



Multiparticipant chat analysis: A survey



David C. Uthus^{a,*}, David W. Aha^b

^a NRC/NRL Postdoctoral Fellow, Washington, DC 20375, United States

^b Navy Center for Applied Research in Artificial Intelligence, Naval Research Laboratory (Code 5514); Washington, DC 20375, United States

ARTICLE INFO

Article history:

Received 13 May 2011

Received in revised form 22 October 2012

Accepted 24 February 2013

Available online 26 February 2013

Keywords:

Multiparticipant chat

Artificial intelligence

Natural language processing

Machine learning

Text analysis

ABSTRACT

We survey research on the analysis of multiparticipant chat. Multiple research and applied communities (e.g., AI, educational, law enforcement, military) have interest in this topic. After introducing some context, we describe relevant problems and how these have been addressed using AI techniques. We also identify recent research trends and unresolved issues that could benefit from more attention.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Multiparticipant chat [48] is a form of *chat* with multiple participants conversing synchronously through textual communication. It is a form of *microtext* (e.g., microblogs, Short Message Service (SMS), transcribed voice communications) [33], which in turn is a subset of computer-mediated textual conversations such as email, discussion forums, and social network sites [45,48].

Multiparticipant chat has a variety of uses: recreation, business [101], distant software development collaboration [18, 43,83], online courses [47], collaborative learning [7,90], gaming [14,50], technical support [107], military command and control [38], and even recreation of Shakespearean plays [24]. Given its widespread use, there are many motivations for analyzing and manipulating it. For example, teachers of online courses may want to summarize what was discussed in chat for others to later review, or military command and control specialists may want to find important, urgent messages that should be brought to a decision maker's attention. Unfortunately, due to its informal, conversational nature multiparticipant chat is difficult to analyze when compared to other textual communications such as email and news articles [98].

We begin this survey with definitions of chat terminology and the current research foci (Section 2), then describe motivations for the needs of multiparticipant chat analysis (Section 3). This is followed by examining the current state of research in this area (Section 4), and we finish with discussions of what is needed for this field to grow (Section 5). Our objective is to foster more recognition of this body of work, which has been reported in a widely dispersed set of publications. Researchers in AI and related fields have addressed many problems in multiparticipant chat that are pertinent to practitioner communities, which may benefit from this progress and identify additional problems that could drive new areas of AI research.

* Corresponding author.

E-mail addresses: david.uthus.ctr@nrl.navy.mil (D.C. Uthus), david.aha@nrl.navy.mil (D.W. Aha).

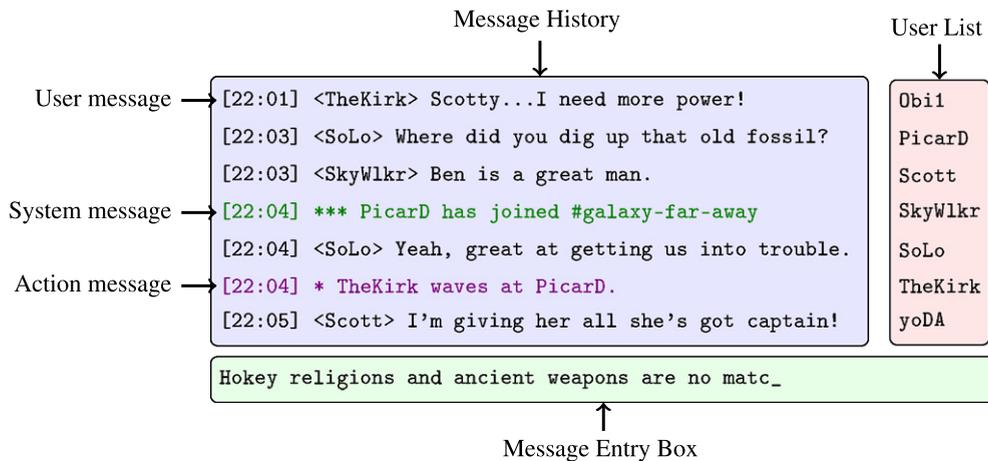


Fig. 1. Abstracted chat client user interface, including examples of user, system, and action messages.

2. Multiparticipant chat analysis: Basic definitions and tasks

Chat is a form of synchronous textual communication between a community of *users*. These users converse in *chat rooms* (sometimes called *channels* [80,101]), which are virtual locations for chatting on the Internet and private networks. These can be supported using, for example, tools for instant messaging (e.g., MSN Messenger, AIM), Internet Relay Chat (IRC), virtual game lounges (e.g., Battle.net, Steam), game environments (e.g., MUDs, MMORPGs), and collaborative learning environments. It is also considered an old medium as MUDs have been around since the late 1970s and IRC was created in 1988 [49,80].

Most chat room user interfaces follow a similar structure, with an abstract user interface shown in Fig. 1. As shown, they include a window that lists the participants, a window that displays the history of typed messages, and a window for typing in a new message. Within the message history's window, messages are displayed in the chronological order that they were received by the system, with new messages appearing at the bottom of the screen.

Each user is identified by a *nickname* [80,101]. Depending on the service, these nicknames can either be registered, so that the nickname is unique to an individual, or unregistered, so that the nickname is available on a first-come, first-serve basis to whomever is online at the time. Users may also be able to change their nickname while online.

Users interact with each other through messages. *User messages* generally consist of three parts: the nickname identifying the message's author, the *timestamp* of their communication, and their authored *utterance*. An utterance can be of any length up to the limit of the supporting application, with some of these limits being small (e.g., IRC only allows for utterances up to 510 bytes [55]). In addition to messages between users, there can also be *system messages*, which are system-generated (e.g., announcing a user entering or leaving the chat room), and *action messages* (also known as *emotes*), which are third-person descriptions of a chat user performing some action (e.g., "Sam is smiling at you"). An example of these types of messages can be seen in Fig. 1.

There are two modes of chat: *dyadic* and *multiparticipant* [48]. Dyadic chat involves two people communicating directly with each other without other participants in the same chat room. Multiparticipant chat involves three or more people communicating within the same chat room, which is this survey's focus. Multiparticipant chat involves challenges that do not exist in dyadic chat. For example, multiparticipant chat may involve multiple, asynchronous, and simultaneous conversations, with messages interwoven among the various conversations [46].

A *conversation thread* is a sequence of messages among users conversing with one another, with messages being relevant to one another within the conversation [8,49]. New conversations are started in two ways: either with a conversation-initiating statement or through a *schism* [35]. A schism occurs when a conversation splits into two conversations – the new conversation is formed due to certain participants branching off from a specific message, refocusing their attention upon each other. Examples of these can be shown in Fig. 2. Four conversations take place in this chat log. Threads 1, 2, and 3 are each started by a conversation-initiating statement. Thread 4 is the result of a schism, with user *Supertemp* asking user *Aquagirl* what is "Lubuntu" due to her earlier statement, resulting in a second conversation about Lubuntu. We will come back to this example later when discussing the difficulties of overcoming entangled conversation threads.

A chat room can have a *theme* or *topic* that describes the types of discussion intended for the given room, which may sometimes be shown explicitly. This can be of importance when determining if a conversation or message is appropriate or on topic for the chat room. Conversation threads and individual messages can also have their own topics, though these are generally not explicitly stated like the topic of a chat room.

Thread	Message
1	[04:19] <mmc_> i have a problem trying to install precise...
1	[04:22] <mmc_> when i boot the installer i get a garbled screen.
2	[04:22] <AquaGirllLove> I have a problem with ELinks not displaying in Lubuntu 12.04 in the lxterminal when I click on the icon in the menu list.
1	[04:22] <dr_willis> !nomodeset mmc_
1	[04:22] <ubottu> mmc_: A common kernel (boot)parameter is nomodeset, which is needed for some graphic cards that otherwise boot into a black screen or show corrupted splash screen. See http://ubuntuforums.org/showthread.php?t=1613132 on how to use this parameter
3	[04:22] <MacDaddy> how do i install ff5?
2	[04:23] <DarwinSurvivor> AquaGirllLove: the lxterminal icon or the elinks icon?
1	[04:23] <mmc_> dr_willis: not sure how to pass a boot parameter to a screen i can't read...
1	[04:23] <dr_willis> mmc_: that url shows you how... .
1	[04:24] <dr_willis> unles you dont even see the boot/syslinux menu... then thats a deeper issue
2	[04:24] <AquaGirllLove> The ELinks icon.
4	[04:25] <Supertemp> AquaGirllLove, noob question, but what is Lubuntu?
2	[04:25] <dr_willis> tried running elinks frim the shell?
1	[04:25] <mmc_> i looked at it... it says something about a purple screen with a logo at the bottom and setting my language... i can't see those screens
3	[04:26] <ActionP> Firefox 5 will be available from the repositories soon.
4	[04:26] <AquaGirllLove> supetemp: Lubuntu is the ubuntu with LXDE
2	[04:26] <AquaGirllLove> The lxterminal opens, but all I see is a cursor.
2	[04:26] <DarwinSurvivor> AquaGirllLove: open lxterminal and type in "elinks" (without quotes) and hit enter
4	[04:27] <Supertemp> What's god about LXDE?
4	[04:27] <ActionP> Superb, LXDE is a lightweight DE, good for older machines
4	[04:28] <Superb> ActionP, ?
4	[04:28] <ActionP> Sorry, that was meant for Supertemp

Fig. 2. An example chat log illustrating the difficulties of identifying conversation threads and problems they cause.

Table 1

Comparison of multiparticipant chat to related media. We exclude media which have less similarity (e.g., formal documents) to reduce table size.

	Metadata	Abbreviations & Emoticons	Restricted Length	Synchronous	Entangled Threads
Email	✓	✓			
Newsgroups	✓	✓			
Message Boards		✓			✓
Transcribed Conversations				✓	✓
SMS	✓	✓	✓	✓	
Microblogs	✓	✓	✓		
IM (Dyadic Chat)		✓	✓	✓	
Multiparticipant Chat		✓	✓	✓	✓

2.1. Comparison to other media

Chat messages possess characteristics that differ from some other forms of text communication. Werry [101] described these characteristics as the frequent use of abbreviations, acronyms, clipped words, and deletion of subject pronouns. He further remarked on the use of emoticons (e.g., smiley faces), abbreviation of nicknames, stripping of vowels to reduce the number of keystrokes, and the creative use of capitalization, spelling, and punctuation to recreate voice, gesture, and tone. Danet et al. [25] commented that words used as actions or sounds are often emphasized with asterisks (e.g., “*waves*” and “*boom*”). Wong and Xia [102] noticed the use of words composed of both letters and numbers that phonetically replace longer words (e.g., “l8r” for “later”) along with the dynamic introduction of new chat terms. Dela Rosa and Ellen [26] observed that chat grammar is comparatively informal and unstructured in comparison to traditional text, with users engaging in a conversational tone.

Chat's characteristics also differentiate it from spoken discussions. Reid [80] noted that it is not possible to transmit through chat the non-verbal attributes of speech required for synchronous communications. Smith et al. [85] commented that chat, compared to spoken interaction from a conversational analysis perspective, was poor at: managing interruptions and turn taking; conveying comprehension between users; and resolving conflicts of floor control. O'Neill and Martin [73] commented that in text chat, users would begin new topics at will, which differed with face-to-face group discussions. Herring [46] argued that the interactional incoherent nature of chat is due to the lack of simultaneous feedback (i.e., a user cannot know what another user is typing until the other has sent their message). Danet et al. [25] discussed how chat can be viewed as a play, with the nicknames being a person's mask that enables them to converse through a virtual persona, unlike how they would in the real world.

Chat's characteristics overlap with and differ from other forms of microtext and conversational media such as email, newsgroups, and message boards. Table 1 summarizes these comparisons. Unlike multiparticipant chat, email, newsgroups, and message boards are usually not restricted in message length and are asynchronous. Email and newsgroups additionally

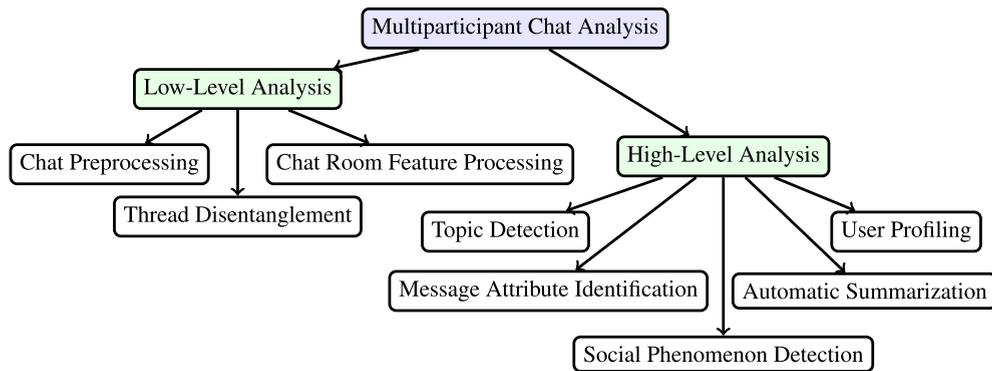


Fig. 3. Topology of multiparticipant chat analysis research.

have metadata available to them that prevent thread entanglement. Multiparticipant chat is most similar to microblogs, SMS, and instant messaging. One key different between chat (dyadic or multiparticipant) and microblogs is that microblogs and SMS can use metadata. For example, Twitter has special word functions, such as hash tags for topics and the “@” character to represent a twitter name. SMS metadata identifies the recipients of the message. Another difference between multiparticipant chat and these other three similar mediums is that chat suffers from entangled conversation threads, which makes analysis of messages more difficult.

2.2. Chat corpora

Chat corpora, composed of *chat logs* (a recording of messages within a chat room), are used for chat research. There are three types of chat corpora: non-annotated, annotated, and synthetic. The first, *non-annotated*, can be obtained in two ways. One way is by obtaining previously logged chat corpora, for example GNUe’s IRC archive [107]. The other way is by entering a chat room and logging the messages. A problem with using self-collected data is that it makes it difficult to compare different approaches as many self-collected corpora are not made publicly available. Unfortunately, a large majority of previous research has used self-collected, non-public data.

The second category of chat corpora is *annotated*. At this time, we are familiar with only three public sources which have been used for AI-related research. One is the NPS Chat Corpus collected by Lin [63], which was later annotated with part-of-speech tagging by Forsyth and Martell [39]. These chat logs involve general chat in rooms specified by age groups. The other two (unnamed) corpora are both annotated for conversation threads and are composed of chat rooms involving technical discussions. One of the two consists of three sets, namely #IPHONE, #PHYSICS, and #PYTHON, which were obtained by Adams [2]. The other corpus is the #LINUX corpus, which was obtained by Elsner and Charniak [35].

The third type of chat corpora is *synthetic* (i.e., they were composed using synthetic data instead of real conversations). Early examples are corpora created by Acar et al. [1] and Çamtepe et al. [21]. Both groups modeled a chat room using statistical analysis of real chat logs. Another example was described by Dela Rosa and Ellen [26]. They created a chat corpus that includes messages with synthetic status updates from US Navy ships.

As previously mentioned, most prior research studies have used non-annotated corpora. We believe this is due to the ease of obtaining such corpora and the difficulty of obtaining annotated or creating synthetic corpora. The problems associated with this will be further discussed in Section 5.1.

2.3. Artificial intelligence research foci

We now describe the many AI research foci on analyzing multiparticipant chat. AI techniques have been extensively used to analyze chat, including methods relating to natural language processing, classification, clustering, and machine learning. In general, AI researchers are interested in understanding and manipulating the contents of chat messages. In this section, we will define only what the foci are that are of current research interest. Section 3 will describe motivations for these research tasks and Section 4 will describe the progress that has been made on each of them. Future possible tasks that have not been a focus of AI research will be discussed in Section 5.4.

A topology of these foci can be shown in Fig. 3. We consider low-level analysis to be tasks which are needed for high-level analysis tasks. While many of these research tasks (e.g., topic detection, automatic summarization) are also tasks found in other text-processing domains, there are new challenges found here that make it difficult to apply traditional techniques to multiparticipant chat in contrast to more traditional forms of written communication [3,13,26,82,91,98,99,107].

We define *chat preprocessing* as the task of preparing chat messages for analysis. Chat preprocessing techniques include hypernym augmentation (generalizing words into more generic terms) [3], lemmatizing (mapping inflected forms of word to a common root) [26], stemming (removing the ending of words to find their common form) [9], stop word removal [90], and phonetic mapping [102].

We define *chat room feature processing* as the task of leveraging chat room characteristics to find relationships among messages. This includes features such as time stamps [3,35,36,70,100], nickname augmentation [3,35], name mentions [35,36,70,90,99], and speaker identity [36].

Thread disentanglement [35] is the task of detecting and extracting individual conversation threads from interwoven conversations in chat. Using Fig. 2 as an example, the goal of thread disentanglement is to group the twenty-two messages into their respective four conversations.

The text mining task, topic detection and tracking, was defined as the task of detecting and tracking new events within a stream of news stories [5]. In this context, *topic detection* has been changed to accommodate the nature of chat: the task is to detect which topics are being actively discussed within a chat room or chat log. This can be viewed as determining which topics have been discussed over a period of time, or examining an entire chat log and detecting which topics were discussed within it.

Message attribute identification is the task of detecting attributes of a chat message. Examples are sentence type tagging (e.g., declarative sentence, conditional sentence) [82], dialogue act tagging (e.g., greeting, system message, Wh-question) [17,39,103], part-of-speech tagging [39], and identifying important messages [10,13,26].

User profiling is the task of determining the profile, or characteristics, of a chat user. Given a set of messages authored by a user, the goal is then determine certain characteristics under study which are hidden due to chat's anonymity. Examples of this includes detecting a users' gender or age [60,63,88], or a users' topic interests [9].

Social phenomenon detection is the task of detecting social phenomenon taking place between the users. Examples of this include detecting social networks [20,70], determining if a user is deceptive [93], or if a user exhibits social abilities such as topic control and leadership [87].

Automatic summarization is the task of creating a textual summary from a textual source [64,86]. In chat, this can be any number of messages, such as within a period of time or within a conversation thread [94,107].

3. Motivations for chat research

Having introduced chat concepts, we next discuss how it is used and how it has been studied. The purpose of this is to highlight the importance of chat analysis research, and to motivate its need. In the following, we summarize perspectives from users interacting with chat, users' behavior and characteristics, online learning, and military use.

3.1. Chat interaction

One area of research concerns how users interact with chat, especially through their clients. There are many challenges that a user must overcome when reading chat, such as its threaded conversations, its synchronous nature, turn taking, and topic change [46,73]. Some researchers have approached these problems by investigating alternative user interfaces. Examples of this include using graphical circles to represent a user and restricting conversations to groups of users who are in near proximity [96]; showing real-time updates of someone typing [97]; displaying chat horizontally on a timeline instead of vertically [97]; and hybridizing chat with Usenet-like postings [85]. Essentially, these approaches are trying to overcome the problem of entangled conversation threads by changing how the user interacts with the client.

Unfortunately, many of these ideas have limitations. For example, when hybridizing chat with Usenet-like postings, new messages appeared within their respective conversation thread instead of at the bottom of the screen, making it difficult for users to predict where new messages would appear [85]. In the interface used by Viégas and Donath [96], the messages scroll across the screen horizontally. Users felt that the messages moved off the screen too quickly when compared to a traditional vertical scroll. Not surprisingly, most chat clients still use a traditional interface. We expect that research on thread disentanglement may lead to intelligent clients that allow users to more easily follow conversation threads of interest.

3.2. User behavior and characteristics

Multiparticipant chat analysis has been the focus of substantial research in the social and behavioral sciences, where studies have examined chat's impact on people and the benefits gained from its use. This has included areas such as social networks [77], cultural impact (e.g., Thai [76], Japanese [57], Lithuanian and Croatian [106]), gender differences of chat users [56,57,76,106], language variation (e.g., regional speech in Germany [6]; code-switching in German-speaking regions of Switzerland [84] and Indian IRC channels [77]), and user preferences in strategic games [14].

A common theme of these research papers is that these studies often require a lot of manual tagging and processing of chat logs or are restricted by the limited ability of the programs to process chat logs. For example, in the gender studies, oftentimes the authors manually determine the gender of each user using clues based on the nicknames and the contents of their messages. Even the recent work by Holmer [51], which attempts to introduce automation, still requires a key task (i.e., the threading of the conversations) to be performed manually. Siebenhaar [84], when investigating code-switching (changing between languages within a conversation), used a program to automatically tag words in the data, but it was restricted to a small number of words due to the program's inability to disambiguate many of the words. This manual labor and weak automation limits the amount of chat logs that can be examined. New techniques, such as author profiling and

message attribute identification, would allow chat logs to be examined more efficiently, and in turn allow for more data to be examined.

3.3. Online learning

Another area of focus has concerned online learning, where chat is used for discussions among teachers and students, with mixed results [15,47,74,90]. While studies found that students maintained focus on what is being discussed [47], explaining concepts took longer compared to face-to-face conversations, it required more effort by the instructors, and the students' work quality was lower than the instructors' expectations [74]. Some programs have been developed to visualize different aspects of chat in online learning, but many of these tools require manually annotating chat logs in order for them to visualize the area under study [15,47].

Online learning has also been used for language learning, as chats' anonymity is an equalizer for users, which can encourage people to practice more [54,89]. The first of these studies required an artificial limit on the number of chat users grouped together in order to prevent thread entanglement, which research on thread disentanglement could alleviate. The second required manual coding of chat logs to investigate a users' ability to clear up misunderstandings with "repair moves" (i.e., methods of user interactions to clarify a statement's meaning that was due to the difficulties of non-native speakers), which motivates the need for message attribute identification.

3.4. Military

The military has a long-term interest in chat analysis. In this section, we focus on the United States' military. While other nations' militaries also use chat for communications, we are most familiar with the unclassified literature on military chat analysis from this nation. We report on chat use in the military since it offers a unique, well-documented history of how chat changed the modes of communication within a group of collaborators, along with the benefits and problems that have arisen since its introduction.

The use of chat communications has grown quickly across all branches of the US military in recent years [38]. With respect to the Navy, chat was first used in 1994 for technical and administrative support. Its use then exploded during Operation Enduring Freedom in Afghanistan, and it further revolutionized command and control operations during Operation Iraqi Freedom [30,38,44,53,67]. Duffy and Bilbrey [29] estimated that up to 60,000 Navy personnel use chat systems on a daily basis. The rise in popularity of chat for command and control communications can be attributed to its many benefits: its ease of use, its ability to support information sharing among large numbers of concurrent users, it decreases language barriers in coalition operations, it increases the effectiveness of scheduling and coordination among distributed teams, it can be saved and documented for subsequent analysis, and its reliability under high latency, low bandwidth situations [30, 31,38,44,61].

In addition to these benefits, chat has also caused some difficulties. First, it can distract users from their primary tasks [12,23,61]. For example, Cummings [23] reported that Tactical Tomahawk operators focused on the task of chatting instead of their primary task of missile control. Second, chat can contribute to information overload [61,67]. For example, Navy watchstanders, who monitor chat rooms in addition to their other duties of monitoring multiple displays and radio traffic, monitor up to 20 chat rooms simultaneously [67]. Some investigators have suggested creating intelligent tools and techniques to aid chat users, such as with advanced alerting systems [67], automated highlighting of critical messages which may be missed in high-tempo situations [19], and automated analysis of chat messages [13]. Finally, the high number of chat messages can complicate analysis. For example, after the completion of training exercises, instructors need to quickly extract information from chat logs to incorporate into reports called "after action reviews". A large amount of chat data is created during a training exercise, making it challenging to find relevant messages for the different trainees and their missions [27,78,79].

The military has substantial interest with respect to chat analysis. For example, analysis of chat logs would benefit from research on thread disentanglement and topic detection. In addition, the military domain suffers from a unique situation of information overload that is not widely seen in non-military domains, resulting in the need to solve new analysis tasks (e.g., message attribute identification and automatic summarization).

4. Current state of multiparticipant chat analysis

We next describe the current state of multiparticipant chat analysis, examining the research foci defined in Section 2.3. There are many difficulties when approaching these research tasks – traditional techniques often do not work in this medium. For example, Wang et al. [98] discussed how the text mining techniques for topic detection and tracking, which can be used on regular documents and broadcast news, are not applicable to chat due to its unique characteristics. Tuulos and Tirri [91] commented that traditional models for topic detection failed to leverage the social network aspect of chat since they focused only on the textual content. Wang and Oard [99] explained that conventional lexical-based clustering does not work well for text streams because such messages are too short and are incomplete. Thus, these challenges have spurred AI research to create new techniques for chat analysis.

4.1. Chat preprocessing

Chat preprocessing techniques overlap a lot with the techniques used for other forms of microtext – normalizing the unusual, informal textual features of the chat messages to make them more amenable for traditional analysis. Many of the techniques that have been applied are the same as seen in traditional natural language processing. One approach though, phonetic mapping, was created to address features rarely seen in traditional text.

Phonetic mapping can be applied to words and numbers to define the meaning of words with unusual spellings [102]. For example, the mixed word “l8r” would be phonetically mapped to “later.” This differs from traditional spell checking [22] in that phonetic mapping does not use n-gram analysis or examine possible mutations of a word through insertion and deletion of characters. Instead, phonetic mapping uses phonetic mappings of both the target word and dictionary word, and calculates the probability of one mapping to the other. Wong and Xia [102] applied this to Chinese text, where roman characters and some Chinese characters can be used phonetically to represent words found in a standard Chinese dictionary. Their models achieved high normalization accuracy and the authors discussed possible errors in the models resulting from either ambiguous chat terminology or chat words that were created through some process other than phonetic mapping.

4.2. Chat room feature processing

As mentioned earlier, chat room feature processing aims to leverage additional information about chat rooms. It is used to find relationships among messages, which is important for applications like thread disentanglement, where relationships help determine conversations, and topic detection, where the commonality of messages can help determine their general topic. For example, Elsner and Charniak [36] mentioned that for thread disentanglement, time stamps and speaker identities are better cues than the contents of the messages in determining whether messages belong within the same conversation thread.

Time stamps can be used to find relations among messages through penalization of the time gap distance. Messages that are far apart in time are less likely to be related to each other while messages that are close together are more likely to be related.

Nickname augmentation, which focuses on the messages’ authors, assumes multiple messages by the same author are more likely to be within the same conversation. Our example in Fig. 2 is both a positive and negative example of this. All of the messages by users `mmc_`, `DarwinSurvivor`, `MacDaddy`, and `Supertemp` are located each within a single conversation, while users `dr_willis`, `AquaGirlLove`, and `ActionP` have messages in multiple conversation threads.

Name mentions are commonly used at the start of chat messages; they specifically address another user’s nickname (i.e., an intended recipient). Name mentions can be used to find explicit links between messages. An example of this is shown in Fig. 2, where some of the messages begin with the user name of the intended recipient.

Speaker identity assumes that conversations are dominated by a few speakers who speak frequently. This is built on the assumption that a small subset of the users in the chat room are involved in a conversation, and that most messages are spoken by a few core speakers. Speaker identity can aid in thread disentanglement, where a few speakers speaking frequently are probably within the same conversation.

These four chat room features have been found to be useful; they help to create additional information for analysis beyond just the content of the messages. What is unknown though is when is it beneficial to use these features. For example, comparing chat from a social channel with chat from a technical support channel, which of these features would help with disentangling conversation threads, and which of these would hinder? Comparisons of these features on different types of chat logs would be of great benefit in helping researchers know which chat room features they should use for different situations.

4.3. Thread disentanglement

Thread disentanglement addresses a major problem of multiparticipant chat: it lacks the type of turn-taking that characterizes spoken communications, which can cause threads to become entangled so that individual threads are frequently interrupted by irrelevant messages [46]. In addition, further complications arise with initiating messages and responses: multiple responses may target a single initiating message and a single response may target multiple initiating messages [46]. O’Neill and Martin [73] additionally noted that participants begin new threads spontaneously, which does not always happen in spoken group discussion. The authors also found that chat users could hold coherent discussions despite the presence of simultaneous conversations.

Disentangling conversation threads allows for subsequent analysis techniques to focus on single conversations. This assists with popular AI tasks such as topic detection and automatic summarization. The approaches presented here also have different capabilities. Some can be used online, allowing for dynamic interfaces (e.g., [3,35,82]). These are approaches that generally work incrementally with chat messages, and as such can work with live messages or be used on previously-recorded chat logs. Other approaches are limited to only working with static chat logs (e.g., [27,36,37,78]).

Thread disentanglement is most commonly studied using clustering methods. Shen et al. [82], Wang and Oard [99], and Wang et al. [98] used single-pass clustering. Starting with the first message as the first cluster, their algorithms assign each subsequent message to an existing cluster if the similarity of the message to the cluster exceeds a threshold, or otherwise

assigns them to a new cluster. Shen et al. represented the messages using a vector space model with term frequency-inverse document frequency (TF-IDF). They used the similarity of the vectors for each message along with sentence types and personal pronouns to determine the probability of a message belonging to a conversation. Wang and Oard expanded thread messages using social and temporal contexts, and used the expanded messages as the basic unit for their clustering algorithm. Wang et al.'s clustering algorithm performs two steps when receiving a new message. The first step performs single-pass clustering, while the second cycles through all conversations and renews the associations between messages based on the new message. This allows messages to be re-clustered whenever the cluster containing the new message has a stronger relationship with a given message.

Elsner and Schudy [37] and Elsner and Charniak [35] both used correlation clustering for disentanglement. Correlation clustering finds a set of clusters that maximizes the number of agreements between pairs within a cluster and maximizes the number of disagreements among pairs in different clusters. Both approaches use the same maximum-entropy classifier to determine if two messages are related. Elsner and Schudy used greedy methods and local search for the NP-hard problem of finding a good solution for correlation clustering, and tested their method on topic clustering in newsgroups and chat disentanglement. When processing the data, their approach conducts multiple passes with random permutations. They reported that using a combined greedy method with post-processing by local search achieved the best performance for their study. Elsner and Charniak [35] used a voting schema for correlation clustering. They processed messages incrementally as they were received. These two publications applied their thread disentanglement techniques to the same thread disentanglement #LINUX corpus as did Wang and Oard [99]. Of these three approaches, Wang and Oard reported the highest F-score. This may be because their method expands the messages in order to exploit social and temporal contexts.

A recent approach by Elsner and Charniak [36] examined disentangling chat with coherence models. They tested their models on both recorded telephone conversations and the four chat corpora for thread disentanglement: #LINUX, #IPHONE, #PHYSICS, and #PYTHON. Their approach used tabu search to find a solution for this problem. They conducted two sets of experiments with each chat corpus: disentangling a single message (assuming the rest of the chat log has the correct structure) and disentangling an entire chat log. When disentangling a single message, their approach did better than previous approaches [35] applied to the same corpora. When disentangling an entire chat log, they reported that tabu search performed poorly when compared to previous approaches [35,37,99] and simple baseline approaches – biases in their model caused it to create too many new conversation threads.

Another recent approach by Mayfield et al. [65] approached thread disentanglement using a two-pass algorithm. In the first pass, their approach would label sentences using a negotiation framework, which would then be used to detect sequences using a single-pass clustering algorithm similar to Shen et al. [82]. In the second pass, these sequences would be grouped together into conversation threads using a cluster classifier. Their approach achieved results that approached near-human performance on an annotated, self-collected corpus.

Wang et al. [100] and Adams and Martell [3] both used connectivity matrices to establish parent-child relationships between posts. They also both used a standard TF-IDF term vector representation to compute message similarity. Their approaches first create a similarity matrix of the messages, and then create a directed graph of the messages, where link existence depends on whether message similarity exceeds a threshold value. Wang et al. created four approaches: a baseline and three extensions of the baseline using different methods to penalize time-distance between messages. Tests on synthetic chat data (they used newsgroup postings stripped of their associated metadata) showed that leveraging this information improved the baseline algorithm's ability to recover threads. Adams and Martell's approach leveraged hypernym augmentation, time-distance penalization, and nickname augmentation to determine whether a message belongs to a thread. They found that time-distance penalization had the largest impact on increasing the F-scores of their algorithm.

A couple of investigations have approached thread disentanglement from a military perspective. In tactical chat from the military domain, conversation threads can be organized around missions [28]. Duchon and Jackson [27] approached the problem of thread disentanglement using a modified semi-supervised clustering approach, with the clusters based on each individual mission. When training on an annotated corpus, their approach had precision and recall scores around 80%, but these scores dropped when it was instead trained on an unannotated corpus. They found this to be due to the way background information was leveraged, and that the algorithm also confuses targets that have near-identical descriptions. Ramachandran et al. [78] extended Adams and Martell's [3] work on message representation by leveraging military-related mission keywords to create initial sets of clusters. They created a multi-pass algorithm, developed from interviewing subject-matter experts, that finds conversations related to specific missions.

Some research investigations have looked at subproblems of thread disentanglement. Khan et al. [58] focused on finding conversation-initiating messages using pattern matching with manually-created patterns. Unfortunately, their approach failed to find many of these messages. Acar et al. [1] and Çamtepe et al. [21] both focused on finding the group of users conversing together in a conversation thread without having to analyze the contents of the messages themselves. They used synthetic data composed of chat room models with assumptions to relax the problem (e.g., assuming a user will only discuss one topic during their time in the chat room). Acar et al. used an unsupervised clustering algorithm called fuzzy c-means for this task. Çamtepe et al. implemented two algorithms, both of which use a clustering subroutine, to find the groups. Both the approaches by Acar et al. and Çamtepe et al. were able to find groups of users with high levels of accuracy, but it is uncertain how well they would perform on real data due to the abstractions of the chat room models.

Thread disentanglement has received the most of research attention so far on multiparticipant chat analysis, which is understandable considering how important it is to this research area. It has benefited recently from having a common

definition of the problem and from having publicly-available annotated corpora, allowing researchers to compare their approaches with one another. However, there are some weaknesses that need to be addressed. The definition by Elsner and Charniak [35] is becoming the standard definition, which views the problem as a clustering task. Unfortunately, this definition does not address how to determine how to cluster messages when a schism occurs, which they recognized. An alternative suggestion, as the authors mention, is to view the problem of thread disentanglement by dropping the viewpoint of clustering (or partitioning) and instead use a graph-based approach, linking children and parent utterances with the edges representing the strength of the relationship. There may be other alternatives that can better address this issue and which may then improve performance in this research area.

4.4. Topic detection

There are features of chat that make topic detection difficult when compared to news events. As chat is dynamic, multiple topics can be simultaneously active and this set will change over time. O'Neill and Martin [73] described how topic change in chat differs from real-life conversations, where people will often use discourse devices to change the topic or propose a topic based on a previous utterance. In chat, there are fewer constraints; new topics can be started whenever a user desires. Herring [46] observed that topics decayed quickly in chat.

Topic detection has many uses. For example, it could be used to assist in dynamic topic analysis [47], which is used to determine whether chat participants are staying on topic. Trausan-Matu et al. [90] leveraged topic detection to aid in determining which topics were being discussed in the context of computer supported collaborative learning. Topic detection can be used to help users find rooms of interest. This is reminiscent of the Butterfly chat tool's goal [95]; it helps users find chat rooms that match their interests by sampling public chat rooms, building term vectors of the rooms, and then making recommendations to the user based on their likes and dislikes. Topic detection can also be used to assist with building user and chat room profiles.

One approach to finding topics involves using supervised learning methods, which classify messages according to a set of pre-determined topics. Elnahrawy [34] investigated three such techniques: Naïve Bayes, k-Nearest Neighbor, and Support Vector Machines (SVM). These were applied to both chat logs and newsgroup postings, with the goal being to classify a log to a pre-determined topic. For both applications, Naïve Bayes classifiers recorded the highest accuracies, had shorter training times than the SVM, and shorter classification times than k-Nearest Neighbor. Özyurt and Köse [75] examined these same techniques for topic detection in chat composed in the Turkish language. They leveraged Khan et al.'s [58] idea of pattern matching to find threads that improve the performance of the supervised learning approaches. Unlike Elnahrawy, they were classifying individual messages, and in this context, SVMs were found to have higher accuracy than the other two classifiers. Tuulos and Tirri [91] investigated using a SVM for this problem. They leveraged the idea of social networks to improve performance, and reported small improvements in accuracy when the social networks were applied. Anjewierden et al. [7] used a Naïve Bayes classifier to classify messages into categories in an educational domain consisting of students collaboratively learning. They used both automated and manual processes to reduce the noise (e.g., misspellings) in their chat data before training and evaluating their classifier. Durham [32] used Latent Dirichlet Allocation models as feature vectors for a SVM. They focused on a binary classification task for chat posts, and found their approach worked best when considering all posts by a user as a single document.

Unsupervised learning approaches can also be used to identify topics. Their advantage is that they do not require labeled training data, and can identify topics that have not previously been found. For example, Kolenda et al. [59] used Independent Component Analysis (ICA) to find topics in a stream of chat. They applied ICA to a day of chat from a news website to find keywords of topics discussed during that day. Bingham et al. [11] presented another unsupervised approach using complexity pursuit, which is closely related to ICA. They reported tests with both an unlabeled chat stream and a labeled newsgroup data (the labels allowed for numerical measurements in evaluation). The latter tests showed that their approach was the most successful among those compared for recognizing topically different newsgroup articles.

Trausan-Matu et al. [90] created a tool for computer supported collaborative learning that can discover new topics when they are introduced during a conversation. Their approach consists of finding frequent words in the chat after irrelevant words are removed. Topics are found by leveraging synonyms to find common words, searching for patterns among the topics that are dynamically introduced, and from user feedback.

Bengel et al. [9] created *ChatTrack*, a tool that archives chat logs and creates profiles of the logs and users. These logs were collected by a bot, which sat in various IRC chat rooms and recorded the messages. This data was then used to create profiles, both in terms of the session and the various users, using a vector space classifier. These profiles would contain the topics being discussed either in the chat session or in the messages spoken by the user.

While there have been many research investigations in topic detection on chat, it is unfortunately impossible to know which approaches work best. This is due to the lack of a public corpus which that could be used for comparative evaluation. That is, the research investigations presented here used self-collected data, and, to our knowledge, there has been no effort to make a public corpus for this research topic.

Additionally, there is a lack of analysis on whether one can apply traditional techniques to chat, or if there are extensions needed to traditional algorithms due to chat's unusual characteristics. The only analysis we are aware of are the evaluations conducted by Tuulos and Tirri [91] in which they investigated whether inferring social networks increased classification accuracy. Other researchers addressed this issue, such as Özyurt and Köse [75], who considered conversation threads, and

Anjewierden et al. [7], who reduced noise in chat data, but they did not report results of whether this made a difference. Considering the popularity and the need for topic detection, it would be beneficial for future investigations to determine if special extensions are required for chat data.

4.5. Message attribute identification

Message attribute identification is used to examine messages to identify an attribute under study. One area that has been investigated is techniques for tagging words and messages, i.e., part-of-speech tagging, sentence type tagging, and dialogue act tagging. Part-of-speech tagging concerns tagging at the individual word level. For example, Forsyth and Martell [39] examined supervised learning techniques (i.e., leveraging previously-tagged text documents) to train a part-of-speech tagger. They included new tags for this parser to cover additional words not found in more formal documents (e.g., emoticons, acronyms, commonly misspelled words).

Sentence type tagging determines the *type* of a sentence (e.g., declarative sentence, conditional sentence). Shen et al. [82] used simple heuristics to determine the types of sentences. For messages composed of multiple sentences, the authors focused only on the first and last sentence of the message, as they were using sentence type tagging for thread disentanglement. Sentence type tagging did improve their performance on thread disentanglement, but the authors did not report how accurate their heuristics were in tagging sentence types.

Dialogue act tagging determines the *type of act* of a sentence (e.g., greeting, system message, Wh-question). Wu et al. [103] created templates that were fed into a Transformation-Based Learning algorithm (a greedy, rule-based learning algorithm) to learn how to tag sentences. In their work, they included dialogue acts that are specific to chat: emotion, emphasis, and system (i.e., system messages). Forsyth and Martell [39] examined dialogue-act tagging with the same tags as described by Wu et al. [103]. They applied a back-propagation neural network and a Naïve Bayes algorithm, finding that the neural network performed better at classifying messages into the set of dialogue acts. Carpenter and Fujioka [17] examined dialogue-act tagging using a larger set of dialogue act types. As with Wu et al. [103], they described four dialogue act types unique to chat: action description, completion, correction, and attention. The authors created a set of hand-coded rules based on patterns seen in their corpus, which were then used in a rule-based pattern-matching approach. While their approach worked well for their corpus, the authors warned that the high performance was because the users in the corpus had developed a shared vocabulary and common patterns of interaction, which might not be seen in other environments.

A few research efforts have focused on urgency detection — more specifically, finding messages of importance so that they can be brought to a user's attention. Berube et al. [10] combined regular expressions, information extraction, and entity classes within a system to detect important messages in military chat logs. Budlong et al. [13] leverage cues to identify uncertainty and urgency within a chat message. They compared a rule-based algorithm, a statistical analysis approach using maximum entropy, and a combined approach to accomplish this task. The maximum entropy approach had the highest recall among the three, but the combined approach recorded a much higher precision and the highest overall F-score. Dela Rosa and Ellen [26] compared four different classifiers (k-Nearest Neighbor, SVM, Rocchio, and Naïve Bayes) for classifying chat utterances. Their goal was to classify messages into either non-important messages containing only social chat or important messages containing social chat interspersed with a Navy ship update. Their results showed the k-Nearest Neighbor and SVM classifiers performed best for this task.

4.6. User profiling

User profiling aims to discover properties about a user despite the anonymity of chat. One application involves helping law enforcement personnel to locate sexual predators (i.e., automated detection of possible older participants preying on youths and children [63,88]). Another application concerns social analysis; automated methods for detecting gender can aid in studying the impact of gender in chat, as discussed in Section 3.2.

Lin [63] examined age and gender detection using Naïve Bayes classifiers, though results showed their approach did not significantly outperform a baseline approach. Tam [88] used Naïve Bayes classifiers and SVMs to detect a user's age. Their results showed that, in general, SVMs outperformed the Naïve Bayes classifier on their task. Köse et al. [60] examined gender identification in Turkish chat. They used a discrimination function with semantic analysis, and compared it against Naïve Bayes and SVM classifiers. Their approach obtained a higher accuracy than the other classifiers for this problem with less computation time.

Related to topic detection is determining which topics a user is interested in. As mentioned earlier, Bengel et al. [9] developed the tool ChatTrack, which creates user profiles. These profiles can be used to identify the topic interests of a user.

4.7. Social phenomenon detection

Social phenomenon detection is important for finding social interactions that go beyond a single user. One example of this is detecting social networks among users. Çamtepe et al. [20] describe an in-situ bot that creates profiles of the users and chat rooms. It records the number of messages a user sent within a time interval. Using this information, it then infers who was talking with whom during a particular time interval and tries to discover communities. Mutton [70] presents an IRC bot called PieSpy. It collects chat room data and uses it to heuristically infer relationships among users to create a

social-network graph (e.g., based on temporal distance, direct relations). Edges representing relationships between two users are weakened over time when they are not interacting. Thus, each graph was a snapshot of the social network at a given point in time.

Strzalkowski et al. [87] described beginning work on discovering social phenomena behaviors such as topic control, task control, and leadership. They considered a two-tiered approach, with the higher-level tier consisting of social roles such as leadership and group cohesion, which builds upon a mid-level tier consisting of social behaviors such as topic and task control. In their current work, they first used an automated process to measure topic and task control by calculating a set of indices for each. These measurements were then used to compute leadership. They evaluated their approach by surveying the participants in the chat corpus, investigating whether chat users agreed with these social phenomenon assessments.

Another area of social phenomenon detection concerns identifying deceptive users. Deceptive users are those who send messages with the aim to create false conclusions [93]. Textual computer-mediated communication hides one factor of deception detection, non-verbal cues, which increases a highly-motivated individual's success of deceiving [42]. Some investigations examined automated techniques for detecting deception in multiparticipant chat; they were motivated to detect deception in chat conversations in the finance industry. Twitchell et al. [93] used speech act profiling to find uncertainty in users, which is an indicator of deception. Their approach examines messages sent and creates a profile, which is then examined by a human user to predict if the messenger is being deceptive. Twitchell et al. [92] later used a similar approach, but changed the data used to train the speech act profiler (i.e., the previous study used non-chat data for training, while this approach used chat data). Their results again showed that uncertainty can be used to detect deception, and they also reported new social behavioral results on how deceptive users act in the chat rooms.

4.8. Automatic summarization

Automatic summarization is the process of creating a textual summary from a textual source [41,64,86]. This can be done using extractive or abstractive methods. *Extractive* methods use the same text as the original source, which is a shallow analysis, while *abstractive* methods create new text from the original source, requiring more extensive textual analysis. To evaluate summaries, it is common to compare machine-created summaries against a gold standard. One can then use automated techniques, such as ROUGE [62], or human assessors to compare the two summaries.

Automatic summarization of chat messages would be valuable to many types of users given the many uses of chat (see Section 3). For example, it could aid students with studying recorded chat sessions, summarizing meetings, or for reviewing the events that have occurred in a technical chat room, especially if it is of interest to other people. However, summarizing chat messages has challenges not seen in other media. For example, Uthus and Aha [94] described how summarizing chat has to overcome thread entanglement, which is not an issue with summarizing dialogues and recorded meetings, and how chat lacks metadata that is available for summarizing newsgroups and mailing lists.

There are different methods for summarizing chat messages. Zhou and Hovy [107] investigated methods for summarizing Linux Kernel chat logs using extractive methods to create summaries similar to human-created summaries, which were used as a gold standard. Their approach initially clustered partial messages under identified topics. From these clusters, it then created a collection of summaries, with one summary per topic. They compared their approach against two baseline approaches, and their approach achieved higher precision and F-score measurements when comparing the extracted chat messages in their summaries against the extracted messages in the human-created summaries. Uthus and Aha [94] described initial work on summarizing chat messages from a military perspective. They proposed two forms of summarization: summarizing a thread of messages and summarizing a group of messages based on temporal distance.

5. Discussion

Chat offers many unique challenges due to its informal, conversational nature, and has been a focus of many research areas. AI-related analysis of chat can still be considered in its infancy, as many problems remain open and there is still a lot of research needed to address these challenges. In this section, we highlight what we believe are some open issues: corpora creation; boundaries with other media; cross-fertilization of ideas with other forms of microtext and across tasks; and research topics in need of further attention.

5.1. Corpora creation

There is a strong need for more annotated, publicly-available chat research corpora. Currently, only a few relevant datasets exist. A set of standard corpora could make feasible the comparison of different approaches and provide baselines for research tasks that lack these. For example, topic detection approaches have all been applied to disjoint corpora, thus preventing comparative analyses. To overcome this, Bingham et al. [11] applied their techniques to a non-chat corpus to obtain labeled data to numerically test their algorithms. An example of a useful corpus is the #LINUX corpus, whose use has fostered comparative evaluations in thread disentanglement. Having similar publicly-assessable corpora would be a boon to research in multiparticipant chat analysis.

5.2. Boundaries with other media

As we have mentioned throughout this paper, chat has characteristics similar to many other media, but at the same time, it has features not found in those media. One understudied issue is the boundaries of chat compared to these other media. More specifically, when working on a task that has also been studied in other media, such as topic detection, can one apply traditional techniques or are new extensions required? Except for thread disentanglement and chat room feature processing, the tasks described here can easily be abstracted to a similar task on a different medium. Identifying these boundaries could determine when it is worthwhile to consider new features for chat, or if it is acceptable to directly apply techniques from other media to chat.

5.3. Cross-leveraging techniques

As described in Section 2.1, multiparticipant chat analysis shares some commonality with other forms of microtext, particularly its informal written content. Many techniques for normalizing the written contents in one form of microtext are applicable to other forms of microtext. For example, Bingham et al. [11] were able to apply their algorithm to both chat and newsgroups due to their related characteristics. Investigations into how other microtext techniques could be applied to chat could encourage their reuse. Fortunately, new meetings have started to encourage cross-fertilization of ideas, such as the AAAI 2011 Workshop on *Analyzing Microtext* [4] and the upcoming AAAI 2013 Spring Symposium of the same name.

Focusing on chat, there has been little research on cross-leveraging techniques from one research task to achieve better results on another. For example, despite the increasing amount of research on thread disentanglement, the ideas and methods have not been re-used for other tasks such as topic detection, social phenomenon detection, and automatic summarization. New analyses on how techniques can be cross-leveraged could lead to better solutions to research tasks and help to identify weaknesses in past approaches.

5.4. Underexplored research foci

In addition to the issues we have described in previous sections, there are some challenging problems that have not received much research attention on multiparticipant chat analysis.

5.4.1. Machine translation

One unexplored research task is machine translation. There are motivating applications for researching machine translation in chat, such as online games [105], distributed (global) software engineering [16], and international military coalitions [71]. There has been some work on comparing off-the-shelf software (e.g., Google Translate compared to Apertium [16] and BizLingo [72]), how users can build common ground using machine translation [104], and how to integrate machine translation into 3D online virtual spaces such as Second Life [105].

To our knowledge, none of these approaches have worked well; many problems arose in the translations. For example, Yamashita et al. [104] studied users solving a task in a controlled environment, and they found that users could solve the tasks more quickly using English as a common language instead of using their native languages (Chinese, Korean, and Japanese) translated by the chat client. This was in part from the difficulty of finding a common reference that they could share for the task being solved due to errors in the translations. Additionally, none of these approaches were designed to handle the unique characteristics of multiparticipant chat. While machine translation algorithms are imperfect, finding ways to better incorporate machine translation in chat communications could lead to better translation quality. An example is finding a way to transform chat messages into more formal text, possibly using synonyms or other word features, which would then be easier to translate.

5.4.2. Bots – design and implications

Another underexplored area is when artificial agents, such as *bots* in chat rooms or *non-player characters* in video games, are participating in chat conversations. There are two areas of concern: designing an agent that can participate in chat conversations, and analyzing chat when some users are not human.

Some bots have been designed to communicate with humans in multiparticipant settings, though usually in a one-to-one conversation. For example, Cobot [52] was designed to sit in a virtual chat room in LambdaMOO and interact with other users. However, from a chat perspective, it could respond to only utterances directed at it, or randomly try to introduce a topic (using external text sources) without regard to what is currently being discussed. A recent paper described initial work with a virtual chat agent [81]. This agent analyzed utterances using word matching, and then used a template to output a message to the chat room with the goal of changing the topic of the chat room. Substantial work would be needed to design a more intelligent bot that could actively understand and participate in chat rooms with multiple participants. This may incorporate ideas from other research on multiparticipant chat analysis described in this paper.

Most research described in this paper assumed the users were human. However, if the users are not human, this could impact tasks such as topic detection (which is a focus of the two previously mentioned bots), user profiling, and social phenomenon detection. In addition, should bots become more intelligent, this could lead to new research areas of trying

to detect bots in a chat setting. An example of this is the work by McIntire et al. [66]. They examined passive (e.g., communication pattern analysis) and active (e.g., human interrogation strategies) methods for detecting malicious “chatbots”, which they applied to transcripts of one-to-one conversations competing in Turing tests. Their passive methods were successful in identifying chatbots in the small set of chat data they examined, though they recommend further research as their approaches would fail as chatbots become more intelligent.

5.4.3. Intelligent user interfaces

The final research foci we discuss concerns intelligent user interfaces. We have already described some research on this topic (see Section 3.1). Additional studies have been conducted in the military context. For example, the Tactical Situation Assessment Technologies (TSAT) project [40,68,69] created a client that permits the user to communicate without a keyboard and use icons to represent words. The authors describe needed advancements in chat analysis to discover topics likely to reoccur and to analyze large databases of chat text. Also, Boiney et al. [12] introduced an approach for military chat room management using visual cues to identify when a chat room becomes more active. They address a common problem: operators may have many concurrently open chat rooms, but can focus on only a few at a time. New research on intelligence user interfaces could lead to clients that incorporate many of the techniques described in this survey, such as those for thread disentanglement and topic detection, making chat easier to use and follow.

6. Conclusion

We have surveyed several tasks concerning multiparticipant chat analysis and we reviewed areas of research that motivated these tasks. Chat is a difficult medium to analyze due to its unique characteristics. This has caused many traditional text analysis techniques to perform poorly when applied, leading to new areas of research for chat. Among the many tasks that have been explored are chat preprocessing, chat room feature processing, thread disentanglement, topic detection, social phenomenon detection, user profiling, message attribute identification, and automatic summarization. We have also described open areas and challenging problems in chat.

To conclude, there is a lack of widely-deployed techniques for automated analysis of multiparticipant chat, which if available could aid a large variety of users and data analysts. Despite the rise of other forms of social communication, chat is still a predominant form of communications in many research and application environments. Thus, multiparticipant chat analysis is an exciting research topic in AI with many research problems whose solutions could have substantial social impact.

Acknowledgements

Thanks to NRL for funding this research and to Jeffrey Ellen and Micha Elsner for their feedback on this paper. David Uthus performed this work while an NRC postdoctoral fellow located at the Naval Research Laboratory. The views and opinions contained in this paper are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of NRL or the DoD.

References

- [1] E. Acar, S.A. Çamtepe, M.S. Krishnamoorthy, B. Yener, Modeling and multiway analysis of chatroom tensors, in: P. Kantor, G. Muresan, F. Roberts, D.D. Zeng, F.-Y. Wang, H. Chen, R.C. Merkle (Eds.), *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, in: *Lecture Notes in Computer Science*, vol. 3495, Springer, Berlin/Heidelberg, 2005, pp. 181–199.
- [2] P.H. Adams, Conversation thread extraction and topic detection in text-based chat, Master's Thesis, Naval Postgraduate School, 2008.
- [3] P.H. Adams, C.H. Martell, Topic detection and extraction in chat, in: *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, IEEE Computer Society, 2008, pp. 581–588.
- [4] D.W. Aha, D.W. Oard, S. Ramachandran, D.C. Uthus, Papers from the 2011 AAAI Workshop on Analyzing Microtext, Technical Report WS-11-05, AAAI, 2011.
- [5] J. Allan, J. Carbonell, G. Doddington, J. Yamron, Y. Yang, Topic detection and tracking pilot study final report, in: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998.
- [6] J. Androutsopoulos, E. Ziegler, Exploring language variation on the internet: Regional speech in a chat community, in: *Proceedings of the Second International Conference on Language Variation in Europe*, Uppsala University Press, 2004, pp. 99–111.
- [7] A. Anjewierden, B. Kollöffel, C. Hulshof, Towards educational data mining: Using data mining methods for automated chat analysis to understand and support inquiry learning processes, in: *Proceedings of the International Workshop on Applying Data Mining in e-Learning*, 2007, pp. 23–32.
- [8] P.M. Aoki, M.H. Szymanski, L. Plurkowski, J.D. Thornton, A. Woodruff, W. Yi, Where's the “party” in “multi-party”? Analyzing the structure of small-group sociable talk, in: *Proceedings of the Conference on Computer Supported Cooperative Work*, ACM, 2006, pp. 393–402.
- [9] J. Bengel, S. Gauch, E. Mittur, R. Vijayaraghavan, ChatTrack: Chat room topic detection using classification, in: H. Chen, R. Moore, D.D. Zeng, J. Leavitt (Eds.), *Proceedings of the Second Symposium on Intelligence and Security Informatics*, in: *Lecture Notes in Computer Science*, vol. 3073, Springer, Berlin/Heidelberg, 2004, pp. 266–277.
- [10] C.D. Berube, J.M. Hitzeman, R.J. Holland, R.L. Anapol, S.R. Moore, Supporting chat exploitation in DoD enterprises, in: *Proceedings of the International Command and Control Research and Technology Symposium*, CCRP, 2007.
- [11] E. Bingham, A. Kabán, M. Girolami, Topic identification in dynamical text by complexity pursuit, *Neural Processing Letters* 17 (2003) 69–83.
- [12] L.G. Boiney, B. Goodman, R. Gaimari, J. Zarrella, C. Berube, J. Hitzeman, Taming multiple chat room collaboration: Real-time visual cues to social networks and emerging threads, in: *Proceedings of the Fifth International ISCRAM Conference*, ISCRAM, 2008, pp. 660–668.
- [13] E.R. Budlong, S.M. Walter, O. Yilmazel, Recognizing connotative meaning in military chat communications, in: *Proceedings of Evolutionary and Bio-Inspired Computation: Theory and Applications III*, SPIE, 2009.

- [14] A. Cadilhac, N. Asher, F. Benamara, Annotating preferences in chats for strategic games, in: *Proceedings of the Sixth Linguistic Annotation Workshop, ACL*, 2012, pp. 139–143.
- [15] M. Cakir, F. Xhafa, N. Zhou, G. Stahl, Thread-based analysis of patterns of collaborative interaction in chat, in: *Proceeding of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, IOS Press, 2005, pp. 120–127.
- [16] F. Calefato, F. Lanubile, P. Minervini, Can real-time machine translation overcome language barriers in distributed requirements engineering?, in: *Proceedings of the Fifth International Conference on Global Software Engineering, IEEE Computer Society*, 2010, pp. 257–264.
- [17] T. Carpenter, E. Fujioka, The role and identification of dialog acts in online chat, in: *Proceedings of the AAAI Workshop on Analyzing Microtext*, AAAI Press, 2011.
- [18] M. Cataldo, J.D. Herbsleb, Communication networks in geographically distributed software development, in: *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, ACM, 2008, pp. 579–588.
- [19] J.M. Catanzaro, M.R. Risser, J.W. Gwynne, D.I. Manes, Military situation awareness: Facilitating critical event detection in chat, in: *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*, vol. 50, Human Factors and Ergonomics Society, 2006, pp. 560–564.
- [20] A. Çamtepe, M.S. Krishnamoorthy, B. Yener, A tool for internet chatroom surveillance, in: H. Chen, R. Moore, D.D. Zeng, J. Leavitt (Eds.), *Proceedings of the Second Symposium on Intelligence and Security Informatics*, in: *Lecture Notes in Computer Science*, vol. 3073, Springer, Berlin/Heidelberg, 2004, pp. 252–265.
- [21] S.A. Çamtepe, M. Goldberg, M. Magdon-Ismael, M. Krishnamoorthy, Detecting conversing groups of chatters: A model, algorithms, and tests, in: *Proceedings of the IADIS International Conference on Applied Computing, IADIS*, 2005, pp. 89–96.
- [22] K.W. Church, W.A. Gale, Probability scoring for spelling correction, *Statistics and Computing* 1 (1991) 93–103.
- [23] M. Cummings, The need for command and control instant message adaptive interfaces: Lessons learned from Tactical Tomahawk human-in-the-loop simulations, *CyberPsychology & Behavior* 7 (2004) 653–661.
- [24] B. Danet, T. Bechar-Israëli, A. Cividalli, Y. Rosenbaum-Tamari, Curtain time 20:00 GMT: Experiments with virtual theater on Internet Relay Chat, *Journal of Computer-Mediated Communication* 1 (1995).
- [25] B. Danet, L. Ruedenberg-Wright, Y. Rosenbaum-Tamari, “Hmmm...where’s that smoke coming from?” Writing, play and performance in Internet Relay Chat, *Journal of Computer-Mediated Communication* 2 (1997).
- [26] K. Dela Rosa, J. Ellen, Text classification methodologies applied to micro-text in military chat, in: *Proceedings of the International Conference on Machine Learning and Applications*, IEEE Computer Society, 2009, pp. 710–714.
- [27] A. Duchon, C. Jackson, Chat analysis for after action review, in: *Proceedings of the Interservice/Industry Training, Simulation & Education Conference, I/ITSEC*, 2010.
- [28] A. Duchon, R. McCormack, B. Riordan, C. Shabarekh, S. Weil, I. Yohai, Analysis of C2 and “C2-Lite” micro-message communications, in: *Proceedings of the AAAI Workshop on Analyzing Microtext*, AAAI Press, 2011.
- [29] L. Duffy, T. Bilbrey, US Navy collaboration and chat systems: Current status, 2008. Presentation given at the Spring Meeting of the Collaborative Interoperability Working Group.
- [30] L.T. Duffy, Internet Relay Chat (IRC): How a grassroots technology became an integral part of operational command and control, in: *SSC San Diego Biennial Review*, 2006.
- [31] L.T. Duffy, DoD collaboration and chat systems: Current status and way ahead, in: *Proceedings of the International Conference on Semantic Computing*, IEEE Computer Society, 2008, pp. 573–576.
- [32] J.S. Durham, Topic detection in online chat, Master’s Thesis, Naval Postgraduate School, 2009.
- [33] J. Ellen, All about microtext: A working definition and a survey of current microtext research within artificial intelligence and natural language processing, in: *Proceedings of the Third International Conference on Agents and Artificial Intelligence*, SciTePress Digital Library, 2011.
- [34] E.M. Elnahrawy, Log-based chat room monitoring using text categorization: A comparative study, in: *International Association of Science and Technology for Development Conference on Information and Knowledge Sharing*, ACTA Press, 2002.
- [35] M. Elsner, E. Charniak, Disentangling chat, *Computational Linguistics* 36 (2010) 389–409.
- [36] M. Elsner, E. Charniak, Disentangling chat with local coherence models, in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL*, 2011, pp. 1179–1189.
- [37] M. Elsner, W. Schudy, Bounding and comparing methods for correlation clustering beyond ILP, in: *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing, ACL*, 2009, pp. 19–27.
- [38] B.A. Eovito, The impact of synchronous text-based chat on military command and control, in: *Proceedings of the Command and Control Research and Technology Symposium*, CCRP, 2006.
- [39] E.N. Forsyth, C.H. Martell, Lexical and discourse analysis of online chat dialog, in: *Proceedings of the International Conference on Semantic Computing*, IEEE Computer Society, 2007, pp. 19–26.
- [40] S. Fugate, E.W. Medina, L. Duffy, D. Magsombo, O. Amezcua, G. Rogers, M.G. Ceruti, Next-generation tactical-situation-assessment technology (TSAT): Iconic language, in: *Proceedings of the First Annual Visual and Iconic Language Conference*, 2007.
- [41] V. Gupta, G. Lehal, A survey of text summarization extractive techniques, *Journal of Emerging Technologies in Web Intelligence* 2 (2010).
- [42] J. Hancock, M. Woodworth, S. Goorha, See no evil: The effect of communication medium and motivation on deception detection, *Group Decision and Negotiation* 19 (2010) 327–343.
- [43] M. Handel, J.D. Herbsleb, What is chat doing in the workplace?, in: *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, ACM, 2002, pp. 1–10.
- [44] N.J. Heacox, R.A. Moore, J.G. Morrison, R.F. Yturralde, Real-time online communications: ‘Chat’ use in navy operations, in: *Proceedings of the Command and Control Research and Technology Symposium*, CCRP, 2004.
- [45] S.C. Herring, Introduction, in: S.C. Herring (Ed.), *Computer-Mediated Communication: Linguistic, Social and Cross-Cultural Perspectives*, John Benjamins, 1996, pp. 1–10.
- [46] S.C. Herring, Interactional coherence in CMC, in: *Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, 1999.
- [47] S.C. Herring, Dynamic topic analysis of synchronous chat, in: *Symposium on New Research for New Media*, 2003.
- [48] S.C. Herring, Computer-mediated conversation: Introduction and overview, *Language@Internet* 7 (2010). Article 2.
- [49] S.C. Herring, Relevance in computer-mediated conversation, in: S. Herring, D. Stein, T. Virtanen (Eds.), *Handbook of the Pragmatics of Computer-Mediated Communication*, Mouton de Gruyter, 2013.
- [50] S.C. Herring, D.O. Kutz, J.C. Paolillo, A. Zelenkauskaitė, Fast talking, fast shooting: Text chat in an online first-person game, in: *Proceedings of the Forty-Second Hawaii International Conference on System Sciences*, IEEE Computer Society, 2009.
- [51] T. Holmer, Discourse structure analysis of chat communication, *Language@Internet* 5 (2008), Article 9.
- [52] C.L. Isbell, M. Kearns, S. Singh, C.R. Shelton, P. Stone, D. Kormann, Cobot in LambdaMOO: An adaptive social statistics agent, *Autonomous Agents and Multi-Agent Systems* 13 (2006) 327–354.
- [53] T. Jara, M. Lisowski, Don’t silence navy chat, *Proceedings of the U.S. Naval Institute* 129 (2003) 52–55.
- [54] K. Jepson, Conversations – and negotiated interaction – in text and voice chat rooms, *Language Learning and Technology* 9 (2005) 79–98.

- [55] C. Kalt, Internet Relay Chat: Client protocol, <http://www.irchelp.org/irchelp/rfc/rfc2812.txt>, 2000.
- [56] S. Kapidzic, S.C. Herring, Gender, communication, and self-presentation in teen chatrooms revisited: Have patterns changed?, 2010. Paper presented at the 96th Annual Convention of the National Communication Association, San Francisco, November 14–17.
- [57] H. Katsuno, C. Yano, Kaomoji and expressivity in a Japanese housewives' chat room, in: *The Multilingual Internet: Language, Culture, and Communication Online*, Oxford University Press, 2007, pp. 278–301.
- [58] F. Khan, T. Fisher, L. Shuler, T. Wu, W. Pottenger, Mining chat-room conversations for social and semantic interactions, Technical Report LU-CSE-02-011, 2002.
- [59] T. Kolenda, L.K. Hansen, J. Larsen, Signal detection using ICA: Application to chat room topic spotting, in: *Proceedings of the Third International Conference on Independent Component Analysis and Blind Source Separation*, 2001, pp. 540–545.
- [60] C. Köse, O. Özyurt, C. İkibaş, A comparison of textual data mining methods for sex identification in chat conversations, in: *Proceedings of the Fourth Asia Information Retrieval Conference on Information Retrieval Technology*, Springer-Verlag, 2008, pp. 638–643.
- [61] S. Landau, Chat guidance and applications, Technical Report Z-3404-06-001, 2006.
- [62] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: M.-F. Moens, S. Szpakowicz (Eds.), *Proceedings of the ACL-04 Workshop on Text Summarization Branches Out*, 2004, pp. 74–81.
- [63] J. Lin, Automatic author profiling of online chat logs, Master's Thesis, Naval Postgraduate School, 2007.
- [64] J. Lin, Summarization, in: M.T. Özsu, L. Liu (Eds.), *Encyclopedia of Database Systems*, Springer, 2009.
- [65] E. Mayfield, D. Adamson, C. Penstein Rosé, Hierarchical conversation structure prediction in multi-party chat, in: *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, Seoul, South Korea, ACL, 2012, pp. 60–69.
- [66] J.P. McIntire, L.K. McIntire, P.R. Havig, Methods for chatbot detection in distributed text-based communications, in: *Proceedings of the International Symposium on Collaborative Technologies and Systems*, 2010, pp. 463–472.
- [67] E.W. Medina, Military textual analysis and chat research, in: *Proceedings of the IEEE International Conference on Semantic Computing*, IEEE Computer Society, 2008, pp. 569–572.
- [68] E.W. Medina, L.T. Duffy, Tactical situation assessment technologies (TSAT), in: *Proceedings of the IEEE International Conference on Semantic Computing*, IEEE Computer Society, 2008, pp. 577–580.
- [69] E.W. Medina, S. Fugate, L. Duffy, D. Magsombol, O. Amezcua, G.J. Rogers, M.G. Ceruti, Next-generation tactical-situation-assessment technology (TSAT): Chat, in: *Proceedings of the Seventh International Conference on Web Engineering*, Springer-Verlag, 2007, pp. 526–532.
- [70] P. Mutton, Inferring and visualizing social networks on Internet Relay Chat, in: *Proceedings of the Eighth International Conference on Information Visualisation*, IEEE Computer Society, 2004, pp. 35–43.
- [71] W. Ogden, A task-based evaluation method for embedded machine translation in instant messaging systems, in: *Advanced Decision Architectures for the Warfighter: Foundation and Technology*, Alion Science and Technology Corp., 2009.
- [72] W. Ogden, R. Zacharski, S. An, Y. Ishikawa, User choice as an evaluation metric for web translation services in cross language instant messaging applications, in: *Machine Translation Summit VII*, 2009.
- [73] J. O'Neill, D. Martin, Text chat in action, in: *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work*, ACM, 2003, pp. 40–49.
- [74] G. Osman, S.C. Herring, Interaction, facilitation, and deep learning in cross-cultural chat: A case study, *The Internet and Higher Education* 10 (2007) 125–141.
- [75] O. Özyurt, C. Köse, Chat mining: Automatically determination of chat conversations' topic in Turkish text based chat mediums, *Expert Systems with Applications* 37 (2010) 8705–8710.
- [76] S. Panyametheekul, S.C. Herring, Gender and turn allocation in a Thai chat room, in: *The Multilingual Internet: Language, Culture, and Communication Online*, Oxford University Press, 2007, pp. 233–255.
- [77] J.C. Paolillo, Language variation on Internet Relay Chat: A social network approach, *Journal of Sociolinguistics* 5 (2001) 180–213.
- [78] S. Ramachandran, R. Jensen, O. Bascara, T. Carpenter, T. Denning, L.S. Succillon, Untangling topic threads in chat-based communication: A case study, in: *Proceedings of the AAAI Workshop on Analyzing Microtext*, AAAI Press, 2011.
- [79] S. Ramachandran, R. Jensen, O. Bascara, T. Denning, S. Succillon, After action review tools for team training with chat communications, in: *Proceedings of the Interservice/Industry Training, Simulation & Education Conference*, I/ITSEC, 2009.
- [80] E.M. Reid, *Electropolis: Communication and community on Internet Relay Chat*, University of Melbourne Honours Thesis, 1991.
- [81] S. Shaikh, T. Strzalkowski, S. Taylor, N. Webb, VCA: An experiment with a multiparty virtual chat agent, in: *Proceedings of the 2010 Workshop on Companionable Dialogue Systems*, Uppsala, Sweden, ACL, 2010, pp. 43–48.
- [82] D. Shen, Q. Yang, J.-T. Sun, Z. Chen, Thread detection in dynamic text message streams, in: *Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2006, pp. 35–42.
- [83] E. Shihab, Z.M. Jiang, A.E. Hassan, Studying the use of developer IRC meetings in open source projects, in: *Proceedings of the IEEE International Conference on Software Maintenance*, IEEE Computer Society, 2009, pp. 147–156.
- [84] B. Siebenhaar, Code choice and code-switching in Swiss-German Internet Relay Chat rooms, *Journal of Sociolinguistics* 10 (2006) 481–506.
- [85] M. Smith, J. Cadiz, B. Burkhalter, Conversation trees and threaded chats, in: *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, ACM, 2000, pp. 97–105.
- [86] K. Spärck Jones, Automatic summarising: The state of the art, *Information Processing & Management* 43 (2007) 1449–1481.
- [87] T. Strzalkowski, G.A. Broadwell, J. Stromer-Galley, S. Shaikh, T. Liu, S. Taylor, Modeling socio-cultural phenomena in online multi-party discourse, in: *Proceedings of the AAAI Workshop on Analyzing Microtext*, AAAI Press, 2011.
- [88] J.K. Tam, Detecting age in online chat, Master's Thesis, Naval Postgraduate School, 2009.
- [89] E. Toyoda, R. Harrison, Categorization of text chat communication between learners and native speakers of Japanese, *Language Learning and Technology* 6 (2002) 82–99.
- [90] S. Trausan-Matu, T. Rebedea, A. Dragan, C. Alexandru, Visualisation of learners' contributions in chat conversations, in: *Proceedings of the Workshop on Blended Learning*, Pearson, 2007, pp. 217–226.
- [91] V.H. Tuulos, H. Tirri, Combining topic models and social networks for chat data mining, in: *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE Computer Society, 2004, pp. 206–213.
- [92] D.P. Twitchell, N. Forsgren, K. Wiers, J.K. Burgoon, J.F. Nunamaker Jr., Detecting deception in synchronous computer-mediated communication using speech act profiling, in: P. Kantor, G. Muresan, F. Roberts, D.D. Zeng, F.-Y. Wang, H. Chen, R.C. Merkle (Eds.), *Intelligence and Security Informatics*, in: *Lecture Notes in Computer Science*, vol. 3495, Springer, Berlin/Heidelberg, 2005, pp. 1355–1374.
- [93] D.P. Twitchell, J.F. Nunamaker Jr., J.K. Burgoon, Using speech act profiling for deception detection, in: H. Chen, R. Moore, D.D. Zeng, J. Leavitt (Eds.), *Intelligence and Security Informatics*, in: *Lecture Notes in Computer Science*, vol. 3073, Springer, Berlin/Heidelberg, 2004, pp. 403–410.
- [94] D.C. Uthus, D.W. Aha, Plans toward automated chat summarization, in: *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, ACL, 2011, pp. 1–7.
- [95] N.W. Van Dyke, H. Lieberman, P. Maes, Butterfly: A conversation-finding agent for Internet Relay Chat, in: *Proceedings of the Forth International Conference on Intelligent User Interfaces*, ACM, 1999, pp. 39–41.

- [96] F.B. Viégas, J.S. Donath, Chat circles, in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1999, pp. 9–16.
- [97] D. Vronay, M. Smith, S. Drucker, Alternative interfaces for chat, in: *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*, ACM, 1999, pp. 19–26.
- [98] L. Wang, Y. Jia, Y. Chen, Conversation extraction in dynamic text message stream, *Journal of Computers* 3 (2008) 86–93.
- [99] L. Wang, D.W. Oard, Context-based message expansion for disentanglement of interleaved text conversations, in: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ACL, 2009, pp. 200–208.
- [100] Y.-C. Wang, M. Joshi, W.W. Cohen, C. Rosé, Recovering implicit thread structure in newsgroup style conversations, in: *Proceedings of the Second International Conference on Weblogs and Social Media*, AAAI, 2008, pp. 152–160.
- [101] C.C. Werry, Linguistic and interactional features of Internet Relay Chat, in: S.C. Herring (Ed.), *Computer-Mediated Communication: Linguistic, Social and Cross-Cultural Perspectives*, John Benjamins, 1996, pp. 47–64.
- [102] K.-F. Wong, Y. Xia, Normalization of Chinese chat language, *Language Resources and Evaluation* 42 (2008) 219–242.
- [103] T. Wu, F.M. Khan, T.A. Fisher, L.A. Shuler, W.M. Pottenger, Posting act tagging using transformation-based learning, in: T. Young Lin, S. Ohsuga, C.-J. Liao, X. Hu, S. Tsumoto (Eds.), *Foundations of Data Mining and Knowledge Discovery*, in: *Studies in Computational Intelligence*, vol. 6, Springer, Berlin/Heidelberg, 2005, pp. 321–331.
- [104] N. Yamashita, R. Inaba, H. Kuzuoka, T. Ishida, Difficulties in establishing common ground in multiparty groups using machine translation, in: *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, ACM, 2009, pp. 679–688.
- [105] T. Yoshino, K. Ikenobu, Availability of multilingual chat communication in 3D online virtual space, in: T. Ishida (Ed.), *Culture and Computing*, in: *Lecture Notes in Computer Science*, vol. 6259, Springer, Berlin/Heidelberg, 2010, pp. 28–40.
- [106] A. Zelenkauskaitė, S.C. Herring, Gender encoding of typographical elements in Lithuanian and Croatian IRC, in: *Proceedings of Cultural Attitudes Towards Technology and Culture 2006*, Murdoch University Press, 2006.
- [107] L. Zhou, E. Hovy, Digesting virtual “geek” culture: The summarization of technical Internet Relay Chats, in: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL, 2005, pp. 298–305.