Discourse Structure Analysis of Chat Communication

Torsten Holmer
Upper Austria University of Applied Sciences Hagenberg, Austria
urn:nbn:de:0009-7-16339

Abstract

This article presents a research method called *Discourse Structure Analysis* (*DSA*) and a software application called *ChatLine* that supports the analysis of chat transcripts according to DSA. The DSA method is based on manual referencing and automatic analysis of chat transcripts in order to create visualizations and measures of their message and interaction structures. The goal of DSA is to provide a comprehensive and extensible method for the data-driven analysis of chat logs that can support both qualitative and quantitative investigations of computer-mediated communication.

Introduction

Chat communication has been, and remains, one of the primary areas of interest in Computer-Mediated Discourse Analysis (Herring, 2004), because discussions via chat are very different from face-to-face discussions (Beißwenger, this issue; Black, Levin, Mehan, & Quinn, 1983; Garcia & Jacobs, 1998, 1999; Herring, 1999). The technology allows many users to "talk" to each other at the same time in multi-party dialogue or polylogue while being physically distant. In multiparticipant, public chat like Internet Relay Chat (IRC), unrestricted access to the shared communication channel allows multiple concurrent threads, which often results in complex chat discussion.

Previous research has focused mainly on the influence of chat as a medium of interaction on the linguistic aspects of messages (e.g., oral style, abbreviations, emoticons) and, to a lesser extent, on the structure of chat discourse (e.g., turn taking, interactional coherence). Although the incoherence of message sequences is one of the most obvious features of a chat log, only a few studies have analyzed the characteristics of these structures and used them for analyzing underlying communication patterns (e.g., Herring & Kurtz, 2006; McDaniel, Olson, & Magee, 1996). Coherence as a quality of chat discourse was intensively addressed by Cornelius and Boos (2003), who developed a coherence measure based on the topics of discussions. Message flows with alternating topics were rated as incoherent, while message sequences on the same topic were coded as coherent. Shi,

Mishra, Bonk, Tan, and Zhao (2006) also used topic as the indicator for whether messages belonged to the same thread. These approaches consider threads to be linear sequences of messages and neglect the possibility that threads of the same topic can split into subthreads, a phenomenon which Egbert (1997) calls schisming. An exception is Herring and Kurtz (2006), who consider the splitting of threads and developed visualizations for these structures, as well as investigating the structure of topical coherence, in which the digression of topics is measured and visualized (Herring, 2003).

The structural properties of online discourse can be used to analyze underlying communication behavior and social structure. Shi et al. (2006) analyzed chat logs and identified the behavior of multitasking, defined as alternating participation in parallel threads. Hara, Bonk, and Angeli (2000) and Gerosa, Pimental, Fuks, and Lucena (2004, 2005) showed for asynchronous discussion forums that the analysis of message structure provides important information that can be used to understand and support communicating participants. While Hara et al. (2000) derived social interaction networks from the relationships between asynchronous messages, Mutton (2004) developed an algorithm to detect exchange patterns in synchronous online discourse based on several heuristics (e.g., mentioning addressee name and response time). The fundamental assumption of these approaches is that sender-receiver relationships can be used for the creation of social networks. In contrast, Rafaeli and Sudweeks (1997) distinguish between declarative (one-way), reactive (two-way), and interactive (dependent) communication. Interactive communication is defined as an alternating continuous exchange of messages between participants in which the messages are not only related to the previous but also to earlier messages instead of simple initiation-response pairs.

Until now, all these approaches have had to be applied separately to the same data, which increases the amount of work required. Moreover, most of the analyses have to be done manually, which hinders the investigation of large chat corpora and the comparison of chat logs on a larger scale. The aim of Discourse Structure Analysis (DSA) is to provide an approach that combines different methods in a comprehensive and extensible way and is implemented in software for automation. In this way, the analysis of large corpora of chat logs can be accelerated and the development and testing of research hypotheses regarding chat communication can be enhanced. The basic idea behind DSA is that the identification of references between messages offers an important key to the analysis of chat communication. Once the structure of these references is identified, a number of measures and visualizations can

be derived by formal analysis without further coding activities. In other words, the "coding and counting" approach that is applied in CMDA (Herring, 2004) is replaced in DSA by a "coding, computing, and counting" approach.

This offers a method of analysis for chat logs in which the amount of manual coding is minimized in order to save resources for, e.g., in-depth analysis of the communication patterns within a chat log. Qualitative analysis is enhanced through supporting visualizations of the discourse structure that show the dynamics of interaction and disentangle intertwined communication threads. The resulting functionality for analyzing and comparing multiple chat logs makes it possible to address research questions that focus on more quantitative aspects (e.g., amount of participation in different threads) and also comparisons of quantitative aspects across a large sample of chat logs (e.g., participation patterns in different IRC channels).[1]

The Method of Discourse Structure Analysis

The discourse structure analysis process consists of four steps: importing, referencing, discourse structure building, and analysis. After chat logs of arbitrary formats are imported into a specific data format, each chat log must be referenced by a coder. Each message must also be coded with respect to a relationship to other messages that could be interpreted as that of an utterance pair (Schegloff & Sacks, 1973), although the messages need not be adjacent in the chat log. These references are used to create the discourse structure, which may consist of several branched threads. This structure is analyzed by *ChatLine* with respect to interaction phenomena such as dialogues, parallel discussions, multitasking participation, and so forth. The results of the analysis describe the discourse structure, individual communication behaviors of the members, and the social network structure of the group.

The discourse structure is visualized by different methods that emphasize selected aspects of the structure and support subsequent qualitative analysis. Each chat log and its measurement data are stored in a single file, which is part of the chat log corpus. These files can be analyzed individually or in groups using the *ChatLine* software.

The following sections explain the process steps of referencing, analyzing, and visualizing.

Finding and Coding References

The reader of a chat log has to cope with the problem of inferring meaningful relationships from the often incoherent structure of chat messages. She has to find questions for answers, openings for closings, and other ties or relations among messages. In this hermeneutic activity, she is trying to find matches between pairs of messages and deciding on the most reasonable relationship that fits the context of the discussion. This is also the situation for the coder of a chat log, who has to code the references between messages. For each message, the coder has to find the answer to the question: Which is the message that would occour immediately before the current message if there where no intervening messages from other discussion threads? In order to explain the resulting data, an example from Vronay, Smith, and Drucker (1999) was manually coded for inter-message references; the results are shown in Table 1.

Reference ID	Order ID	Message
1	1	Black: Did you see that new Mel Gibson movie - I think it is called "Payback"?
2	2	Pink: I saw the academy awards last night. Did you watch it?
1	3	Pink: yep.
3	4	Pink: It was very violent, but funny.
3	5	Black: You saw it? You liked it?
2	6	Black: How did it end up - who won?
1	7	Red: I heard it was good.
6	8	Pink: It was OK. At least Titanic didn't win everything.

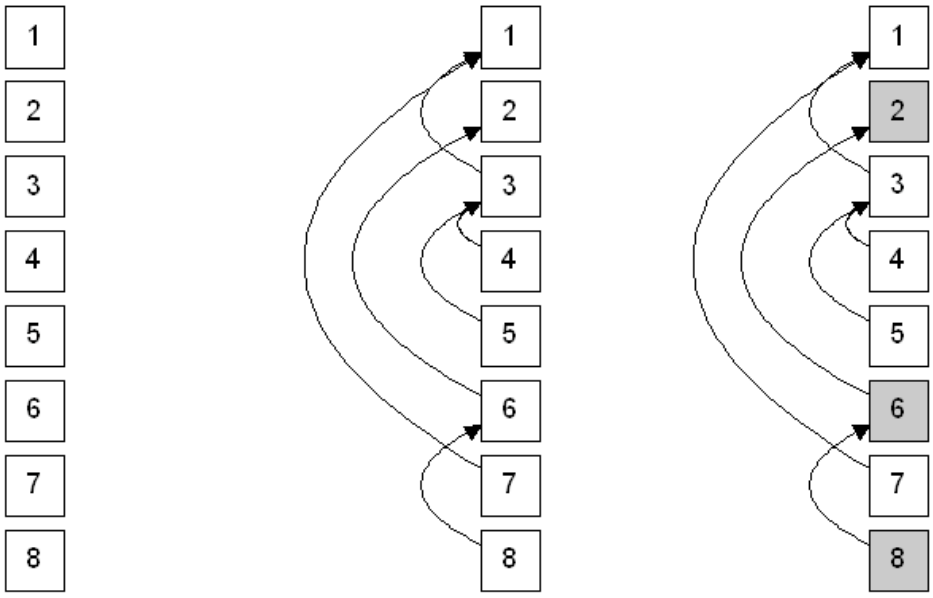
Table 1. Manually referenced chat log example from Vronay at al. (1999)

The message column features the messages of the chat log in the original order, which is indicated by the entries in the Order ID column. The Reference ID column contains the number of the Order ID of the referred message. If a message has no referring message, then the number of the Order ID is identical with its Reference ID; the message refers to itself.

In the example above, the first two messages open up new topics, and the senders do not seem to react to one another. But in the third message, the user Pink is responding to Black's question and is complementing his or her former statement in the fourth message. In the fifth message, Black is responding to Pink's affirmation of having seen the Mel Gibson movie and in the sixth message Black is asking several questions about the academy awards. In the seventh message, the user Red comes into play with a statement that seems to respond to the first message. In the eighth message, Pink is responding to Black's last question regarding the awards.

In cases where the content of a message references more than one other message, the coder has two choices. One is to split the messages in two separate parts and give each message a different reference. Herring and Kurtz (2006) adopt this approach, which requires a researcher to read the chat log before importing it, because the log itself has to be modified (e.g., the message entries have to be split and the modified content has to be copied into the log). This approach is recommended when different parts of the message have different addressees. If a message refers to multiple messages by the same author, then referencing the last possible message is recommended if the chat log should not be manipulated by splitting messages (e.g., in order to get an accurate count of messages). The idea of making multiple references to multiple preceding messages has not yet been implemented in DSA, because it creates a new set of problems, such as deciding if the references are of the same kind or not. This would require additional decisions on the part of the coder and a more complex coding scheme, which would have to be learned. It would also require a new method of calculating the metrics, e.g., sender-receiver ratio (some messages are entirely addressed to a user, others only partly so). The trade-off between this additional complexity and the low frequency of messages with multiple references seems too high.

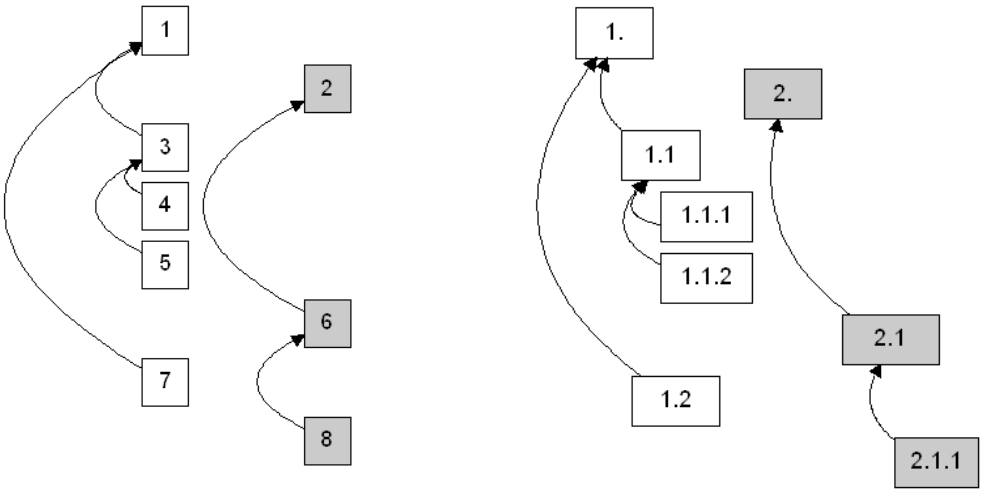
The coding of references is the only part of DSA that has to be done manually. It is supported by the *ChatLine* software, in the sense that the coder is supported by a user interface to find the message pairs. The resulting steps of the analysis are done by implemented procedures, as shown in Figure 1 for the Vronay et al. (1999) sample.



1. unreferenced chat log

2. referenced chat log

3. building of threads



4. separation of threads

5. indexing messages

Figure 1. Processing steps in DSA for building the discourse structure

Figure 1 shows the process of DSA that derives communication threads out of the structure of relating messages. Messages with a reference to themselves initiate new threads, each of which is assigned a unique thread number. Messages referencing other messages inherit their thread number. The thread number is part of the index number, based on the position in the

resulting thread tree. Threads can contain branched trees as well as chains, or they can be single entries. The index number shows the exact position in the thread tree and is used for calculating measures of structural complexity, e.g., size, depth, and breadth of a thread.

Analyzing Discourse Structures

After building the discourse structure, DSA performs analyses in order to produce metrics and visualizations. The resulting metrics are of three types: metrics describing aspects of the discourse structure, metrics about participants' individual behavior, and metrics about social interaction. These are illustrated in Figure 2 for a prototypical group chat sample.

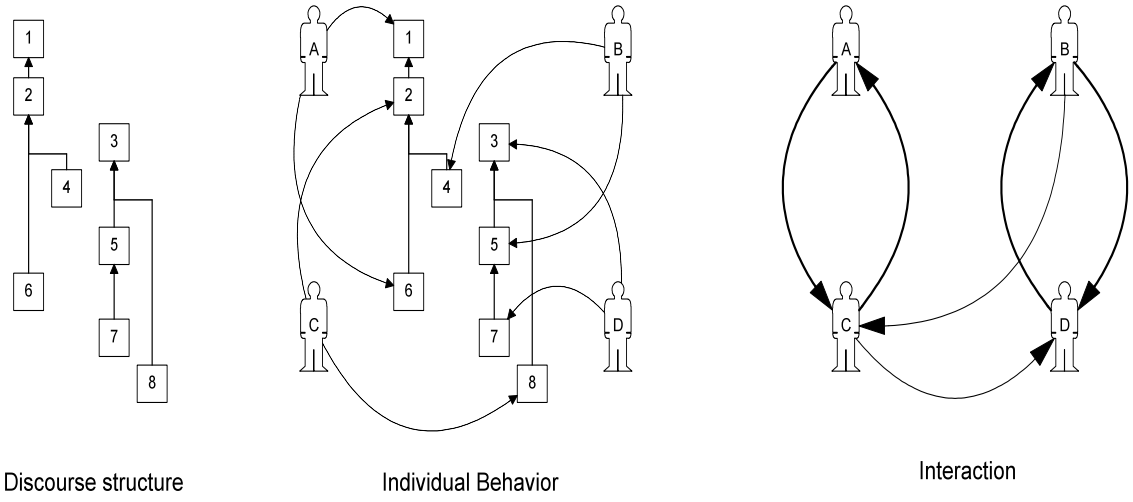


Figure 2. Three groups of DSA metrics: discourse structure (e.g., structure of threads, distance between references), individual behavior (e.g., participation, multitasking behavior) and interaction (e.g., social network properties like density, isolated persons, and sub-groups)

The three groups of metrics in Figure 2 are generated from left (discourse structure) to right (interaction), because each step needs the results of the former step as input for its analysis procedures. Metrics about discourse structure contain information about discourse element types, the size and structure of threads, and overlapping of references and threads.

The discourse element types are *seed* (beginning of a new thread), *chain* (a message with one reply), *fork* (a message with multiple replies), *tail* (a message inside a thread with no further replies), and *isolated* (a message without reference or replies). The proportion of these types can be used to

describe different discourse structures, e.g., a chat with seeds, chains, and tails only vs. a chat with many forks. The first structure contains linear sequences of messages only, which usually suggests discussion between dyads, while the second structure encompasses multiple answers to messages, which suggests discussion in threads with more than two participants. The overlapping of references and threads is analyzed by looking at the messages that are between a message and the message it is referencing. The more messages in between, the greater the distance. The more messages from other threads (which can be detected by comparing the thread numbers), the more threads are running in parallel. In general, these metrics describe the non-linear nature of chat discourse, its concurrency, and its coherence.

The metrics of discourse structure do not take into account that different authors are participating in chat discourse. This is reflected in metrics about individual behavior, which analyze who has created which messages with respect to participation in general and in threads, relative amounts of discourse element types, sender-receiver ratio, participation in interactive dialogue sequences (exchanges between two participants), and multi-tasking (concurrent participation in multiple dialogues). The information shown for each participant is his or her amount and percentage of messages and characters in relation to the whole discussion and to each thread, which allows one to see the pattern of his or her engagement (i.e., broad participation vs. concentration in a few threads). The number of discourse element types is calculated by first detecting the discourse element type of each message and then counting the types for each participant. The distribution of the types shows whether a participant creates more initiating (number of seeds and isolates) messages or more responsive (number of chains, forks, and tails) messages. This is complemented by a count of messages with references (sent) and messages with replies (received) from the current participant.

Interaction metrics describe the relations among interacting participants, e.g., who is talking to whom, how many connections are established in the chat session, and isolated participants and subgroups. These metrics are used to create social networks that can be analyzed by tools and methods of social network analysis, such as Netdraw.[2]

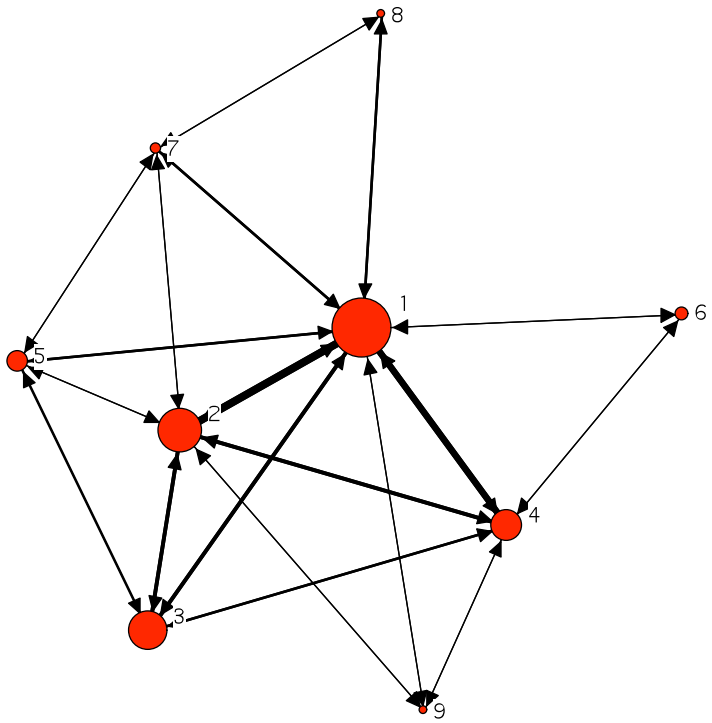


Figure 3. Social network: Network of relationships based on analysis of dialogue sequences of nine users and 329 messages

The network in Figure 3 shows the structure of interaction among nine participants in a 90-minute session that used an unmoderated chat system for a psychotherapeutic Internet chat group (Haug, Strauss, Gallas, & Kordy, 2008), in which the therapist (1) discussed with his patients (2-9) their experiences in the last week. The size of the nodes corresponds to the number of messages produced, and the thickness of the arrows reflects the number of interactive exchanges between participants. This diagram can be used to describe and analyze the structure of the group (e.g., no participant is isolated; participant 1 is the only one who is connected to all other members; not all participants are connected to the same degree; the strongest connections are found inside a small group).

Visualizing Discourse Structures

Beyond the coding–computing–counting approach of the analysis procedures, the ChatLine software allows visualization of the discourse structure and some of its features by means of three different graphical forms. The first one is vertical reference (Figure 4), which was inspired by a visualization in Herring (1999).

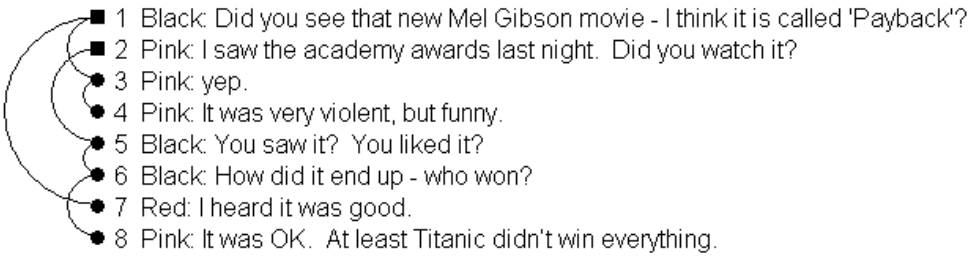


Figure 4. Vertical references using the example from Vronay at al. (1999) with graphical references between messages. Black squares indicate thread starts (seeds); arcs show the distance between referring messages by their span size.

This visualization gives an impression at a glance of the complexity of the discussion. The more overlapping between references and the greater the arc size, the more complex is the discourse. This visualization shows how many messages a participant has to go up with her gaze in order to identify the message the current message is responding to, and illustrates the complexity of reading and understanding the specific chat log. However, the intertwining lines hide important features of the structure that can be visualized by presenting the structure in a two-dimensional thread diagram (Figure 5). The data on which Figure 5 and 6 are based are from a larger chat log (289 messages posted in 90 minutes); for space reasons, the figures show only the first 100 messages. The unmoderated discussion took place between 14 participants in October 1999 and was about e-learning topics.

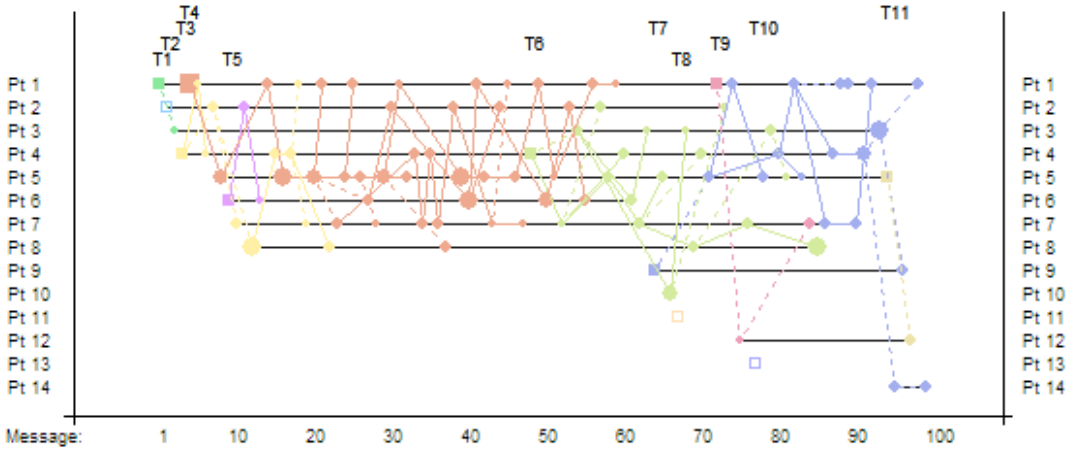


Figure 5. Thread diagram: Messages are represented horizontally from left to right; participants (Pt 1-14) are sorted along the vertical axis; threads (T1-T11), consisting of messages and references, have unique colors; circle size indicates message size.

The thread diagram in Figure 5 allows a quick grasp of the number of active participants, when they were active (the first and last messages are connected by a black line) and how much they wrote (by comparing the number and size of the circles along the horizontal lines). For example, participant 5 seems to write the most and some of the longest messages. The diagram shows how many overlapping references and threads and how many dialogue sequences were created between whom (dialogue sequences are visualized as zigzag patterns with solid lines in contrast to dotted lines). In the chat visualized in Figure 5, participants 1-8 are engaged in multiple dialogues, while participants 9-14 are not engaged in any dialogue. This diagram provides a complex representation of the interactions among users and among threads and displays the dynamics of interaction in a concise and comprehensive manner.

Vertical references and thread diagrams can also be filtered by participant or thread, allowing a more detailed inspection of individual communication behavior. By visually comparing the communication patterns of the participants, the dynamics of communication become easier to grasp (see Figure 6).

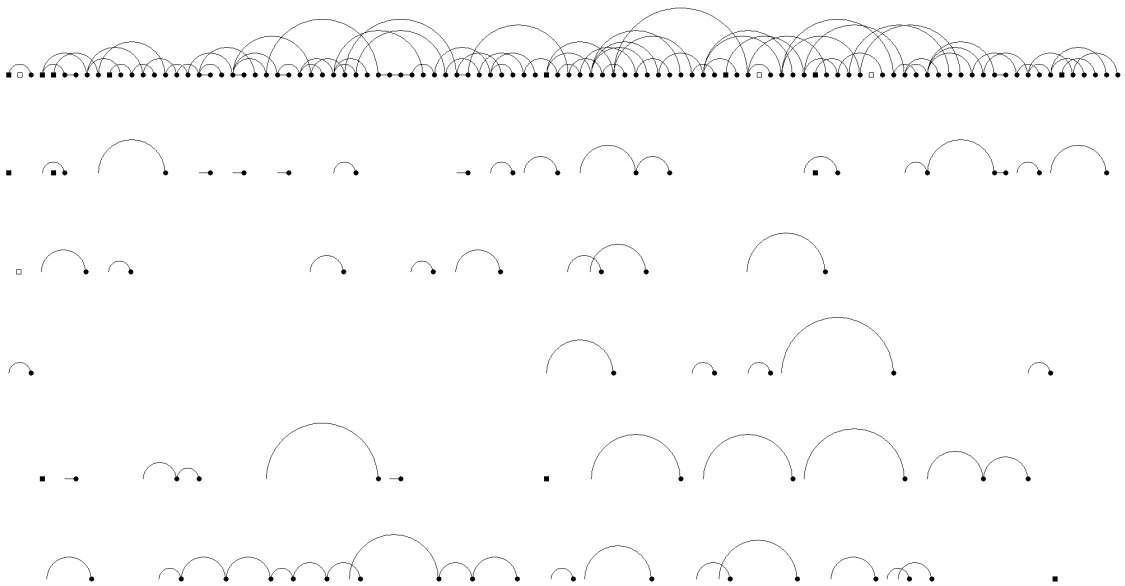


Figure 6. Vertical references filtered by participant and rotated: In the top row, all messages and their references from the chat log are shown in chronological order from left to right; the rows below show the messages and the references produced by participants 1-5 (from top to bottom)

Figure 6 shows how a filtered and adapted visualization presents the dynamics of message production and references. The pattern for participant

1 (represented in the second row) shows that spans of his references are relatively short, which means that his reactions to messages were quite fast. A different pattern is shown by participant 4 (in the fifth row): The spans of his references are quite long, which can be interpreted as a slow response time. Participant 5 (in the sixth row) shows a pattern consisting of two message sequences in which the participant is referring to his own messages. In these phases, the communication behavior resembles a monologue. The above discussion demonstrates that the visualizations can be used in order to detect communication patterns and interesting parts of the discourse, which then can be analyzed further (e.g., why was participant 5 engaging in a monologue?).

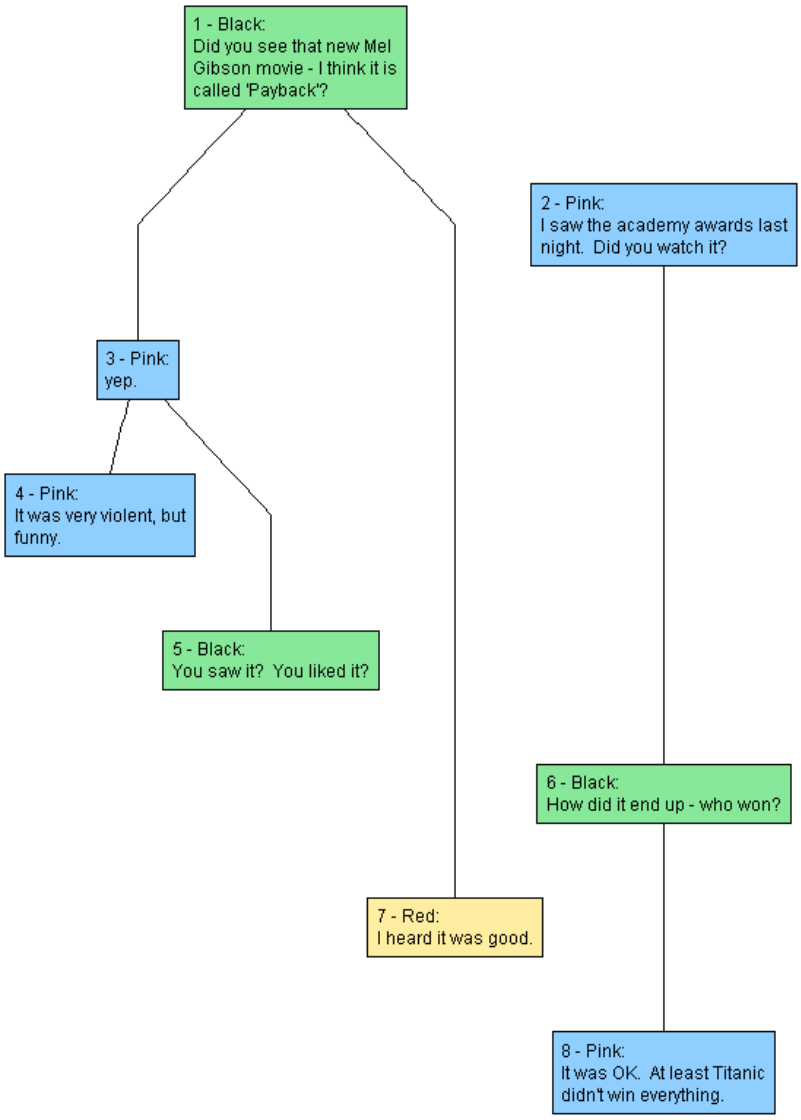


Figure 7. Chat Graph: Message flow is top-down; messages are connected through references; color indicates different users

Another goal of DSA is to create a more appropriate way of representing chat logs than the usual linear list representation. This is done in DSA by representing the discourse structure as a chat graph (see Figure 7). The chat graph representation is designed to make referenced chat logs more readable and to avoid problems of incoherence by separating unrelated messages and threads from one another and by visually connecting and arranging messages that belong together. It may also aid in the process of verifying previously referenced chat logs and in checking the plausibility of references. The visualisation was created by exporting the graph data from *ChatLine* into a format which can be read by uDraw(Graph)-visualization software.[3]

Conclusion and Further Perspectives

In this article, the method of Discourse Structure Analysis was explained and demonstrated. DSA and *ChatLine* were developed in order to support the data-driven analysis of chat logs, with moderate effort required for the manual coding of references. After this necessary phase, all subsequent steps are processed by software that minimizes the required time for quantitative analysis and allows analysts to allocate more time to qualitative aspects. The different visualizations of the discourse structure can help to identify interesting parts of the discourse, which can then be examined in greater depth (e.g., why and how a person is multitasking, or in which phases the discussion contains multiple threads). The functionalities of automatic analysis for groups of chat logs enable investigation and comparison across larger data sets. A topic for future research is the usefulness of these visualizations (especially the chat graph) for understanding the content of a chat log. In the future, I am planning experimental studies in which different forms of discourse visualizations are compared with respect to user understanding of complex and multi-threaded discussions.

Although DSA was developed for the analysis of chat logs, the metrics and visualizations can be applied in part to newsgroup and forum discussions, as well. The structural element of references is more explicit in such discussions and in most cases does not have to be interpreted and coded manually, because users do this with the reply function (exceptions are when users hit "reply" but change the topic, or when users start a new response accidently without hitting "reply"). Messages in forums and newsgroups are linked by the users; this information can be used directly to analyze intermessage coherence. Some metrics (e.g., number of parallel threads) have the same meaning, but others differ significantly in the way they must be interpreted. For example, multitasking in chat means that a user is simultaneously

engaged in multiple discussions and has only limited time for switching between threads. In an asynchronous medium like a newsgroup, being engaged in multiple threads is easier and occurs more often because the time available for reading and responding can last hours instead of seconds. The coherence is different, because the asynchronous environment better supports the following of threads than does the typical chat log.

Another application domain of DSA and *ChatLine* are logs from chats that allow active referencing by the users themselves. Systems such as *KOLUMBUS*, developed by the University of Dortmund (Holmer, Kienle, & Wessner, 2006) and *ConcertChat*, developed by Fraunhofer IPSI (Mühlpfordt & Wessner, 2005), support the creation of reference information by users during communication. The resulting chat logs already contain the reference information and can be analyzed immediately, without having to reference the log through manual coding. This makes the application of DSA to these kinds of logs very easy and increases its usability.

The problem of manual coding could be overcome though automatic referencing. For this purpose, the rules for detecting references would have to be formalized and implemented. Although first steps in this direction have been taken by researchers in artificial intelligence and computational linguistics (e.g., Mutton, 2004), the quality thus far is too poor to be of help to manual coders. With more experience in the method of referencing and better understanding of the dynamics of chat communication, however, it should be possible for researchers to create tools that provide valuable assistance. Accordingly, I am working on a collection of referenced chat logs that can serve as a basis for comparison between automated and manually referenced chat logs, in order to improve the automatization of referencing.

Notes

1. The *ChatLine* software was developed by the author and can be ordered by email. *ChatLine* requires Windows OS (98, 2000, XP, Vista) and is free of charge for non-commercial uses.
2. NetDraw: Graph Visualization Software, Analytic Technologies, Harvard. http://www.analytictech.com/Netdraw/netdraw.htm
3. uDraw(Graph): Graph visualization software, University of Bremen, Germany http://www.informatik.uni-bremen.de/uDrawGraph/

References

Black, S., Levin, J., Mehan, H., & Quinn, C. (1983). Real and nonreal time interaction: Unraveling multiple threads of discourse. *Discourse Processes, 6*(1), 59-75.

Cornelius, C., & Boos, M. (2003). Enhancing mutual understanding in synchronous computer-mediated communication by training: Trade-offs in judgmental tasks. *Communication Research, 30*(2), 147-177.

Egbert, M. (1997). Schisming: The collaborative transformation from a single conversation to multiple conversations. *Research on Language & Social Interaction, 30*(1), 1-51.

Garcia, A. C., & Jacobs, J. B. (1998). The interactional organization of computer mediated communication in the college classroom. *Qualitative Sociology, 21*(3), 299-317.

Garcia, A. C., & Jacobs, J. B. (1999). The eyes of the beholder: Understanding the turn-taking system in quasi-synchronous computer-mediated communication. *Research on Language and Social Interaction, 32*(4), 337-367.

Gerosa, M., Pimentel, M., Fuks, H., & Lucena, C. (2004). Analyzing discourse structure to coordinate educational forums. *The 7th International Conference on Intelligent Tutoring Systems - ITS-2004* (pp. 262-272). Berlin & Heidelberg: Springer.

Gerosa, M., Pimentel, M., Fuks, H., & Lucena, C. (2005). No need to read messages right now: Helping mediators to steer educational forums using statistical and visual information. *Proceedings of the Computer Supported Collaborative Learning Conference – CSCL 2005, 01-04 June, Taipei, Taiwan* (pp. 160-169). International Society of the Learning Sciences.

Hara, N., Bonk, C. J., & Angeli, C., (2000). Content analysis of online discussion in an applied educational psychology course. *Instructional Science*, *28*(2), 115-152.

Haug, S., Strauss, B., Gallas, C., & Kordy, H. (2008). New prospects for process research in group therapy: Text-based process variables in psychotherapeutic internet chat groups. Psychotherapy Research , 18 (1), 88-96.

Herring, S. C. (1999). Interactional coherence in CMC. *Journal of Computer-Mediated Communication, 4*(4). Retrieved August 22, 2008 from http://jcmc.indiana.edu/vol4/issue4/herring.html

Herring, S. C. (2003). Dynamic topic analysis of synchronous chat. In: New Research for New Media: Innovative Research Methodologies Symposium Working Papers and Readings. Minneapolis, MN: University of Minnesota

School of Journalism and Mass Communication. Retrieved November 28, 2009 from http://ella.slis.indiana.edu/~herring/dta.2003.pdf

Herring, S. C. (2004). Computer-mediated discourse analysis: An approach to researching online communities. In S. A. Barab, R. Kling, & J. H. Gray (Eds.), *Designing for virtual communities in the service of learning* (pp. 338-376). New York: Cambridge University Press.

Herring, S. C., & Kurtz, A. J. (2006). Visualizing dynamic topic analysis. *Proceedings of CHI'06*. New York: ACM Press.

Holmer, T., Kienle, A., & Wessner, M. (2006). Explicit referencing in learning chats: Needs and acceptance. In W. Nejdl & K. Tochtermann (Eds.), *Innovative approaches for learning and knowledge sharing* (pp. 170-184). Berlin: Springer.

McDaniel, S., Olson, G., & Magee, J. (1996). Identifying and analyzing multiple threads in computer-mediated and face-to-face conversations. In M. Ackerman (Ed.), *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work* (pp. 39-47). Boston, MA: ACM Press.

Mühlpfordt, M., & Wessner, M. (2005). Explicit referencing in chat supports collaborative learning. *Proceedings of the 2005 Conference on Computer Support for Collaborative Learning: Learning 2005: the Next 10 Years! (Taipei, Taiwan, May 30 - June 4, 2005)* (pp. 460-469). International Society of the Learning Sciences.

Mutton, P. (2004). Inferring and visualizing social networks on Internet Relay Chat. *Proceedings of the Eighth International Conference on Information Visualisation (IV'04)* (pp. 35-43). Washington, DC: IEEE Computer Society.

Rafaeli, S., & Sudweeks, F. (1997). Networked interactivity. *Journal of Computer-Mediated Communication*, *2*(4). Retrieved August 22, 2008 from http://jcmc.indiana.edu/vol2/issue4/rafaeli.sudweeks.html

Schegloff, E., & Sacks, H. (1973). Opening up closings. *Semiotica, 8*(4), 289-327.

Shi, S., Mishra, P., Bonk, C. J., Tan, S., & Zhao, Y. (2006). Thread theory: A framework applied to content analysis of synchronous computer mediated communication data. *International Journal of Instructional Technology and Distance Learning*, *3*(3), 19-38.

Vronay, D., Smith, M., & Drucker, S. (1999). Alternative interfaces for chat. *UIST '99: Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology* (pp. 19-26). New York: ACM Press.

Biographical Note

Torsten Holmer [torsten.holmer@web.de] is a professor for knowledge media in the School of Informatics, Communications, and Media at the Upper Austria

University of Applied Sciences, Hagenberg, Austria. His research interests include human-computer interface design, collaborative working and learning, and computer-mediated communication and discourse.