Enron dataset is Diane sends an email to Kimberley, *Can you please provide me the details about the amounts from prior months by this Friday?*. In this email, Diane delegates a commitment to Kimberley that can be represented as C1 = C(KIMBERLEY, DIANE, $\top$, provide the information). Later, Kimberley sends an email back to Diane with the message, *I cannot not give you the details by Friday*, thereby canceling her existing commitment toward Diane.

4. IDENTIFYING TASKS AND COMMITMENTS

Our process for the identification of commitments from emails and chat conversations is depicted in Figure 2. We first extract sentences from the text of conversations. We then identify the tasks and then candidate commitments or changes in the lifecyle of commitments using a combined NLP-based and machine learning approach. Using NLP and a set of heuristic-based rules applied on features extracted from the text of conversations, we identify certain tasks and commitments. However, the application of NLP-based rules is limited to identifying pre-determined classes and patterns of tasks and commitments. We augument our approach with a supervised machine learning approach for the identification of commitments and their lifecycle. This helps in the identification of commitments for which their various expressions and forms in the natural language may not be captured in patterns and rules.

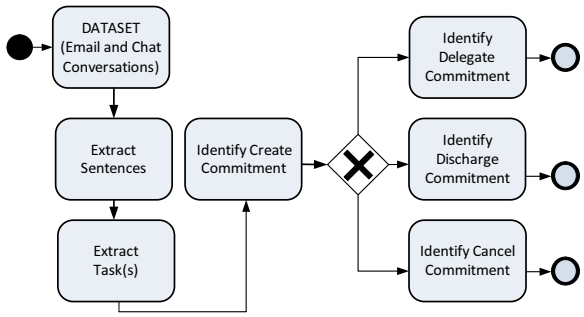

Figure 2: Process followed to identify commitments.

For identification of a task T from a sentence, we check whether the sentence has a subject, an object and an action verb is present. Although it sounds simple, the identification of tasks in a sentence is nontrivial. For example, changing from active to passive voice results in shifting positions of words that indicate subject, object, or the action. To deal with this issue, we chose to parse the sentences using the typed dependency method [1] which outputs the relations between individual words in a sentence, and is largely independent of the exact sentence structure. A relation between any two words is defined as

***Definition** 6.* **(Relation)** *A relation is a triplet of the name of the relation, governor, and dependent where the governor and dependent are words from a sentence and it can be represented as*

$$relation(governor, dependent)$$

Marneffe *et al.* [1] represent such relationships in a hierarchical manner with the most generic relation as the root. For example, a relationship can start with an *arg* (argument) and it can branch into the *subj* (subject) and the *comp* (complement) relationship. Consider a sentence, *Lorraine, I will be in a meeting during this time*. In the sentence, the triplets are root (root, be), nsubj (be, I), aux (be, will), advmod (be, Lorraine), prep_in (be, meeting), det (meeting, a), prep_during (be, time), det (time, this). Figure 3 shows the triplets in a graph format. In this sentence we can see that there is an *nsubj* relation in which *I* is the *dependent* and *be* is the *governor*. This relation suggests that the subject in the sentence is *I* and his action is *be*.

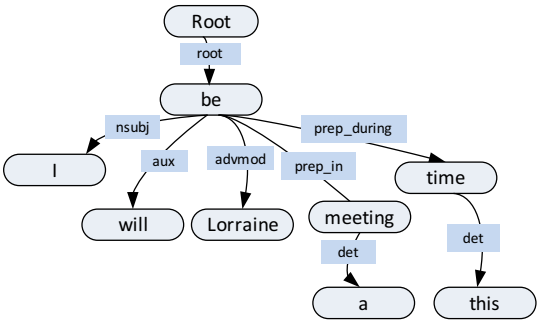

Figure 3: Typed dependencies derived from a sentence "Lorraine, I will be in a meeting during this time" chosen from the Enron corpus.

4.1 Identifying Conversations and Sentences

We preprocess the Enron and the HP IT incident management datasets to make them suitable for parsing and extracting features. Since the email and chat dataset types 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 features. Unlike, emails we do not prepare conversation threads for chat conversations as they are already listed chronologically.

4.2 Extracting Features

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

- **Co-reference 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 co-reference 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 find the debtor for a commitment.

- **Named entity resolution** (NER), identifies a noun whether it's 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 to tag a word with a part of speech. The POS tags help to identify the type of personal pronoun for a subject and the state of the verb associated with the subject so as to identify the debtor of a commitment and the state of a commitment, respectively. The present tense of the verb

indicates that the commitment is either a commissive or a directive whereas the past tense indicates the commitment is discharged. Examples of POS tags used in our work include NN (for nouns), NNP (pronouns), PRP (personal pronouns), VB (present tense verbs), VBD (past tense verb), VBZ (3rd person singular verb), VBN (past participle) and VBP (non-3rd person singular).

- **Typed dependencies extraction** As discussed in Section 4, a typed dependency relates words in a sentence and gives a clue as to its logical structure.

Before explaining the algorithms for identifying tasks and commitments, we briefly mention the features in the feature vector that are used to train classifiers for email and chat datasets. 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. For example, the sentence *He will handle the issuance of the LC.* has a modal verb *will* that indicates the creation of a commitment. An *action verb* indicates whether a commitment is present in a sentence. For example, *I can imagine that your family reunions are just a hoot!* is not a commitment because the verb *imagine* is not an action verb. The *present tense* signals the creation, delegation, or cancellation of a commitment. In the following example: *She and I will get together on the results of these meetings.* that suggests a create commitment, *handle* and *get* are in the present tense. And, the *past tense* signals the discharge of a commitment. In the following example: *I have reviewed the list you sent me regarding items you would like to see in the Data Room.* that suggests a discharge commithas has the action verb *reviewed* in the past tense.

The *debtor* of a commitment is the task performer. The *creditor* of a commitment is the one debtor commits to. A *deadline* indicates a commitment creation or delegation. For example, in the sentence *It will be posted today and the policy will go into effect for Friday's gas day.* indicates *today* and *Friday* as the deadlines. The prior *creation of a commitment* is a prerequisite for discharge, delegation, and cancellation if the create commitment already exists. And a *delegation signal* is identified when a debtor occurring as a creditor indicates that the debtor delegates an existing commitment to a new debtor. The *negative verb* indicates the presence of a canceled commitment. For example, in the sentence *I cannot give you the amount details not give* indicates a negative verb.

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, third). The *bigram of a modal verb and a second person PRP* indicates a directive creation. For example, in the sentence *Can you help me with the following outstanding items relating to the Info Memo*, the bigram *Can you* indicates directive creation. The *bigram of a first person PRP and a modal verb* indicates a commissive. For example, in the sentence *We will expedite materials and installation*, the bigram *We will* indicates commissive creation.

The *bigram of 'please' and an action verb* indicates a directive. For example, in the sentence *Please review and send along to your attorney as soon as possible*, the bigram *please review* indicates a directive commitment creation. And, a *question mark* in a sentence indicates a directive commitment creation.

4.3 Identifying Tasks

To identify a task from a sentence, we first extract features discussed in Section 4.2. The features are input to Algorithm 1 to identify subject, object and the action that the subject performs for a candidate task. Our algorithm slightly differs between emails and chats especially in extracting the subject and the object as in case of chats it is more difficult to find them than in emails. In case of emails, it is easier to find them because the metadata provides information such as sender's and receiver's names. In case of chat messages, it is difficult because we only get the chat initiator name as the metadata and the co-reference resolution does not work well for resolving pronouns such as *you* in multiparty conversations.

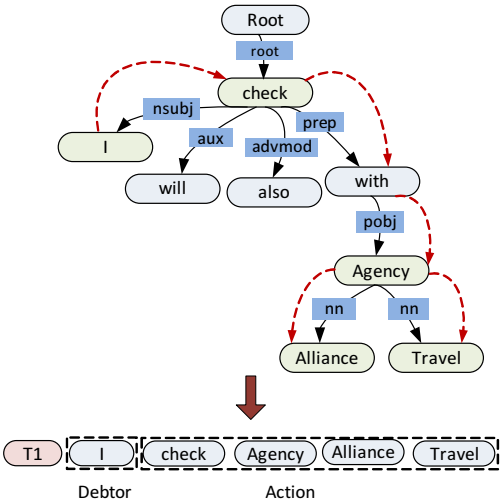

Figure 4: Steps to identify task from a email sentence "I will also check with Alliance Travel Agency".

Once we parse a sentence, we get a typed dependency array as in the example shown in Figure 4. We define the typed dependency array as

***Definition** 7.* **(Typed dependency array)** *The typed dependency array consists of relations derived from a sentence using the typed dependency method. The typed dependency array is represented in terms of relations as, typed dependency array = {relation$_1$, relation$_2$, relation$_3$, . . . }.*

In the above definition, a *relation* can be an *nsubj*, *aux*, *advmod*, and so on. In a typed dependency array, we look for *nsubject* relation and check if the dependent is a *valid subject* and the governor is a *valid verb*. We define subject validity as

***Definition** 8.* **(Valid subject)** *A dependent is considered a valid subject if the POS tag associated with the dependent is NNP or NER resolves the subject as a* PERSON *or an* ORGANIZATION.

We define verb validity as

***Definition** 9.* **(Valid verb)** *A governor is considered valid if the POS tag associated with the governor is either VB, VBD, VBP, VBZ, or VBN and if the governor is an action verb.*

We define an action verb as

***Definition** 10.* **(Action verb)** *The verb that expresses an action or doing something.*

By checking for an action verb, we eliminate words that are verbs but do not express actions. If both the governor and the dependent are valid, we store the dependent as the subject and the governor as the action for the subject. We extract the action details using the action verb by finding its dependencies in the array of triplets by looking for nouns or verbs associated with the action verb. Algorithm 1 shows the steps to extract subject, object, and action from

a sentence in emails. In this algorithm if the POS tags associated to the dependent is PSP first person (I, we) then the subject is the sender of the email. In case of chats, the subject is the chat initiator. If the POS tag associated with the dependent is PRP second person (you), then the receiver is the subject of the task. In case of chats, it is difficult to resolve *you* in a multiparty conversations. If the POS tag associated with the dependent is PRP third person, then the approach for both emails and chats is the same. We consider the dependent as the subject. If the POS tags associated with the dependent is personal PRP (he, she, they) then we resolve them with co-reference resolution.

Algorithm 1: extract task($sentence$)

```
1  typed dependency array ← get typed dependencies(sentence);
2  foreach nsubj ∈ typed dependency array do
3      if get POS(dependent ∈ nsubj) is PRP first person then
4          subject ← sender of the email;
5          object ← receiver of the email;
6      else
7          if get POS(dependent ∈ nsubj) is PRP second person
           then
8              subject ← receiver of the email;
9              object ← sender of the email;
10     else
11         if get POS(dependent ∈ nsubj) is PRP third person
           then
12             subject ← dependent;
13             object ← extractObject(sentence);
14     if valid subject(dependent) ∧ valid verb(governor) then
15         action ← extract action details(governor);
```

Figure 4 represents the typed dependency array for the sentence *I will also check with Alliance Travel Agency*. We extract the *nsubj* relationship i.e., nsubj (check, I). The subject is *I* and the action verb is *check*. Once we extracted the subject and the action, we extract nouns or verbs related to action such as *Agency*, *Alliance*.

4.4 Identifying Create

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 in a task has a relationship with a modal verb and the word *please*. A modal verb can be defined as

***Definition** 11.* **(Modal verb)** *A modal verb is a word that expresses possibility, likelihood, or obligation. Words that indicate modality are will, shall, can, could, would, should, may, might and must.*

The word *please* indicates a request or a delegation of a task. If the verb has a relation with these words (modal or please), then we label the task as commitment creation and store the subject as the *debtor*, the object as the *creditor*, and the action as the *consequent* respectively of the commitment. Also, as discussed in Section 3.2, commitments can be created in two ways: the commissives or directives. We mark a sentence as a commissive when an action verb is following a modal verb. We mark a sentence as a directive when a second person personal pronoun such as *you* is following a modal verb or an action verb following the word *please*. Algorithm 2 summarizes the method for identifying commitments from sentences.

In Figure 5, the action verb *check* is in the present tense (VB) and has a relationship with a modal verb *will*. Therefore, the task is considered as creating a commitment. Since here the action verb

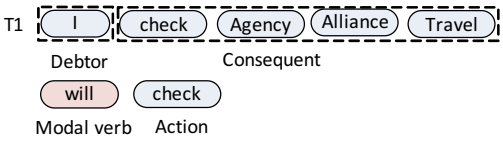

Figure 5: Steps to identify a commitment from a email sentence "I will also check with Alliance Travel Agency".

follows the modal verb, we consider it a commissive. In Algorithm 2, we check if the action is in present tense (VB). Then, we check if there is a relation that associates the action with a modal verb or *please*.

Algorithm 2: identify create commitment(sentence)

```
1  T1 ← extract task(sentence);
2  commitment ← ⊥;
3  if getPOS(action verb ∈ T1) is VB then
4      foreach relation ∈ typed dependency array do
5          if (dependent ∈ relation is action) ∧ (governor ∈
           relation is (modal verb ∨ please)) then
6              commitment ← ⊤;
7          else
8              if (governor ∈ relation is action) ∧ (dependent ∈
               relation is (modal verb ∨ please)) then
9                  commitment ← ⊤;
```

4.5 Identifying Delegate

We discussed in Section 3.4 that there are two kinds of delegation. In one, the old creditor is unknown to the new debtor, whereas, in another, the old creditor is known to the new debtor. Consider Figure 6(a) that shows an example where the new debtor Steven does not know about the old creditor Gregory. Similarly, consider Figure 6(b) where the new debtor Kim is committed to the old creditor Robert.

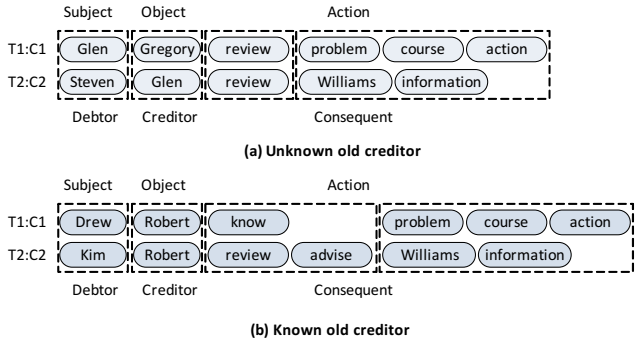

Figure 6: Steps to identify delegate.

To identify delegation in sentences, we check a commitment C2 created after another commitment C1. To find the unknown creditor delegation, we check if the debtor in C1 is the creditor in C2 and if the consequents in C1 and C2 are the same. To find the known creditor delegation, we check if the creditor in C1 is the creditor in C2 and if the consequents in C1 and C2 are the same. To match the consequents in the commitments, we check whether the action verbs in both commitments are the same or related as *synonyms*, *hypernyms*, or *hyponyms*. If they are the same or related, we check

whether nouns in both the commitments are the same or related using the co-reference resolution. Finding delegation in case of email is easier than chats as we know in advance the debtor and creditor of the commitment based on the sender and receiver information. Unlike emails, in chats, finding delegation is difficult as sometimes it is extremely hard to find the creditor of a commitment.

Algorithm 3: identify delegate commitments(sentence)

```
C2 ← identify create commitments(sentence);
delegation ← ⊥;
unknown creditor ← ⊥;
known creditor ← ⊥;
foreach C1 ∈ commitment array do
    if debtor ∈ C1 is creditor' ∈ C2 then
        unknown creditor ← ⊤;
    else
        if debtor ∈ C1 is creditor' ∈ C2 then
            known creditor ← ⊤;
    if action verb ∈ C1 is action verb ∈ C2 then
        if nouns ∈ C1 is nouns ∈ C2 then
            if unknown creditor ∨ known creditor then
                delegation ← ⊤;
```

4.6 Identifying Discharge

If an identified task is in the past tense, it may signals a discharge commitment, and we need to compare it with existing commitments. For greater clarity, let us consider an example of a commitment C1 and a task T2.

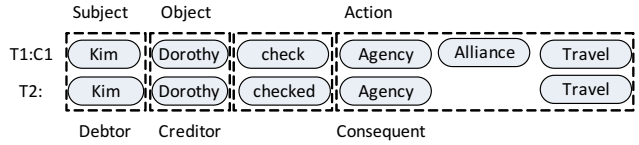

Figure 7: Steps to identify discharge.

To check if T2 discharges C1, we compare the subject and object in T2 with the debtor and creditor in C1 respectively. If they are the same, we compare their action verbs. We check whether the action verb in T2 is in the past tense (VBD). Then we compare the action verb in T2 by converting it into its base form and trying to match with the action verb in C1. If both the verbs are the same or related as either *synonyms*, *hypernyms*, or *hyponyms*, we compare the nouns in both the tasks. If they are the same or related, we mark T2 as discharging C1. Figure 7 clearly shows that debtor and creditor in both C1 and T2 are the same. The main action verb in T2 is *checked* and it is in the past tense. Therefore, we convert it into its base form *check* and compare it with the action verb in C1. Note that the base form of a verb is the simplest form in which it appears in a dictionary without any ending. Since they are the same, we compare the nouns in C1 and T2. We see that the nouns are related. Therefore, we mark T2 as discharging C1. Algorithm 4 describes these steps.

4.7 Identifying Cancel

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*.

Algorithm 4: identify discharge commitments($sentence$)

```
T2 ← extract task(sentence);
discharge ← ⊥;
foreach C1 ∈ commitment array do
    if (subject ∈ T2 is debtor ∈ C1) ∧ (object ∈ T2 is
    creditor ∈ C1) then
        if (getPOS(action verb ∈ T2) is VBD) ∧
        (getBaseVerb(action verb ∈ T2) is action verb ∈ C1)
        then
            if nouns ∈ T2 is nouns ∈ C1 then
                discharge ← ⊤;
```

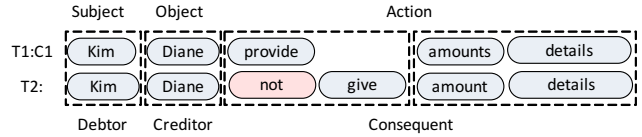

Figure 8: Steps to identify cancel.

Figure 8 describes an example of a commitment C1 and a task T2. Once we find that the debtor and the creditor in C1 and T2 are the same, we compare their main action verbs and nouns respectively. If they are same, we check whether the action verb is a negative verb. Note that a verb considered as negative if either it has a relation with a negative word such as *not*. Algorithm 5 describes the above steps.

Algorithm 5: identify cancel commitments(sentence)

```
T2 ← extract task(sentence);
cancel ← ⊥;
foreach C1 ∈ Commitment Array do
    if (subject ∈ T2 is debtor ∈ C1) ∧ (object ∈ T2 is
    creditor ∈ C2) then
        if getBaseVerb(action verb ∈ T2) is action verb ∈ C2
        then
            if nouns ∈ T2 is nouns ∈ C1 then
                if negative verb(action verb ∈ T2) then
                    cancel ← ⊤;
```

5. EVALUATION AND PROTOTYPE

To validate our approach, we use two methods: (1) automatic labeling of data (sentences) using Algorithms 2, 3, 4, and 5, and then using 10-fold cross validation, and (2) manually labeling a subset of data used for training and testing our approach. We applied three machine learning classifiers and evaluated them. We describe the data, the extracted features from the datasets below.

5.1 Data

For evaluation and validation purpose, from the Enron email corpus, we selected 4161 email sentences that were exchanged between Kimberly Watson, an employee of Enron, and more than 50 people which includes her co-workers at Enron, clients, friends, and family members. The emails were collected and prepared by combining the data from text files provided by CMU [3] as well as from a database dump [2] provided by UC Berkeley. We combine the data from both so as to reduce the missing information. For the chat data, we selected 271 conversations from HP IT incident

management logs comprising of 7154 sentences.

5.2 Labeling Data

Once the features are extracted for each sentence from the dataset, we had two annotators to label the sentences. We cross checked the results for any possible conflict resolution. We resolved conflicts by allowing two annotators to collaborate and discuss their labels for sentences. Next, we ran classifiers the three classifiers of Naïve Bayesian (NB), Logistic Regression (LR), and Support Vector Machines (SVM) and used 10-fold cross validation to produce results.

We annotated 4161 email and 7154 chat sentences. The two independent annotators achieved overall an interrater agreement (kappa score) of 0.83 for both email and chat. Table 3 shows the distribution of the email and chat sentences as annotated.

6. RESULTS

We annotated 4161 email and 7154 chat sentences. The two independent annotators achieved overall an interrater agreement (kappa score) of 0.83 for both email and chat. Table 3 shows the distribution of the email and chat sentences as annotated.

Table 3: Distribution in email sentences.

Classes	Email	Chat
Commissive create	342	532
Directive create	162	214
Discharge	38	250
Cancel	7	16
Delegate	12	12
None	3540	6130

Table 1 and Table 2 represent results for email and chat data respectively, using NB, LR, and SVM classifiers and ten-fold cross validation. In both the tables, we represent C-create as commissive create, D-create as directive create, P as precision, R as recall and F as F-measure.

First, we considered one of the simplest probabilistic text classification approach, Naïve Bayes (NB). The NB classifier approach assumes that attributes in consideration are independent of each other. Using NB, for emails, our results show high precision for commissive creation(84%) and directive creation(81%) while low precision for delegation(32%), discharge(21%), and cancellation(0%). In case of chats, we obtain slightly lower precision for commissive creation (73%) than email, however, we obtain higher precision for directive (85%) and discharge (60%). The precision remains same for cancel (0%). For delegate, the precision (0%) was low compared to precision for email.

Second, we used the Logistic Regression (LR) classifier. Using LR, in emails, we obtain significantly high precision values for commissive (90%), directive (92%) compared to NB while low precision values for delegate (64%), discharge(27%), and cancel (0%). In case of chats, LR performs better than NB for commissive creation and cancellation with precision of 80% and 22% respectively. However, results for other classes are lower, i.e., 74% for directive creation, 64% for discharge, and 0% for delegate. Overall, the results for chat using LR are lower compared to emails except discharge and cancel.

Third, we used the Support Vector Machine (SVM) classifier. Using SVM, in emails, we obtain significantly higher precision, compared to NB and LR, for commissive creation (87%), directive creation(94%), discharge(100%), and delegation(86%) while same precision for cancellation (0%). For chats, using SVM, we obtain lower precision than emails for commissive creation (79%), directive creation (73%), and discharge (63%). However, the f-measure and recall value for discharge is higher in chats than emails. Again, this is due to the higher percentage distribution of discharge in emails and chats.

Overall we obtain high precision, recall and f-measure for a commissive creation and a directive creation for both emails and chats using NB, LR, and SVM. The results for both the classes are high because both the classes are independent of each other and the distribution of these classes are high in both the datasets. The results for other classes are low because other classes depends on the prior existence of create commitment classes and it is difficult to find this specific feature automatically. Compared to discharge and delegate the results for delegate is higher in case of emails because we can easily find out the debtor and the creditor of a commitment based on the sender's and receiver's information. The results for discharge in email is low because the percentage of distribution of discharge is extremely low as lot of tasks indicating discharge are executed, however, are never mentioned in emails. In case of chat, the precision for discharge is higher because the distribution of discharge is high as people in chat conversations immediately report their progress. However, the overall percentage is low for both emails and chats because it is difficult to compare consequent by matching verbs and nouns. For emails, we find a high precision value 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 got 0% precision in emails and 22% in chats, This is because it is again difficult to find out the prior existence of a commitment as well find out the negative words associated with the action verb.

We evaluated our training model on independent test datasets. Our test datasets contains 1326 email and 2299 chat sentences. For emails we used SVM and for chats we used LR.

Table 4: Evaluation on test datasets using SVM for emails and LR for chats 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

6.1 Initial Prototype

Figure 9 represents the architecture of our interactive tool for commitment identification and tracking. The tool offers an agent that can be pluggined into online chat messengers, and identifies discovered commitments in a panel for verification and confirmation to conversation participants. When a person sends a chat message, it parses the sentence using our *parser component* and extract potential *task details* and *features*. Using our predictor component and a trained model we identify the class based on extracted features. The parser and predictor components are Java-based and use the Stanford parser and Weka libraries respectively. We predict classes using the SVM classifier. Finally, the display component produces the summary of all tasks and commitments based on chats. Figure 10 shows a screenshot of the prototype tool.

Table 1: Results for emails.

	C-create			D-create			Discharge			Cancel			Delegate			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 2: Results for chats.

	C-create			D-create			Discharge			Cancel			Delegate			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

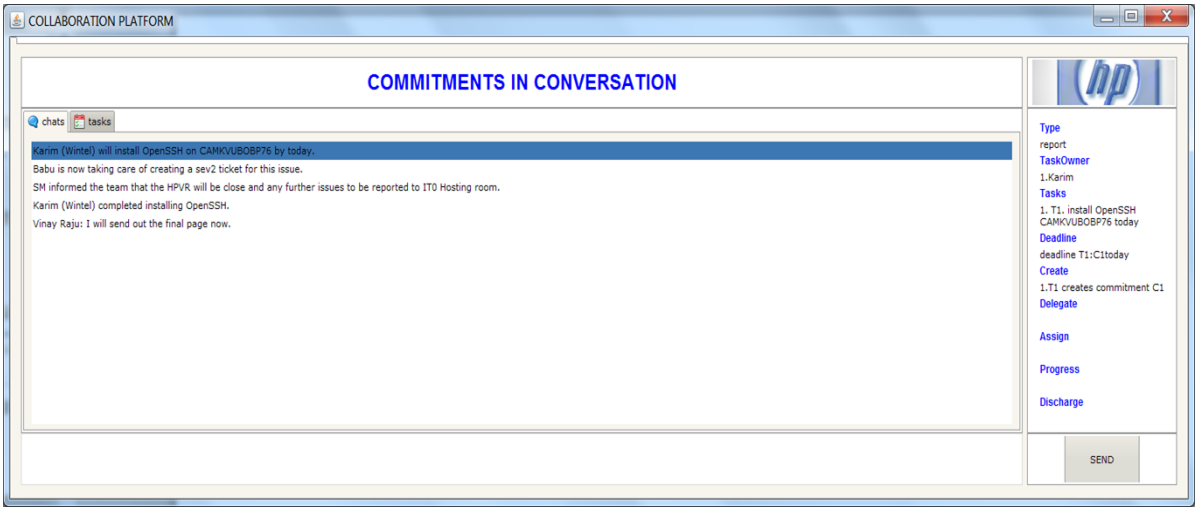

Figure 10: A screenshot of our prototype tool for commitment identification and monitoring.

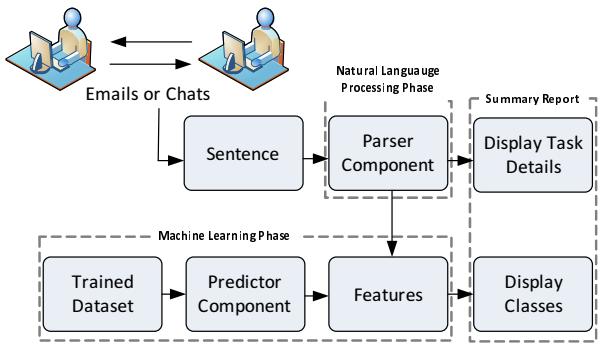

Figure 9: The architecture of our commitment identification and monitoring tool.

7. LIMITATIONS

Our work has several limitations. First, to identify a task structure we depend on the typed dependency feature which does not work accurately when the sentence is either too long or too short. Identifying multiple tasks in a sentence is again difficult as the typed dependency feature does not work well for sentences with conjunctions such as *and* and *but*. Second, for judging the similarity in consequents of two commitments, we rely on comparing words. Third, we rely on the type of a personal pronoun such as first person, second person, and third person and map the pronouns either to a sender or a receiver of an email. Without the presence of a sender and a receiver we cannot determine the debtor and the creditor of a commitment accurately. Fourth, in case of chats we cannot find debtor every time and it is difficult to resolve pronoun such as *you*. Fifth, if a sentence has a missing subject and an object then we cannot determine if *a prior commitment exists*.

8. RELATED WORKS

The research in this space are just emerging. While there has not been previous work on commitment lifecycle identification, there are few works on using NLP and machine learning techniques for identifying action verbs and tasks from texts.

Qadir and Riloff [7] classify sentences from message board posts as commissives, directives, expressives, and representatives. For creating a training model, they extract lexical, syntactic, and semantic features for messages. Using these features, they train their model using the SVM classifier. They obtain 45% precision com-

missives class and 97% for directives. In the identification of commissive and directive commitment creation, we consider *personal pronoun*, *modals*, and *sentence begins with modal/verb*. Based on these and other features and our novel algorithms, we obtain a higher precision for commissive create and directive creation.

Scerri et al. [8] focus on action items in emails and check whether these action items fall under the request, suggest, assign, and deliver classes. For identifying the classes, they use a rule-based classification model. To evaluate their classification model they compare the classes identified automatically with the classes identified by a group of 12 people. Their evaluation results show the precision value of 56% and recall value of 60% respectively. In contrast, first we identify classes for each sentence based on a novel NLP-based algorithm. Next, we apply machine learning to identify classes based on the identied features from the text, which allows us to identify classes in case where it is difficult to capture corresponding rules. In addition, we monitor and discover the lifecycle of a commitment, once it is created.

Purver et al. [6] focus on multiparty meetings and identify task owner, task, deadline, and agreement classes from these meetings. For building a training model in the baseline approach, they label the data manually and train it using SVM classifier. Their result shows a precision of 25%. Apart from the baseline approach they use hierarchical annotation techniques where they identify classes using the NOMOS annotation software and build their model using the Naïve n-gram classifier. The precision values for task description, task owner, deadline, and agreement are 23%, 12%, 19%, and 48% respectively. In contrast, we completely rely on the type-dependency feature to extract task owner, task, and deadline in a sentence. Although the type-dependency extracts features somewhat accurately, it extracts features incorrectly in many cases. Therefore, we emply a machine learning-based approach to identify the structure of a task more accurately.

9. CONCLUSIONS AND FUTURE WORK

With the rise in the informal collaboration and web-based communication tools, the problem of organizing work in the collaborational setting and in the context of people-centric processes becomes more vital for the organizations. To the best of our knowledge, this is is the first work that presents an approach for the automated identification and monitoring of commitment lifecycle from the text of conversations amog people. We introduced an approach for extracting tasks and commitments from sentences in email and chat conversations. Our NLP-base algorithms leverages typed dependency for identifying tasks and its related parameters. Our machine learning approach accurately identifies whether a task indicates a commitment and further identify whether a task progresses a commitment or terminates it. Using our machine learning approach we get high precision for commissive and directive creation on both emails and chats. Our approach also provides promising results for the delegate, discharge and cancel classes. Finally, we provide a tool that implements the approach, and can be used for assisting workers in capturing commitments in collaboration tools.

We have several future directions. First, we plan to improve results for delegation, discharge, and cancellation, specifically in chat conversations. Second, we plan to explore and compare other approaches for identifying tasks from sentences. Third, we plan to consider applying unsupervised machine learning such as Hidden Markov Models (HMM) to this problem as the supervised machine learning approach relies on manual labeling of data, which is tedious to provide. Fourth, another future step is to reconstruct conversations from both emails and chats and then apply our approach.

10. REFERENCES

[1] M. De Marneffe, B. MacCartney, and C. Manning. Generating typed dependency parses from phrase structure parses. *Proceedings of LREC*, 6:449–454, 2006.

[2] A. Fiore and J. Heer. UC Berkeley Enron email analysis. 2004.

[3] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226. Springer Berlin / Heidelberg, 2004.

[4] 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 2010 14th IEEE International Enterprise Distributed Object Computing Conference*, EDOC '10, pages 247–256, Washington, DC, USA, 2010. IEEE Computer Society.

[5] H. R. Motahari-Nezhad, C. Bartolini, S. Graupner, and S. Spence. Adaptive case management in the social enterprise. In *Service-Oriented Computing*, volume 7636 of *Lecture Notes in Computer Science*, pages 550–557. Springer Berlin Heidelberg, 2012.

[6] M. Purver, P. Ehlen, and J. Niekrasz. Detecting action items in multi-party meetings: Annotation and initial experiments. In *Proceedings of the Third International Conference on Machine Learning for Multimodal Interaction*, pages 200–211. Springer-Verlag, 2006.

[7] 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*, pages 748–758, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[8] 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)*, 2010.

[9] J. R. Searle. *Speech Acts*. Cambridge University Press, Cambridge, UK, 1969.

[10] J. R. Searle. A taxonomy of illocutionary acts. In K. Gunderson, editor, *Language, Mind and Knowledge*. University of Minnesota Press, Minneapolis, 1975.

[11] M. P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7(1):97–113, Mar. 1999.

[12] P. R. Telang and M. P. Singh. Enhancing Tropos with commitments. In A. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. K. Yu, editors, *Conceptual Modeling: Foundations and Applications*, volume 5600 of *LNCS*, pages 417–435. Springer, 2009.

[13] P. R. Telang and M. P. Singh. Specifying and verifying cross-organizational business models: An agent-oriented approach. *IEEE Transactions on Services Computing*, 5(3):305–318, July 2012.

[14] T. Winograd. A language/action perspective on the design of cooperative work. In *Proceedings of the 1986 ACM Conference on Computer-supported Cooperative Work*, CSCW '86, pages 203–220, New York, NY, USA, 1986. ACM.

Identifying Business Tasks and Commitments from Email and Chat Conversations

A. K. Kalia[1,2]*, H.R. Motahari-Nezhad[1], C. Bartolini[1], M. P. Singh[2]

[1]Hewlett Packard Laboratories, Palo Alto, CA, United States
{hamid.motahari, claudio.bartolini}@hp.com
[2]NC State University, Raleigh, NC, United States
{akkalia, singh}@ncsu.edu

ABSTRACT

Case management applications and, more generally, people-centric processes involve the definition, resolution and communication of commitments for tasks over channels such as chat and email. Identifying and tracking tasks and commitments can help in streamlining the collaborative work in business environments. However, doing so proves challenging due to the syntactical, grammatical, and structural incompleteness of human conversations over chat and email channels. We present a novel approach to automatically identify tasks and commitment creation, delegation, completion, and cancellation in email and chat conversations, based on techniques from natural language processing and machine learning domains. We discover tasks and related parameters from the text of conversations, identify when a commitment to a task emerges and find the state changes of a commitment based features extracted from the text of the conversations. We have developed a prototype and evaluated our approach using real-world chat and email datasets. Our experiments shows high precision for create class i.e., 90% in emails (Enron email corpus) and 80% in a real-world chat dataset and also provides promising results for discharge, delegate, and cancel classes.

1. INTRODUCTION

Many processes in organizations are people-driven, and are managed in a collaborative and conversation-oriented manner. Conversations around people-centric processes involve defining and coordinating tasks through commitments among workers over informal channels such as email and chat. Typical examples of such conversations include the handling of insurance claims and IT incidents. For instance, in IT incident management, incident reports are assigned to help desk workers and in some cases to specialized IT experts. The handling usually proceeds through team collaboration and communication over email and chat, in addition to keeping record in specialized systems. Keeping track of all agreed-upon tasks and commitments made during a conversation is a daunting task, and a major source of inefficiencies [4], as the size and number of interactions and processes that a worker is involved in increases.

We investigate the problem of identifying tasks (and related parameters) and commitments from multiparty textual conversations over email and chat. Commitments provide a level of deep modeling that facilitates appropriate progression of tasks. A commitment is created when a worker becomes responsible for a task, whether by volunteering or being directed. Once created, a commitment may be completed, delegated, or canceled. Identifying and monitoring commitments automatically in human conversations is challenging as these conversations are ill-structured, are not necessarily grammatically correct, and contain domain-specific information. For example, such as IP traces, IT-related conversations may contain codes and error logs. In addition, as conversations grows in length and the number of workers involved increases, it becomes difficult to determine the status of the commitments made by any of the workers. Moreover, a single sentence may carry information about several commitments, or several state changes of a given commitment.

The few existing works dealing with tasks or commitments in natural language processing (NLP) [5, 6, 7] consider only part of the problem, i.e., identifying action verb classes in a source such as message board, email corpus, or chat logs. They do not support the identification of commitments, monitoring their progression or lifecycle. In addition, they report low accuracy results.

We introduce a novel approach for the identification of tasks and commitments based on the speech act theory, people-centric processes, and machine learning. We summarize the novel contributions of this paper as follows:

- Define commitments and their lifecycle in the context of people-centric processes and case management applications [**?**].
- Present an NLP-based algorithm by leveraging typed dependency [1] for identifying a task and its related parameters (owners, deadline, actions) from a chat or email message.
- Determine whether the communication around a task signals the creation, delegation, cancellation, or discharge of a commitment, by extracting selected features from conversations and a classification-based approach. Analyze complex sentence structures to discover cases where a sentence contains multiple task definitions or several lifecycle changes of a commitment such as creation and delegation.
- Develop an automated agent that monitors conversations to identify the tasks and progression in a commitment, and non-

*The main work was done while the first author was a summer intern at HP Labs, Palo Alto

intrusively presents them to case workers who determine whether to accept its suggestions.

We have experimentally validated our approach on real-world datasets of email and chat conversations. Our approach yields significantly better accuracy than existing work for task and commitment identification, and performs well for identifying the delegation, cancellation, and discharge of a commitment, which have not been studied by others.

The paper is structured as follows. Section 2 discusses the key background topics on commitments. Section 3 defines tasks and commitments in the context of conversations around people-centric processes. Section 4 describes our approach to identifying commitments from email and chat conversations. Section 5.3 explains our dataset, experimentation and the evaluation results. Section 6 highlights the limitations of our approach. Section 7 discusses related works. We conclude and discuss future directions in Section 8.

2. BACKGROUND

Several researchers have studied how people converse with each other to create tasks and commitments and collaborate with each other. We review speech acts [8], language/action perspective [13], and commitments [10] as key concepts to build the foundation for the approach presented in this paper. These techniques are foundations for providing automated support for conversation-oriented methods for conducting people-centric processes [4].

2.1 Speech Acts

Searle [8, 9] classified illocutionary acts into five classes: commissives, directives, representatives, expressives, and declarations. A message is classified as a *commissive* if the sender of the message promises to take an action in the future. A message is classified as a *directive* when the speaker intends the receiver to do something. A message is classified as a *representative* if the sender commits to the truthfulness of the message. A message is classified as an *expressive* when the sender expresses his or her psychological state. A message is classified as a *declaration* when the sender of the message brings about a change in the status of the referred object or objects. Searle theory provides the basic idea of identifying commissive and directive actions in people's conversations, however, no work is reported on how commissive and directive actions progress in conversation.

2.2 Language/Action Perspective

Winograd [13] extends speech acts to understand human cooperative activity as conversations. In this model, a message in a conversation is identified commissive when it is either *requested*, *offered*, or *counter-offered*, let's say from a party (A) to another party (B). The conversation progresses when B accepts the offer from A and assert A that the conditions are met. Now, if A declares that he or she is satisfied, the conversation reaches a *completion* state.

2.3 Commitments

Unlike Winograd's approach that captures every request as a commitment, Singh's model of commitments [10] capture business relationships between any two entities. These entities can be either employees within a company or the companies themselves. Specifically, commitments denote business meanings underlying the interactions between these business entities. In this model, a commitment is a conditional business relationship directed from a debtor to a creditor, which can be formalized as

$$C(\text{DEBTOR}, \text{CREDITOR}, antecedent, consequent).$$

The formula shows that the debtor is committed to bringing about the consequent for the creditor provided the antecedent holds. When a debtor offers a commitment to a creditor, the commitment is created and becomes active. When the antecedent is brought about, the commitment is detached and 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 and the antecedent may be omitted:

$$C(\text{DEBTOR}, \text{CREDITOR}, \top, consequent).$$

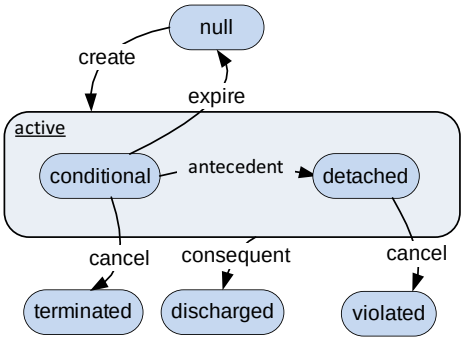

Figure 1: The life cycle of a commitment [12].

Telang and Singh [12] present the commitment life cycle as shown in Figure 1. According to Figure 1, a commitment transits 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 gets created when a debtor voluntarily offers to do a task or when he is assigned to do a task.
- *detach(c)* detaches a commitment. A commitment gets detached if a condition or an antecedent present for a commitment holds 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 debtor delegates his commitment.

3. TASKS AND COMMITMENTS IN TEXT

In this section, we discuss tasks and commitments in connection with textual conversations. To explain the concepts, we use examples from the Enron email corpus and a chat corpus from an HP IT incident management application. The email corpus contains over 250,000 emails sent or received by over 87,000 people. It consists of emails exchanged by Enron employees in the time leading up to the Enron bankruptcy that were revealed as part of the Federal investigation of Enron. The chat corpus contains conversations related to HP IT incident management systems. The corpus is collected from the logs of chat conversations among workers in a customer facing incident management system for big IT accounts. We extracted logs from the months of May and June 2012 and analyzed them for our work.

3.1 Task

A task is a business activity that is either pre-defined (part of a best practice process) or created on-the-fly by people in a conversation [4, 11]. We represent a task as T and define it as

$$T(\text{SUBJECT}, \text{OBJECT}, action)$$

In this definition, a **subject** is a business entity that performs the action. An **object** is a business entity for whom an action is being performed by a subject. An **action** is a business activity performed by a subject. An action can be a disjunction or a conjunction of subactions. And finally, a **deadline** represents the time-out condition for an action to be performed. If the time-out occurs, then we consider the action to be expired. Consider an example from the Enron corpus where Kim sends an email to John: *I will pick it up tonight*. Here, the subject is Kim and the object is John. The action is *pick it up* and the deadline is *tonight*. We can formally represent this task as T1 = T(KIM, JOHN, *pick it up*).

3.2 Create Commitment

We adopt the Singh's model of commitments [10], reviewed in Section 2.3, as an action that is performed by a debtor for a creditor. In our context, we map a task to a commitment as follows:

***Definition** 1.* **(Mapping a task with a commitment)** *If a task T is a commitment C, then subject ∈ T is debtor ∈ C, object ∈ T is creditor ∈ C, action ∈ T is consequent ∈ C, and deadline ∈ T is consequent_timeout ∈ C.*

Consider an example from the Enron dataset where Earl Chanley sends the following email to Jo Williams *We will expedite materials and installation in an attempt to meet the target date*. To extract the structure of a commitment from the sentence, first, we identify a task from it and formally write it as T1 = T(EARL CHANLEY, JO WILLIAMS, expedite materials ∧ expedite installation). Second, we map T1 to a commitment C1 = C(EARL CHANLEY, JO WILLIAMS, ⊤, expedite materials ∧ expedite installation) as the email indicates a commitment from Earl (as subject) to Jo (as object). By the classification of Searle's illocutionary acts [9] the above examples represents an offer from Earl to Jo. Therefore we define a commissive creates as

***Definition** 2.* **(Commissive create)** *The creation of a commitment C is commissive when the debtor voluntarily offers to perform the consequent for the creditor*

Similar to a commissive create, a commitment can be a directive. We define a directive create as:

***Definition** 3.* **(Directive create)** *The creation of a commitment C is directive when the creditor delegates the consequent to the debtor*

Consider an example of a directive create from the Enron corpus where Steven Schleimer sends the the following email to Kim Watson *Please review and send along to your attorneys as soon as possible*. We formally write the email as C(KIM WATSON, STEVEN SCHLEIMER, ⊤, review ∧ send).

In our machine learning approach, we label sentences as either *ccreate* or *dcreate* based on whether the sentence indicates a commissive or a directive respectively.

3.3 Discharge Commitment

A commitment is discharged when the debtor executes the consequent thereby making it true. An example from the Enron dataset is one where Kim sends an email to Dorothy with the following message, *I will also check with Alliance Travel Agency (formerly Travel Agency in the Park) to see what they may be able to do for us*. The message indicates the creation of a commitment from Kim to Dorothy and can be represented as C1 = C(KIM WATSON, DOROTHY MCCOPPIN, ⊤, check with Travel Agency). In following, Kim sends another email to Dorothy with the following message, *I checked with our Travel Agency and they cannot secure cheaper tickets*. The task T2 from this email T2 = T(KIM, MCCOPPIN, checked with Travel Agency) discharges C1 as Kim checked with the travel agency.

3.4 Delegate Commitment

A commitment is delegated when its debtor outsources it to a new debtor. There can be two cases of delegation. In the first case, the new debtor does not know about the old creditor. We define the delegation as

***Definition** 4.* **(Unknown creditor delegation)** *When a debtor delegates his or her commitment C1 to a debtor', then debtor ∈ C1 is the creditor' ∈ C2 and both the commitments C1 and C2 can be written as*

$$C1 = C(debtor, creditor, \top, consequent)$$

$$C2 = C(debtor', debtor, \top, consequent)$$

Consider an example from the Enron dataset where Gregory sends an email to Glen with the following message, *Please take a few moments to review the same and let me know your thought*. In this email, Gregory delegates a commitment to Glen for reviewing something. We formally represent the commitment as C1 = C(GLEN HASS, GREGORY KLATT, ⊤, review statements). In the follow up, Glen sends another email to Steven with the message, *This appears to be 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*. In this email, Glen delegates his commitment to Steven thereby creating another commitment from Steven to him. This commitment can be formalized as C2 = C⟨STEVEN HARRIS, GLEN HASS, ⊤, review statements⟩. As, we can see this commitment has a new debtor (debtor') as Steven while the creditor' Glen is the debtor of the previous commitment.

Now, in the second case of the delegation, the new debtor is committed to the old creditor. We define the delegation as

***Definition** 5.* **(Known creditor delegation)** *When a debtor delegates his or her commitment C1 to a debtor', then creditor ∈ C1 is the creditor' ∈ C2 and both the commitments C1 and C2 can be written as*

$$C1 = C(debtor, creditor, \top, consequent)$$

$$C2 = C(debtor', creditor, \top, consequent)$$

Consider an example from the Enron dataset where Robert sends an email to Drew with the following message, *Please let me know if your business unit has any problem with this course of action*. In this email, Drew creates a commitment with Robert for letting him know something. The commitment is formalized as C(DREW, ROBERT, ⊤, inform about business unit having any problem). In following up, Drew sends another email to Kim; *Please review this message and advise Robert Williams of any relevant information*. In this email, Drew delegates his commitment to Kim thereby creating another commitment from Kim to Robert. This commitment

can be formalized as C(KIM, ROBERT, $\top$, provide information). As we can see, this commitment has a new debtor as Kim while the creditor is still the same i.e., Robert.

3.5 Cancel Commitment

A commitment is canceled when the debtor of the commitment terminates the commitment. An example from the Enron dataset is Diane sends an email to Kimberley, *Can you please provide me the details about the amounts from prior months by this Friday?*. In this email, Diane delegates a commitment to Kimberley that can be represented as C1 = C(KIMBERLEY, DIANE, $\top$, provide the information). Later, Kimberley sends an email back to Diane with the message, *I cannot not give you the details by Friday*, thereby canceling her existing commitment toward Diane.

4. IDENTIFYING TASKS AND COMMITMENTS

Our process for the identification of commitments from emails and chat conversations is depicted in Figure 2. We first extract sentences from the text of conversations. We then identify the tasks and then candidate commitments or changes in the lifecycle of commitments using a combined NLP-based and machine learning approach. Using NLP and a set of heuristic-based rules applied on features extracted from the text of conversations, we identify certain tasks and commitments. However, the application of NLP-based rules is limited to identifying pre-determined classes and patterns of tasks and commitments. We augment our approach with a supervised machine learning approach for the identification of commitments and their lifecycle. This helps in the identification of commitments for which their various expressions and forms in the natural language may not be captured in patterns and rules.

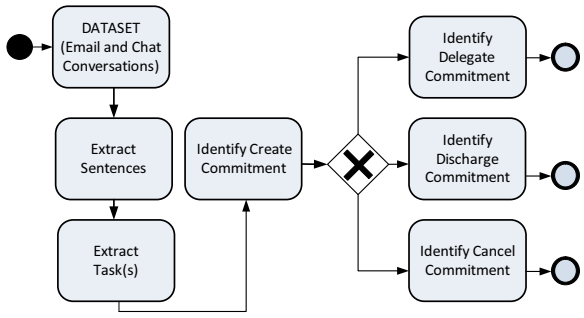

Figure 2: Process followed to identify commitments.

For identification of a task T from a sentence, we check whether the sentence has a subject, an object and an action verb. Although it sounds simple, the identification of tasks in a sentence is non-trivial. For example, changing from active to passive voice results in shifting positions of words that indicate subject, object, or the action. To deal with this issue, we chose to parse the sentences using the typed dependency method [1] which outputs the relations between individual words in a sentence, and is largely independent of the exact sentence structure. A relation between any two words is defined as

***Definition** 6.* **(Relation)** *A relation is a triplet of the name of the relation, governor, and dependent where the governor and dependent are words from a sentence and it can be represented as*

$$relation(governor, dependent)$$

Marneffe *et al.* [1] represent such relationships in a hierarchical manner with the most generic relation as the root. For example, a relationship can start with an *arg* (argument) and it can branch into the *subj* (subject) and the *comp* (complement) relationship. Consider a sentence, *Diwalkar reports the team will do hardware investigation on EMEGHP151*. In the sentence, the triplets are root (root, reports), nsubj (do, team), nsubj (reports, Diwalkar), aux (do, will), prep_on (do, EMEGHSP151) det (team, the), dobj(do, investigation), and nn(investigation, hardware). (time, this). Figure 3 shows these triplets in a graph In the graph, we can see that there is an *nsubj* relation in which *team* is the *dependent* and *do* is the *governor*. This relation suggests there is a subject in the sentence is *team* and his action is *do*. Even a sentence can have multiple subjects. As we can see, there is another *nsubj* relationship in which *Diwalkar* is the subject and his action is *reports*.

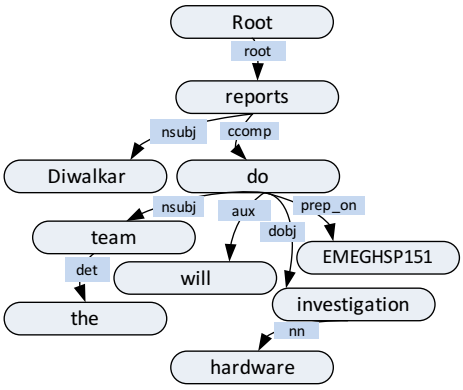

Figure 3: Typed dependencies derived from a sentence "Diwalkar reports the team will do hardware investigation on EMEGHP151" chosen from the chat corpus.

4.1 Identifying Conversations and Sentences

We preprocess the Enron and the HP IT incident management datasets to make them suitable for parsing and extracting features. Since the email and chat dataset types 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 features. Unlike, emails we do not prepare conversation threads for chat conversations as they are already listed chronologically.

4.2 Extracting Features

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

- **Co-reference 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 co-reference 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 find the debtor for a commitment.

- **Named entity resolution** (NER), identifies a noun whether it's 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 to tag a word with a part of speech. The POS tags help to identify the type of personal pronoun for a subject and the state of the verb associated with the subject so as to identify the debtor of a commitment and the state of a commitment, respectively. The present tense of the verb indicates that the commitment is either a commissive or a directive whereas the past tense indicates the commitment is discharged. Examples of POS tags used in our work include NN (for nouns), NNP (pronouns), PRP (personal pronouns), VB (present tense verbs), VBD (past tense verb), VBZ (3rd person singular verb), VBN (past participle) and VBP (non-3rd person singular).

- **Typed dependencies extraction** As discussed in Section 4, a typed dependency relates words in a sentence and gives a clue as to its logical structure.

Before explaining the algorithms for identifying tasks and commitments, we briefly mention the features in the feature vector that are used to train classifiers for email and chat datasets. 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. For example, the sentence *He will handle the issuance of the LC*. has a modal verb *will* that indicates the creation of a commitment. An *action verb* indicates whether a commitment is present in a sentence. For example, *I can imagine that your family reunions are just a hoot!* is not a commitment because the verb *imagine* is not an action verb. The *present tense* signals the creation, delegation, or cancellation of a commitment. In the following example: *She and I will get together on the results of these meetings.* that suggests a create commitment, *handle* and *get* are in the present tense. And, the *past tense* signals the discharge of a commitment. In the following example: *I have reviewed the list you sent me regarding items you would like to see in the Data Room.* that suggests a discharge commitment has has the action verb *reviewed* in the past tense.

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. For example, in the sentence *It will be posted today and the policy will go into effect for Friday's gas day.* indicates *today* and *Friday* as the deadlines. The prior *creation of a commitment* is a prerequisite for discharge, delegation, and cancellation if the create commitment already exists. And a *delegation signal* is identified when a debtor occurring as a creditor indicates that the debtor delegates an existing commitment to a new debtor. The *negative verb* indicates the presence of a canceled commitment. For example, in the sentence *I cannot give you the amount details not give* indicates a negative verb.

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, third). The *bigram of a modal verb and a second person PRP* indicates a directive creation. For example, in the sentence *Can you help me with the following outstanding items relating to the Info Memo*, the bigram *Can you* indicates directive creation. The *bigram of a first person PRP and a modal verb* indicates a commissive. For example, in the sentence *We will expedite materials and installation*, the bigram *We will* indicates commissive creation.

The *bigram of 'please' and an action verb* indicates a directive. For example, in the sentence *Please review and send along to your attorney as soon as possible*, the bigram *please review* indicates a directive commitment creation. And, a *question mark* in a sentence indicates a directive commitment creation.

4.3 Identifying Tasks

To identify a task from a sentence, we first extract features discussed in Section 4.2. The features are input to Algorithm 1 to identify subject, object and the action that the subject performs for a candidate task. Our algorithm slightly differs between emails and chats especially in extracting the subject and the object as in case of chats it is more difficult to find them than in emails. In case of emails, it is easier to find them because the metadata provides information such as sender's and receiver's names. In case of chat messages, it is difficult because we only get the chat initiator name as the metadata and the co-reference resolution does not work well for resolving pronouns such as *you* in multiparty conversations.

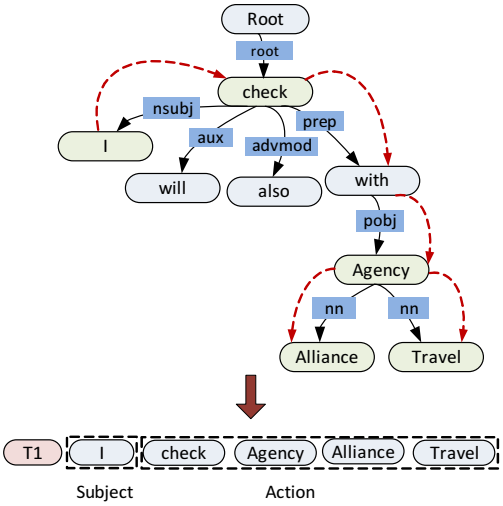

Figure 4: Steps to identify task from a email sentence "I will also check with Alliance Travel Agency".

Once we parse a sentence, we get a typed dependency array as in the example shown in Figure 4. We define the typed dependency array as

***Definition** 7.* **(Typed dependency array)** *The typed dependency array consists of relations derived from a sentence using the typed dependency method. The typed dependency array is represented in terms of relations as, typed dependency array = {relation$_1$, relation$_2$, relation$_3$, ... }.*

In the above definition, a *relation* can be an *nsubj*, *aux*, *advmod*, and so on. In a typed dependency array, we look for the *nsubject* relation and check if the dependent is a *valid subject* and the governor is a *valid verb*. We define subject validity as

***Definition** 8.* **(Valid subject)** *A dependent is considered a valid subject if the POS tag associated with the dependent is NNP or NER resolves the subject as a* PERSON *or an* ORGANIZATION.

We define verb validity as

***Definition** 9.* **(Valid verb)** *A governor is considered valid if the POS tag associated with the governor is either VB, VBD, VBP, VBZ, or VBN and if the governor is an action verb.*

We define an action verb as

***Definition** 10.* **(Action verb)** *The verb that expresses an action or doing something.*

By checking for an action verb, we eliminate words that are verbs but do not express actions. If both the governor and the dependent are valid, we store the dependent as the subject and the governor as the action for the subject. We extract the action details using the action verb by finding its dependencies in the array of triplets by looking for nouns or verbs associated with the action verb. Algorithm 1 shows the steps to extract subject, object, and action from a sentence in emails. In this algorithm if the POS tags associated to the dependent is PRP first person (I, we) then the subject is the sender of the email. In case of chats, the subject is the chat initiator. If the POS tag associated with the dependent is PRP second person (you), then the receiver is the subject of the task. In case of chats, it is difficult to resolve *you* in a multiparty conversations. If the POS tag associated with the dependent is PRP third person, then the approach for both emails and chats is the same. We consider the dependent as the subject. If the POS tags associated with the dependent is personal PRP (he, she, they) then we resolve them with co-reference resolution.

Algorithm 1: extract task($sentence$)

```
typed dependency array ← get typed dependencies(sentence);
foreach nsubj ∈ typed dependency array do
    if get POS(dependent ∈ nsubj) is PRP first person then
        subject ← sender of the email;
        object ← receiver of the email;
    else
        if get POS(dependent ∈ nsubj) is PRP second person then
            subject ← receiver of the email;
            object ← sender of the email;
    else
        if get POS(dependent ∈ nsubj) is PRP third person then
            subject ← dependent;
            object ← extractObject(sentence);
    if valid subject(dependent) ∧ valid verb(governor) then
        action ← extract action details(governor);
```

Figure 4 represents the typed dependency array for the sentence *I will also check with Alliance Travel Agency*. We extract the *nsubj* relationship i.e., nsubj (check, I). The subject is *I* and the action verb is *check*. Once we extracted the subject and the action, we extract nouns or verbs related to action such as *Agency*, *Alliance*.

4.4 Identifying Create

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 in a task has a relationship with a modal verb and the word *please*. A modal verb can be defined as

***Definition** 11.* **(Modal verb)** *A modal verb is a word that expresses possibility, likelihood, or obligation. Words that indicate modality are will, shall, can, could, would, should, may, might and must.*

The word *please* indicates a request or a delegation of a task. If the verb has a relation with these words (modal or please), then we label the task as commitment creation and store the subject as the *debtor*, the object as the *creditor*, and the action as the *consequent* respectively of the commitment. Also, as discussed in Section 3.2, commitments can be created in two ways: the commissives or directives. We mark a sentence as a commissive when an action verb is following a modal verb. We mark a sentence as a directive when a second person personal pronoun such as *you* is following a modal verb or an action verb following the word *please*. Algorithm 2 summarizes the method for identifying commitments from sentences.

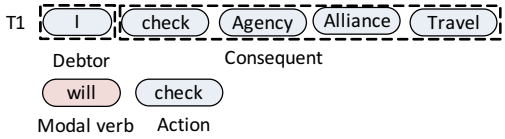

Figure 5: Steps to identify a commitment from a email sentence "I will also check with Alliance Travel Agency".

In Figure 5, the action verb *check* is in the present tense (VB) and has a relationship with a modal verb *will*. Therefore, the task is considered as creating a commitment. Since here the action verb follows the modal verb, we consider it a commissive. In Algorithm 2, we check if the action is in present tense (VB). Then, we check if there is a relation that associates the action with a modal verb or *please*.

Algorithm 2: identify create commitment(sentence)

```
T1 ← extract task(sentence);
commitment ← ⊥;
if getPOS(action verb ∈ T1) is VB then
    foreach relation ∈ typed dependency array do
        if (dependent ∈ relation is action) ∧ (governor ∈ relation is (modal verb ∨ please)) then
            commitment ← ⊤;
        else
            if (governor ∈ relation is action) ∧ (dependent ∈ relation is (modal verb ∨ please)) then
                commitment ← ⊤;
```

4.5 Identifying Delegate

We discussed in Section 3.4 that there are two kinds of delegation. In one, the old creditor is unknown to the new debtor, whereas, in another, the old creditor is known to the new debtor. Consider Figure 6(a) that shows an example where the new debtor Steven does not know about the old creditor Gregory. Similarly, consider Figure 6(b) where the new debtor Kim is committed to the old creditor Robert.

To identify delegation in sentences, we check a commitment C2 created after another commitment C1. To find the unknown creditor delegation, we check if the debtor in C1 is the creditor in C2 and if the consequents in C1 and C2 are the same. To find the known creditor delegation, we check if the creditor in C1 is the creditor in C2 and if the consequents in C1 and C2 are the same. To match the consequents in the commitments, we check whether the action verbs in both commitments are the same or related as *synonyms*, *hypernyms*, or *hyponyms*. If they are the same or related, we check whether nouns in both the commitments are the same or related using the co-reference resolution. Finding delegation in case of email is easier than chats as we know in advance the debtor and creditor of the commitment based on the sender and receiver information.

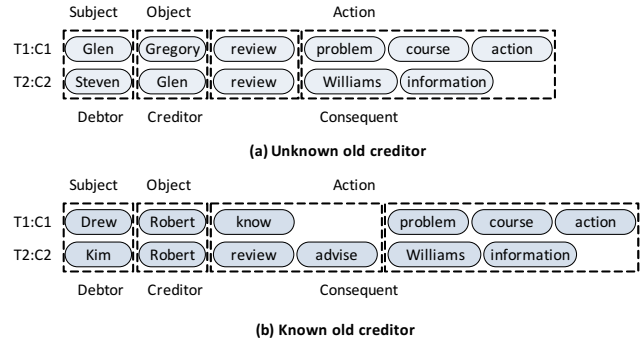

Figure 6: Steps to identify delegate.

Unlike emails, in chats, finding delegation is difficult as sometimes it is extremely hard to find the creditor of a commitment.

Algorithm 3: identify delegate commitments(sentence)

```
C2 ← identify create commitments(sentence);
delegation ← ⊥;
unknown creditor ← ⊥;
known creditor ← ⊥;
foreach C1 ∈ commitment array do
    if debtor ∈ C1 is creditor' ∈ C2 then
        unknown creditor ← ⊤;
    else
        if debtor ∈ C1 is creditor' ∈ C2 then
            known creditor ← ⊤;
    if action verb ∈ C1 is action verb ∈ C2 then
        if nouns ∈ C1 is nouns ∈ C2 then
            if unknown creditor ∨ known creditor then
                delegation ← ⊤;
```

4.6 Identifying Discharge

If an identified task is in the past tense, it may signals a discharge commitment, and we need to compare it with existing commitments. For greater clarity, let us consider an example of a commitment C1 and a task T2.

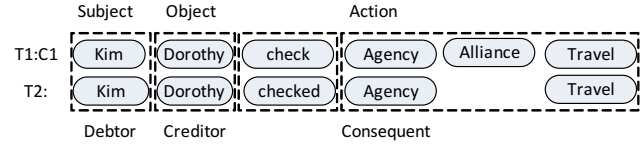

Figure 7: Steps to identify discharge.

To check if T2 discharges C1, we compare the subject and the object in T2 with the debtor and the creditor in C1 respectively. If they are the same, we compare their action verbs. We check whether the action verb in T2 is in the past tense (VBD). Then we compare the action verb in T2 by converting it into its base form and trying to match with the action verb in C1. If both the verbs are the same or related as either *synonyms*, *hypernyms*, or *hyponyms*, we compare the nouns in both the tasks. If they are the same or related, we mark T2 as discharging C1. Figure 7 clearly shows that the debtor and the creditor in both C1 and T2 are the same. The main action verb in T2 is *checked* and it is in the past tense. Therefore, we convert it into its base form *check* and compare it with the action verb in C1. Note that the base form of a verb is the simplest form in which it appears in a dictionary without any ending. Since they are the same, we compare the nouns in C1 and T2. We see that the nouns are related. Therefore, we mark T2 as discharging C1. Algorithm 4 describes these steps.

Algorithm 4: identify discharge commitments($sentence$)

```
T2 ← extract task(sentence);
discharge ← ⊥;
foreach C1 ∈ commitment array do
    if (subject ∈ T2 is debtor ∈ C1) ∧ (object ∈ T2 is
    creditor ∈ C1) then
        if (getPOS(action verb ∈ T2) is VBD) ∧
        (getBaseVerb(action verb ∈ T2) is action verb ∈ C1)
        then
            if nouns ∈ T2 is nouns ∈ C1 then
                discharge ← ⊤;
```

4.7 Identifying Cancel

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*.

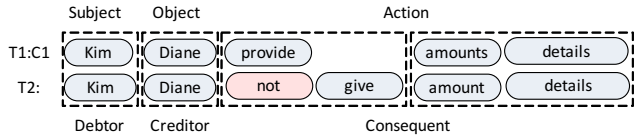

Figure 8: Steps to identify cancel.

Figure 8 describes an example of a commitment C1 and a task T2. Once we find that the debtor and the creditor in C1 and T2 are the same, we compare their main action verbs and nouns respectively. If they are the same, we check whether the action verb is a negative verb. Note that a verb considered as negative it has a relation with a negative word such as *not*. Algorithm 5 describes the above steps.

Algorithm 5: identify cancel commitments(sentence)

```
T2 ← extract task(sentence);
cancel ← ⊥;
foreach C1 ∈ Commitment Array do
    if (subject ∈ T2 is debtor ∈ C1) ∧ (object ∈ T2 is
    creditor ∈ C2) then
        if getBaseVerb(action verb ∈ T2) is action verb ∈ C2
        then
            if nouns ∈ T2 is nouns ∈ C1 then
                if negative verb(action verb ∈ T2) then
                    cancel ← ⊤;
```

5. EVALUATION AND PROTOTYPE

To validate our approach, we use two methods: (1) automatic labeling of data (sentences) using Algorithms 2, 3, 4, and 5, and (2) manually labeling a subset of data and using them for training and testing our approach. We applied three machine learning classifiers and evaluated them.

5.1 Data

For evaluation and validation purpose, from the Enron email corpus, we selected 4161 email sentences that were exchanged between Kimberly Watson, an employee of Enron, and more than 50 people which includes her co-workers at Enron, clients, friends, and family members. The emails were collected and prepared by combining the data from text files provided by CMU [3] as well as from a database dump [2] provided by UC Berkeley. We combine the data from both so as to reduce the missing information. For the chat data, we selected 271 conversations from HP IT incident management logs comprising of 7154 sentences.

5.2 Labeling Data

Once the features are extracted for each sentence from the dataset as give in Section 4.2, we had two annotators to label the sentences. We cross checked the results for any possible conflict resolution. We resolved conflicts by allowing two annotators to collaborate and discuss their labels for sentences. Next, we ran classifiers the three classifiers of Naïve Bayesian (NB), Logistic Regression (LR), and Support Vector Machines (SVM) and used 10-fold cross validation to produce results.

We annotated 4161 email and 7154 chat sentences. The two independent annotators achieved overall an interrater agreement (kappa score) of 0.83 for both email and chat. Table 3 shows the distribution of the email and chat sentences as annotated.

5.3 Results

Table 1 and Table 2 represent results for email and chat data respectively, using NB, LR, and SVM classifiers and ten-fold cross validation. In both the tables, we represent C-create as commissive create, D-create as directive create, P as precision, R as recall and F as F-measure.

Table 3: Distribution in email sentences.

Classes	Email	Chat
Commissive create	342	532
Directive create	162	214
Discharge	38	250
Cancel	7	16
Delegate	12	12
None	3540	6130

First, we considered one of the simplest probabilistic text classification approach, Naïve Bayes (NB). The NB classifier approach assumes that attributes in consideration are independent of each other. Using NB, for emails, our results show high precision for commissive creation (84%) and directive creation (81%) while low precision for delegation (32%), discharge (21%), and cancellation (0%). In case of chats, we obtain slightly lower precision for commissive creation (73%) than email, however, we obtain higher precision for directive (85%) and discharge (60%). The precision remains same for cancel (0%). For delegate, the precision (0%) was low compared to precision for email.

Second, we used the Logistic Regression (LR) classifier. Using LR, in emails, we obtain significantly high precision values for commissive (90%), directive (92%) compared to NB while low precision values for delegate (64%), discharge (27%), and cancel (0%). In case of chats, LR performs better than NB for commissive creation and cancellation with precision of 80% and 22% respectively. However, results for other classes are lower, i.e., 74% for directive creation, 64% for discharge, and 0% for delegate. Overall, the results for chat using LR are lower compared to emails except discharge and cancel.

Third, we used the Support Vector Machine (SVM) classifier. Using SVM, in emails, we obtain significantly higher precision, compared to NB and LR, for commissive creation (87%), directive creation (94%), discharge (100%), and delegation (86%) while same precision for cancellation (0%). For chats, using SVM, we obtain lower precision than emails for commissive creation (79%), directive creation (73%), and discharge (63%). However, the f-measure and recall value for discharge is higher in chats than emails. Again, this is due to the higher percentage distribution of discharge in emails and chats.

Overall we obtain high precision, recall and f-measure for a commissive creation and a directive creation for both emails and chats using NB, LR, and SVM. The results for both the classes are high because both the classes are independent of each other and the distribution of these classes are high in both the datasets. The results for other classes are low because other classes depends on the prior existence of create commitment classes and it is difficult to find this specific feature automatically. Compared to discharge and delegate the results for delegate is higher in case of emails because we can easily find out the debtor and the creditor of a commitment based on the sender's and receiver's information. The results for discharge in email is low because the percentage of distribution of discharge is extremely low as lot of tasks indicating discharge are executed, however, are never mentioned in emails. In case of chat, the precision for discharge is higher because the distribution of discharge is high as people in chat conversations immediately report their progress. However, the overall percentage is low for both emails and chats because it is difficult to compare consequent by matching verbs and nouns. For emails, we find a high precision value 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 got 0% precision in emails and 22% in chats, This is because, as we said earlier, it is difficult to find out the prior existence of a commitment as well find out the negative words associated with the action verb.

We evaluated our training model on independent test datasets. Our test datasets contains 1326 email and 2299 chat sentences. For emails we used SVM and for chats we used LR.

Table 4: Evaluation on test datasets using SVM for emails and LR for chats 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

5.4 Prototype Tool

Figure 9 represents the architecture of our interactive tool for commitments identification and tracking. The tool offers an agent that can be pluggined into online chat messengers. The agent inside identifies discovered commitments in a panel for verification and confirmation to conversation participants. When a person sends a chat message, it parses the sentence using our *parser component*

Table 1: Results for emails.

	C-create			D-create			Discharge			Cancel			Delegate			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 2: Results for chats.

	C-create			D-create			Discharge			Cancel			Delegate			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

and extract potential *task details* and *features*. Using our predictor component and a trained model we identify the classes based on the extracted features. The parser and predictor components are Java-based. We use the Stanford parser and Weka libraries to parse and train datasets respectively. We predict classes using the SVM classifier. Finally, the display component produces the summary of all tasks and commitments based on chats. Figure 10 shows a screenshot of the prototype tool.

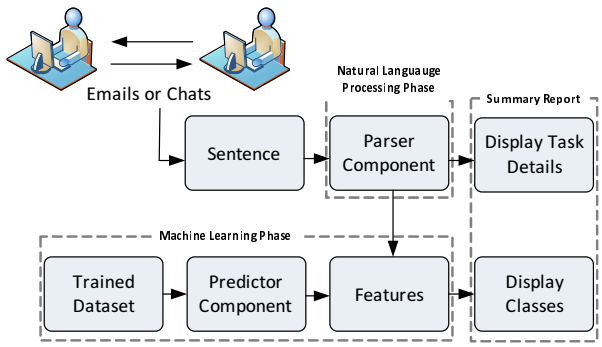

Figure 9: The architecture of our commitment identification and monitoring tool.

6. LIMITATIONS

Our work has several limitations. First, to identify a task structure we depend on the typed dependency feature which does not work accurately when the sentence is either too long or too short. Identifying multiple tasks in a sentence is again difficult as the typed dependency feature does not work well for sentences with conjunctions such as *and* and *but*. Second, for judging the similarity in consequents of two commitments, we rely on comparing words. Third, we rely on the type of a personal pronoun such as first person, second person, and third person and map the pronouns either to a sender or a receiver of an email. Without the presence of a sender and a receiver we cannot determine the debtor and the creditor of a commitment accurately. Fourth, in case of chats we cannot find debtor every time and it is difficult to resolve pronouns such as *you*. Fifth, if a sentence has a missing subject and an object then we cannot determine if *a prior commitment exists*.

7. RELATED WORKS

The research in this space are just emerging. While there has not been previous work on commitment lifecycle identification, there are few works on using NLP and machine learning techniques for identifying action verbs and tasks from texts.

Qadir and Riloff [6] classify sentences from message board posts as commissives, directives, expressives, and representatives. For creating a training model, they extract lexical, syntactic, and semantic features for messages. Using these features, they train their model using the SVM classifier. They obtain 45% precision commissives class and 97% for directives. In the identification of commissive and directive commitment creation, we consider *personal pronoun*, *modals*, and *sentence begins with modal/verb*. Based on these and other features and our novel algorithms, we obtain a higher precision for commissive create and directive creation.

Scerri et al. [7] focus on action items in emails and check whether these action items fall under the request, suggest, assign, and deliver classes. For identifying the classes, they use a rule-based classification model. To evaluate their classification model they compare the classes identified automatically with the classes identified by a group of 12 people. Their evaluation results show the precision value of 56% and recall value of 60% respectively. In contrast, first we identify classes for each sentence based on a novel NLP-based algorithm. Next, we apply machine learning to identify classes based on the identied features from the text, which allows us to identify classes in case where it is difficult to capture corresponding rules. In addition, we monitor and discover the lifecycle of a commitment, once it is created.

Purver et al. [5] focus on multiparty meetings and identify task owner, task, deadline, and agreement classes from these meetings. For building a training model in the baseline approach, they label the data manually and train it using the SVM classifier. Their result shows a precision of 25%. Apart from the baseline approach they use hierarchical annotation techniques where they identify classes using the NOMOS annotation software and build their model using the Naïve n-gram classifier. The precision values for task description, task owner, deadline, and agreement are 23%, 12%, 19%, and 48% respectively. In contrast, we completely rely on the type-dependency feature to extract task owner, task, and deadline in a sentence. Although the type-dependency extracts features somewhat accurately, it extracts features incorrectly in many cases. Therefore, we emply a machine learning-based approach to identify the structure of a task more accurately.

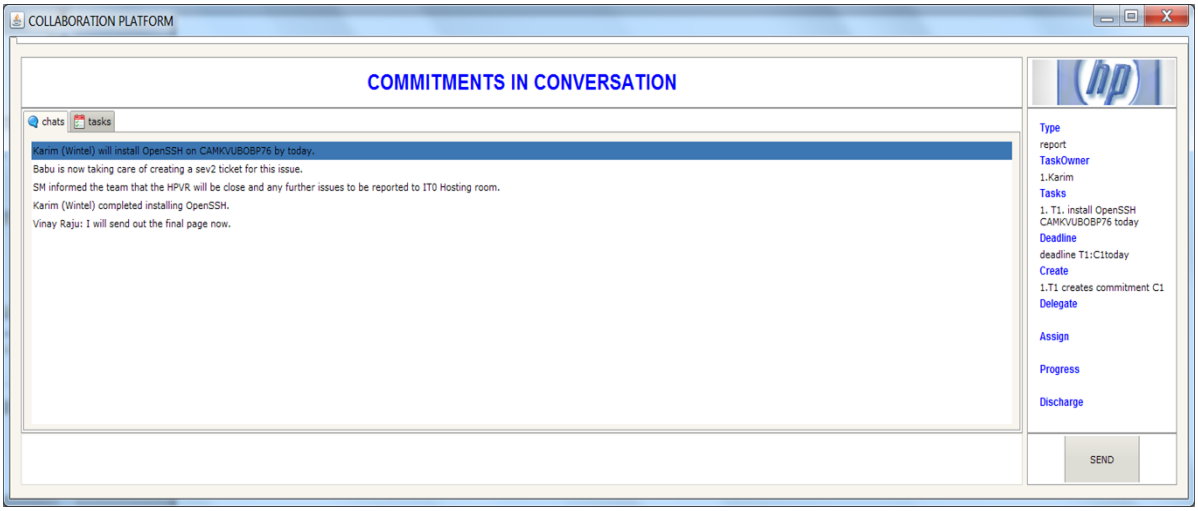

Figure 10: A screenshot of our prototype tool for commitment identification and monitoring.

8. CONCLUSIONS AND FUTURE WORK

With the rise in the informal collaboration and web-based communication tools, the problem of organizing work in the collaborational setting and in the context of people-centric processes becomes more vital for the organizations. To the best of our knowledge, this is is the first work that presents an approach for the automated identification and monitoring of commitment lifecycle from the text of conversations amog people. We introduced an approach for extracting tasks and commitments from sentences in email and chat conversations. Our NLP-base algorithms leverages typed dependency for identifying tasks and its related parameters. Our machine learning approach accurately identifies whether a task indicates a commitment and further identify whether a task progresses a commitment or terminates it. Using our machine learning approach we get high precision for commissive and directive creation on both emails and chats. Our approach also provides promising results for the delegate, discharge and cancel classes. Finally, we provide a tool that implements the approach, and can be used for assisting workers in capturing commitments in collaboration tools.

We have several future directions. First, we plan to improve results for delegation, discharge, and cancellation, specifically in chat conversations. Second, we plan to explore and compare other approaches for identifying tasks from sentences. Third, we plan to consider applying unsupervised machine learning such as Hidden Markov Models (HMM) to this problem as the supervised machine learning approach relies on manual labeling of data, which is tedious to provide. Fourth, another future step is to reconstruct conversations from both emails and chats and then apply our approach.

9. ACKNOWLEDGMENTS

This research was initiated by HP Labs, Palo Alto, US 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.

10. REFERENCES

[1] M. De Marneffe, B. MacCartney, and C. Manning. Generating typed dependency parses from phrase structure parses. *Proceedings of LREC*, 6:449–454, 2006.

[2] A. Fiore and J. Heer. UC Berkeley Enron email analysis. 2004.

[3] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226. Springer Berlin / Heidelberg, 2004.

[4] 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 2010 14th IEEE International Enterprise Distributed Object Computing Conference*, EDOC '10, pages 247–256, Washington, DC, USA, 2010. IEEE Computer Society.

[5] M. Purver, P. Ehlen, and J. Niekrasz. Detecting action items in multi-party meetings: Annotation and initial experiments. In *Proceedings of the Third International Conference on Machine Learning for Multimodal Interaction*, pages 200–211. Springer-Verlag, 2006.

[6] 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*, pages 748–758, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[7] 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)*, 2010.

[8] J. R. Searle. *Speech Acts*. Cambridge University Press, Cambridge, UK, 1969.

[9] J. R. Searle. A taxonomy of illocutionary acts. In K. Gunderson, editor, *Language, Mind and Knowledge*. University of Minnesota Press, Minneapolis, 1975.

[10] M. P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7(1):97–113, Mar. 1999.

[11] P. R. Telang and M. P. Singh. Enhancing Tropos with commitments. In A. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. K. Yu, editors, *Conceptual Modeling: Foundations and Applications*, volume 5600 of *LNCS*, pages 417–435. Springer, 2009.

[12] P. R. Telang and M. P. Singh. Specifying and verifying cross-organizational business models: An agent-oriented

approach. *IEEE Transactions on Services Computing*, 5(3):305–318, July 2012.

[13] T. Winograd. A language/action perspective on the design of cooperative work. In *Proceedings of the 1986 ACM Conference on Computer-supported Cooperative Work*, CSCW '86, pages 203–220, New York, NY, USA, 1986. ACM.