Abstract
In recent years, online chat has become a dominant mode of communication. This text-based medium has the potential of improving information awareness within an organization, but only if the critical information within messages can be identified and directed to where it is most needed. Such a goal has many challenges that traditional Information Extraction (IE) approaches have rarely addressed: the text is “dirty” (containing typos, misspellings, sparse punctuation, etc.), messages are fragmented and refer implicitly to previous messages and shared knowledge, messages from multiple topics are interleaved, etc.

Past work in conversation analysis has included in-depth discussions of dialog acts, i.e., the individual utterances that comprise conversations. This paper describes how dialog acts within online chat differ from those within two-person voice conversations. It then presents methods for identifying dialog acts and the role that dialog acts play in identifying individual conversations within a chat stream.

Identifying conversations is a necessary step for extracting actionable information, such as identifying individuals with specific expertise, recognizing reports of offline activities, and alerting decision makers to critical developments.

Finally, we describe Chat-IE, a prototype software system that performs live dialog identification on chat streams.

Introduction
In recent years, online chat has become a critical communication tool for businesses and the military. In some cases, it has replaced radio communications and teleconferencing, since it allows almost instantaneous communication without the challenges that voice communications sometimes face, such as extraneous noise and required synchronicity. It also reduces the potential for misunderstood utterances and easily supports complete, searchable communication logs.

However, identifying how this new avalanche of information can be used to fulfill information needs is no small task. Features of online chat that complicate automated retrieval and extraction of relevant information include:

**Typos and misspellings** – Users of chat are less likely to proofread their messages than writers of news articles.

**Fragment sentences and little punctuation** – Complete sentences are rare, and punctuation is even rarer.

**Conversational context** – Individual messages often make little sense unless taken in the context of a conversation. Consider this scenario that might occur in the chatroom of a web development team:

Speaker1: which server software are you using
Speaker2: Tomcat
Speaker3: Did the unit tests pass last night
Speaker1: what version
Speaker1: yup
Speaker2: 5.5

There are two distinct conversations, expressing two separate ideas: 1) Speaker1 is using Tomcat 5.5; and 2) the unit tests ran successfully the previous night.

**Abbreviations and acronyms** – Online chat communities develop lexicons of domain-specific abbreviations to reduce the amount of typing needed to express an idea.

**Emotional indicators** – Acronyms (e.g., LOL, IMO) and emoticons are used liberally to indicate an emotional tone that could be expressed using inflection in spoken conversation.

This paper describes an approach to dealing with these challenges when processing text from online chat. The foundation of this approach is dialog act identification (cf., Stolcke, et al., 2000). We discuss how dialog acts in online chatrooms differ from those of one-on-one voice conversations, as well as the unique problem of identifying interleaved conversations within a single chat stream. We then describe a loosely coupled collection of techniques that we applied to the problem of identifying dialog acts.
within chat logs collected in a software development environment.

**Dialog Acts in Online Chat**

The following dialog act types are common both in most forms of conversation, regardless of the medium, the domain, or the number of participants:

- **questions** – The majority of questions are relatively straightforward, composed primarily of yes-no questions and wh-word questions. In online chat, these types of questions can be easily recognized even when punctuation is not present. Questions are particularly useful, as they are good markers for organizing dialogs. That is, questions often indicate the beginning of a new dialog or sub-dialog, and they set up an expectation for answers to follow.

- **answers** – The “answers” category refers specifically to yes/no type answers. These are relatively easy to recognize and are critical in assessing the certainty of information extracted from a dialog. An interesting thing to consider is the relative likelihood of certain answers in online chat vs. speech. When a speaker wants to give a definitive “yes” answer, he is as likely to say “uh huh” as “yes”. However, in online chat, when “uh huh” is used to mean “yes”, it’s likely to be less definite than words like “yes”, “yup”, “yeah”, etc.

- **agreement/disagreement** – These are similar to answers, except that they are responses to statements rather than to questions. Again, these types of utterances impact the certainty of extract information.

- **discourse markers** – These include backchannels (e.g., “uh huh”), linking comments (e.g., “so”, “anyway”), verbal pauses (e.g., “umm”, “let’s see.”), and hedges (e.g., “maybe”). In spoken language, these are often recognized by intonation features that aren’t present in online chat. For example, verbal pauses are often drawn out in duration. However, in online chat, the “drawn out” intonation may be represented by an ellipsis, or by the user ending a message.

- **statements** – Statements are both the simplest and the hardest to tag. It is pretty simple to recognize a statement, and the majority of dialog acts will fall into this category. However, statements have a large variety of dimensions that need to be considered. In online chat, these dimensions may be indicated by emoticons and emotion words (e.g., “hrmph”, “grin”, etc.). And, although these dimensions are shown to be specific to statements, online chat may use emoticons and emotion words to qualify other categories. For example, if the question, “Did I hurt your feelings?” is answered, “yes”, that has a different meaning than the answer “yes ;-)

<table>
<thead>
<tr>
<th>Dialog Act</th>
<th>% of corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>statement-non-opinion</td>
<td>20.97%</td>
</tr>
<tr>
<td>statement-opinion</td>
<td>19.17%</td>
</tr>
<tr>
<td>action-description</td>
<td>8.82%</td>
</tr>
<tr>
<td>yes-no-question</td>
<td>8.73%</td>
</tr>
<tr>
<td>action-directive</td>
<td>6.30%</td>
</tr>
<tr>
<td>commit</td>
<td>5.31%</td>
</tr>
<tr>
<td>agree-accept</td>
<td>3.87%</td>
</tr>
<tr>
<td>other</td>
<td>3.60%</td>
</tr>
<tr>
<td>wh-question</td>
<td>2.61%</td>
</tr>
<tr>
<td>thanking</td>
<td>2.43%</td>
</tr>
<tr>
<td>affirmative-answer</td>
<td>1.62%</td>
</tr>
<tr>
<td>completion</td>
<td>1.53%</td>
</tr>
<tr>
<td>declarative-yes-no-question</td>
<td>1.53%</td>
</tr>
<tr>
<td>hmm</td>
<td>1.44%</td>
</tr>
<tr>
<td>response-acknowledge</td>
<td>1.35%</td>
</tr>
<tr>
<td>apology</td>
<td>1.26%</td>
</tr>
<tr>
<td>appreciation</td>
<td>1.26%</td>
</tr>
<tr>
<td>negative-answer</td>
<td>1.17%</td>
</tr>
<tr>
<td>offer</td>
<td>1.17%</td>
</tr>
<tr>
<td>correction</td>
<td>0.90%</td>
</tr>
<tr>
<td>hedge</td>
<td>0.72%</td>
</tr>
<tr>
<td>maybe-accept-part</td>
<td>0.63%</td>
</tr>
<tr>
<td>open-question</td>
<td>0.63%</td>
</tr>
<tr>
<td>reject</td>
<td>0.63%</td>
</tr>
<tr>
<td>hold-before-answer-agreement</td>
<td>0.45%</td>
</tr>
<tr>
<td>other-answer</td>
<td>0.45%</td>
</tr>
<tr>
<td>summarize-restate</td>
<td>0.45%</td>
</tr>
<tr>
<td>rhetorical-question</td>
<td>0.27%</td>
</tr>
<tr>
<td>conventional-closing</td>
<td>0.18%</td>
</tr>
<tr>
<td>quotation</td>
<td>0.18%</td>
</tr>
<tr>
<td>downplayer</td>
<td>0.09%</td>
</tr>
<tr>
<td>option</td>
<td>0.09%</td>
</tr>
<tr>
<td>or-clause</td>
<td>0.09%</td>
</tr>
<tr>
<td>self-talk</td>
<td>0.09%</td>
</tr>
<tr>
<td>abandoned</td>
<td>0.00%</td>
</tr>
<tr>
<td>acknowledge-backchannel (mm-hmm)</td>
<td>0.00%</td>
</tr>
<tr>
<td>attention</td>
<td>0.00%</td>
</tr>
<tr>
<td>backchannel-question</td>
<td>0.00%</td>
</tr>
<tr>
<td>conventional-opening</td>
<td>0.00%</td>
</tr>
<tr>
<td>declarative-wh-question</td>
<td>0.00%</td>
</tr>
<tr>
<td>repeat-phrase</td>
<td>0.00%</td>
</tr>
<tr>
<td>signal-non-understanding</td>
<td>0.00%</td>
</tr>
<tr>
<td>tag-question</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

- **control acts** – These include commands (“Come to my office.”), requests (“May I come over?”), offers (“Do you want a concert ticket?”), promises (“I’ll send you an e-mail.” “I won’t leave.”), etc. These are
somewhat troublesome, because the effects of many control acts are physical actions, not observable in the chat stream. However, recognizing such control acts may help set up expectations for further chat. For example, “Come to my office” may actually indicate the end of a dialog, since the conversation is expected to continue in person.

- **conversational conventions** – These include conversation openers and closers (e.g., “hello”, “bye”), exclamations, apologies, thanks, “you’re welcome” (also, “no problem”), congratulations, etc.

Within multi-party online chat, particularly in support of a larger activity, we identified four additional dialog act types. While these can be incorporated into the previously described types, marking them separately is valuable to the process of identifying and extracting from conversations.

- **action description** – Because the chat corpora we examined were from chat rooms being used in support of a larger activity, it was common for contributors to report ongoing and completed activities. For example, “I’m finishing up the new UI design.”

- **completion** – Stolcke, et al. (2000) identified a dialog act type called “collaborative completion”. However, in the online chat domain, self completion was far more common, so we generalized this dialog act type to include both.

- **correction** – Mistakes are common in online chat activity. Since a message cannot be edited once it is sent, new messages are frequently posted correcting previous mistakes. For example:
  
  Speaker1: I’m working on defect 567  
  Speaker1: I meant 568
  OR
  Speaker1: I’m working on defect 567  
  Speaker1: 567=568

- **attention** – In chatrooms with more than two people, participants often begin a message with the name (or call sign) of the user to whom they are speaking.

### Online Chat Conversations

The critical aspect of dialog processing within a chat stream is being able to recognize which messages follow from which previous messages. Within formal discourse, this process would involve analyzing one or more sentences, summarizing the topic and intent of the utterance, and determining what role the utterance plays in the dialog. Within a chat stream, the same process must be followed, with two complications. First, a given message may not follow directly from the message occurring immediately prior. It may be the beginning of a new dialog or it may follow from a message occurring further back in the chat history (or both). Second, a given message may be sparse (as little as a single word) and is likely to be ungrammatical, making traditional sentence analysis virtually unusable.

Constructing dialogs requires the system to be able to recognize the following:

- Is there an explicit connection between the message and the current context? This connection may be determined through keyword recognition.
- Is the message a question/request? (Note that punctuation, such as question marks, may not be available.)
- Is the message really a continuation of a previous message? i.e., did the speaker divide a single “utterance” into two or more separate messages? Or perhaps it is a self-correction.
- Could this be a response to a previous message? This requires that an ongoing dialog be present and that the current message is topically and structurally compatible with the ongoing dialog. (Note that dialog structural clues will be good positive indicators, but their absence is not a good negative indicator for dialog membership.)

Many dialogs will contain just a small number of messages, so it is important to recognize when a new dialog is starting, and develop a reasonable means to phase out recognized dialogs that are unlikely to be continued. A given dialog may be interrupted by unrelated messages, or may contain fragmented/elliptical messages. Recognizing such complexities requires knowledge of the concepts being discussed as well as an understanding of how communication acts [Grice, 1975] [Cohen and Levesque, 1990] structure dialogs. Some issues that we considered during our research:

- **Dialogs between more than two people.** Almost all research in the psychology of dialog and the recognition of dialogs assumes exactly two participants.
- **Messages that belong in multiple dialogs.** For example, a request with multiple responses is a dialog. However, if one of those responses spawns an additional discussion, that discussion is also a dialog.
- **Messages that are too ambiguous to determine definitive dialog membership.** For example, if two statements are followed by a third message consisting simply of, “OK”, it is unclear which statement is being acknowledged.

### Dialog Act Identification

During our research, we developed three complementary approaches to dialog act identification that we elaborate here.
Long String Matching

InfoTracker (Creswick, Fujioka, and Goan, 2008) is a technology developed by Stottler Henke to identify phrases that are important to a particular set of documents. In previous work, we have used this technology to identify information leaks and to assist in information redaction. We wondered if the same technology would work to classify dialog act types.

For this experiment, an analyst tagged 1111 chat messages collected from a software development group’s IRC chatroom for one specific project. We then divided this into a training set (90%) and a testing set (10%). Our set of dialog act types is large relative to this sample size (43 types). In addition, the vast majority of dialog acts are covered by only a handful of dialog act types: statement-non-opinion, statement-opinion, action-description, yes-no-question, action-directive, commit, and agree-accept (see Table 1). Given this limitation, we adjusted our tests in a couple of different ways. First, if a dialog act type had fewer than 10 messages, then 1 message was chosen as a test message and the rest were used for training. Second, we reran the test five times, randomly creating the training and testing sets each time.

The most common dialog act types were identified correctly about 60% of the time. Not surprisingly, the system’s performance on the least common dialog act types were significantly lower.

However, upon examination of the incorrectly classified messages, we found that a large proportion of them have patterns that are conducive to recognition using a rule-based pattern matching.

Rule-Based Pattern Matching

Although chat messages are notoriously ungrammatical, there are some language conventions that even chat users are willing to follow. We developed a set of simple hand-coded rules that describe many of the patterns we observed during the Long String Matching experiments. Each rule was defined with up to three primary features: 1) “starts with”, 2) “contains”, and 3) “ends with”. In addition, “starts with” and “contains” rules can have the additional feature of “followed by” or “not followed by”. The value of a feature could be a string literal or a list of parts of speech. Each rule was assigned a strength that could be used for disambiguation.

For example, a message starting with the string “I’m” (case-insensitive) following by an “-ing” verb is likely to be an “action description” dialog act. Some of the rules we defined for our domain include:

- **Wh-questions** – Messages starting with wh-words (what, which, why, etc.).
- **Yes-no-questions** – Messages starting with “does” or “will” and ending with “?”.
- **Describe-other** – Messages starting with a proper noun followed by the string “does”.
- **Statement-opinion** – Messages containing one of: “might”, “maybe”, “should”, “seems”, “i think”, “looks like”, “look like”, “probably”, or “i’m sure”.
- **Action-directive** – Messages starting with infinitive verbs.
- **Action-description** – Messages starting with “i”, “i just”, “i have”, “i’m”, etc., followed by a past tense verb or “-ing” verb.
- **Commit** – Messages starting with “i will”, “i’ll”, “i’m going to”, or “i am going to”. Also, messages starting with “will” followed by an infinitive verb.

These rules were simple to define and quick to execute, and the rule matching component was dependent only on tokenization and part of speech tagging. They shored up the accuracy of the dialog act identification to about 90% for this corpus. This level of performance is made possible, in part, by the constrained context of these messages: the users had developed a strongly shared vocabulary, as well as common patterns of interaction, that promoted more regular language usage than might be seen in other environments (e.g., generic chatrooms, support center conversations).

A natural extension to our rule syntax would be to allow “variable strings”, such as times, dates, and locations, as well as domain-specific concepts (e.g., defect numbers, class names), to increase the expressability of the patterns.

Fragment Identification

A special class of dialog act that appears frequently in online chat are “fragments”. We define a fragment message as one that has no verb. Such fragments most often are responses to questions. For example, a question like, “When will that task be done?” would have a response like “tomorrow” or “in 2.5 hours”.

Using Dialog Acts to Identify Dialogs

Although the research described here focused on the identification of dialog acts, we developed a two-stage system for extracting discrete dialogs from a message stream.

The first stage identifies when a dialog act has been split across messages. This may occur when the user accidentally types <Enter>, or when the user posts a self-correction. Although corrections are legitimately a separate dialog act, combining the original utterance with the correction simplified the second stage of dialog identification.
For the second stage, we hand-coded dialog patterns that were suitable for this corpus. Matching these patterns require that dialog acts occur in the order described, but do not need to be sequential. A single dialog act may be matched by multiple patterns. Such overlap may occur due to actual dialog overlap (e.g., a new dialog was launched because of a message that belongs to an ongoing dialog), due to message ambiguity, or due to the extraction needs of the user.

Patterns we defined for our demonstration included:

- **Status updates** – An action-directive or wh-question, followed by any number of action-descriptions.
- **Directed request with acknowledge** – An attention followed by any number of utterances, followed by a response-acknowledge by the person mentioned in the first utterance.
- **Confirmed expertise (1)** – An action-description followed by a thanking or a response-acknowledge (preferably mentioning the initial speaker). (First speaker demonstrated expertise.)
- **Confirmed expertise (2)** – A yes-no-question or wh-question followed by a describe-other. (Second speaker demonstrated expertise.)

### The Chat-IE Prototype

Chat-IE is a prototype software system that analyzes online chat content, either from log archives or by monitoring a live IRC chat room, in order to identify individual conversations. All processing done by Chat-IE is event-driven. Each analysis algorithm is self-contained as an expert in the system and executes when an event fires indicating that information that expert consumes has become available. This allows many experts to execute simultaneously or to not execute at all.

Figure 1 shows the experts we developed for this prototype. Processing in this system proceeds as follows:

- When a new message becomes available, the Message Preprocessor tags the message with the timestamp, the message type, the user (speaker), and the user role (if applicable). It then fires a “new message” event.
- The message is tokenized using Antlr (http://www.antlr.org/). This tokenizer is specialized for domain-specific tokens, as described earlier. A “new tokens” event is fired.
- Several experts respond to the “new tokens” event in no particular order and fire events when complete:
  - Recognized acronyms are expanded.
  - Domain-specific terms and phrases are matched.
  - Using a slightly modified version of the Hepple part-of-speech tagger [Hepple, 2000], each non-whitespace token is tagged with one or more parts of speech.
- Participant names are tagged. This expert may start with a list of team members who will participate in the chat. Additional participants are saved from the user tags stored by the Message Preprocessor.
- Two experts respond to the “Part of Speech Tags” event and fire the “new dialog act” event:
  - The message extension expert identifies dialog acts that cross message boundaries, even if there are messages from other users in between, and combines them into a single annotation. We are unaware of any other IE application that allows such non-contiguous annotation.
  - The dialog act splitter identifies when a single message contains multiple dialog acts and splits them accordingly.
- The dialog act identification experts described earlier all respond to the “new dialog act” event and fire “new dialog act type” events.

![Figure 1: The experts currently implemented in the Chat-IE prototype.](image)

- **Action Act** conflict resolver handles cases where the dialog act identification experts have produced more than one possible tag.
- The dialog model matcher attempts to add the new dialog act(s) into a previously started dialog object, or starts a new dialog object. This expert is also responsible for retiring stale dialog objects. It fires an “dialog changed” event for each dialog object that has changed.

When in use, the User Interface also responds to all fired events, displaying all identified tags. This interface was adapted from the GATE open source project [Cunningham, et al., 2002] and allows the user to view any set of tag types. As with GATE, each type is assigned a different highlight color, and the colors combine when tags overlap. Further details are displayed in the annotations box. Chat-
IE also allows the results of many annotations to be edited by the user.

Note that the experts shown in Figure 1 are not a static set; experts may be added, replaced, or modified without retooling the other components. (Of course, consideration must be given to the reliance that many components have on the results of previous components. For example, the part of speech tagger only operates once tokens have been identified.) In addition, as with most text processing systems, the underlying knowledge used by the experts will change from domain to domain (e.g., different terminology, different language patterns, etc.)

Further, different domains or applications may have additional experts. For example, chatrooms that have “bot” participants may need an expert that combines multiple, sequential “status dump” messages into a single dialog act. Applications that require high-fidelity geotagging would include one or more “location experts” that use gazetteers to identify location names within the chat text.

Lessons Learned

- **Users have very specific needs for chat analysis.** When speaking with potential users, each had a very specific application they wanted to enable. For example, one user wanted to automatically classify chat dialogs and e-mail messages/threads into topics or “bins” so he could concentrate only on text that was relevant to his current task. Another user wanted an application that could monitor chat rooms for triggering events, which would initiate a live feed of related messages to the user’s screen. This implies that user-friendly configuration and deployment are almost as important as the analysis technology.

- **Everything hinges on the tokenizer.** In chat applications, even simple categories of tokens, such as punctuation, become complicated, as users combine characters in novel ways (e.g., ??!!, <---->, etc.) In addition, special domains tend to have special tokens (e.g., “/usr/bin/chatLogs”, “65.4N”) that a generic tokenizer will miss. Since most analysis and extraction techniques make assumptions about the tokens, it is critical to process them accurately.

- **Partial dialogs may need to be retired without being “finished”)**. When analyzing large numbers of messages (on the order of tens of thousands), the number of partial dialogs that a message could belong to can cause serious performance problems. Heuristics for retiring a partial dialog need to consider features such as time elapsed, number of messages posted, and participation of speakers in other dialogs. In addition, access to external context, particularly in quickly changing environments, can indicate when a dialog topic has become stale.

Conclusions

The project described in this paper has several results that improve on the state-of-the-art and facilitate accelerated future research. The first of these is an event-driven framework for creating new expert-based IE applications. This framework has been employed both for Chat-IE and for an application for producing error-free geospatially tagged text.

A second significant result of this work is the creation of dialog act type identification algorithms that work on dialogs that have more than two participants, and are specialized for online chat messages. These two advancements will be critical moving forward, as activity within online communities continues to increase and the need to analyze this activity grows.

Finally, the focus of this work on processing messages incrementally, rather than waiting for some level of completeness, is vital for extracting critical information in the short timeframes that are customary in operational settings. Although more work is needed before Chat-IE will be ready for such conditions, the work described within this paper provides a solid foundation for moving toward that goal.

References


