

SolutionChat: Real-time Moderator Support for Chat-based Structured Discussion

Sung-Chul Lee¹, Jaeyoon Song², Eun-Young Ko¹, Seongho Park¹, Jihee Kim¹, Juho Kim¹

¹KAIST, Daejeon, Republic of Korea

²Seoul National University, Seoul, Republic of Korea

{leesungchul, eunyoungko, coloroflight, jiheekim, juhokim}@kaist.ac.kr song@jaeyoon.io

ABSTRACT

Online chat is an emerging channel for discussing community problems. It is common practice for communities to assign dedicated moderators to maintain a structured discussion and enhance the problem-solving experience. However, due to the synchronous nature of online chat, moderators face a high managerial overhead in tasks like discussion stage management, opinion summarization, and consensus-building support. To assist moderators with facilitating a structured discussion for community problem-solving, we introduce SolutionChat, a system that (1) visualizes discussion stages and featured opinions and (2) recommends contextually appropriate moderator messages. Results from a controlled lab study ($n=55$, 12 groups) suggest that participants' perceived discussion trackability was significantly higher with SolutionChat than without. Also, moderators provided better summarization with less effort and better managerial support using system-generated messages with SolutionChat than without. With SolutionChat, we envision untrained moderators to effectively facilitate chat-based discussions of important community matters.

Author Keywords

Online Discussion; Structured Discussion; Computer Mediated Communication; Moderator Support; Chat Interface

CCS Concepts

•Human-centered computing → Interactive systems and tools;

INTRODUCTION

More communities today are turning to online chat as a channel for discussing community matters and making decisions. Examples include: Comcast's employees organized a rally on Slack, a group messaging platform, which started the #Tech-HasNoWalls movement [28]; Strong Towns, a non-profit organization for promoting discussion about building financially resilient communities, uses Slack for discussion with dedicated moderators [36]; AOL (America Online) chatrooms in 1990s were used to discuss political issues [19].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '20, April 25–30, 2020, Honolulu, HI, USA

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-6708-0/20/04...\$15.00

DOI: <https://doi.org/10.1145/3313831.3376609>

The high accessibility and familiarity of online chat has led to its adoption for community discussion, as more members can easily join in a discussion without geographical constraints. Online chat is also socially engaging, and research shows that the dynamics of online chat discussion are similar to those in face-to-face discussions [3]. The synchronous nature of online chat, however, may feel too fast and chaotic [16], making it difficult for participants to keep track of discussion and follow up a missed conversation, thereby limiting its effectiveness in discussing important community matters.

Providing a predefined discussion structure can make an online chat discussion more productive and efficient. *Structured discussion* can help participants focus on the agenda while reducing topic changes and irrelevant messages [14], and scaffold the problem-solving process while compensating for participants' limited knowledge [45]. Communities practicing structured discussion often introduce predefined stages and voting mechanisms to scope their discussion. To help manage and facilitate structured community discussions, many communities assign moderators: they guide participants' focus [22] and direct the problem-solving process [44]. While online platforms like Loomio [4], LiquidFeedback [10], and Mood [43] have been designed for structured asynchronous discussions with moderation support, little research has applied structured discussion to synchronous online chat.

To understand the challenges of supporting structured discussion in online chat, we conducted a formative study. First, participants had limited awareness of the discussion structure and often relied on the moderator to remind them. Second, moderators reported a high burden in managing the discussion in the midst of fast chat flow and high volume of messages. Moderators had to coordinate various tasks in real-time to smoothly run the discussion. Their tasks included introducing the current stage, summarizing the main points raised by participants, and asking for supporting evidence, among others. Many moderation messages were recurring across stages.

To address the issues of limited structure awareness and moderator support in chat-based online discussion, we introduce SolutionChat. SolutionChat is a web-based chat interface that (1) visualizes the discussion structure and featured opinions (FO) for each stage and (2) recommends contextually appropriate moderation messages to moderators (Figure 1).

To improve structure awareness for discussion participants, SolutionChat provides an agenda panel (AP) (Figure 1A), which visualizes the overall discussion structure and highlights the current discussion stage (Figure 1B). The moderator can

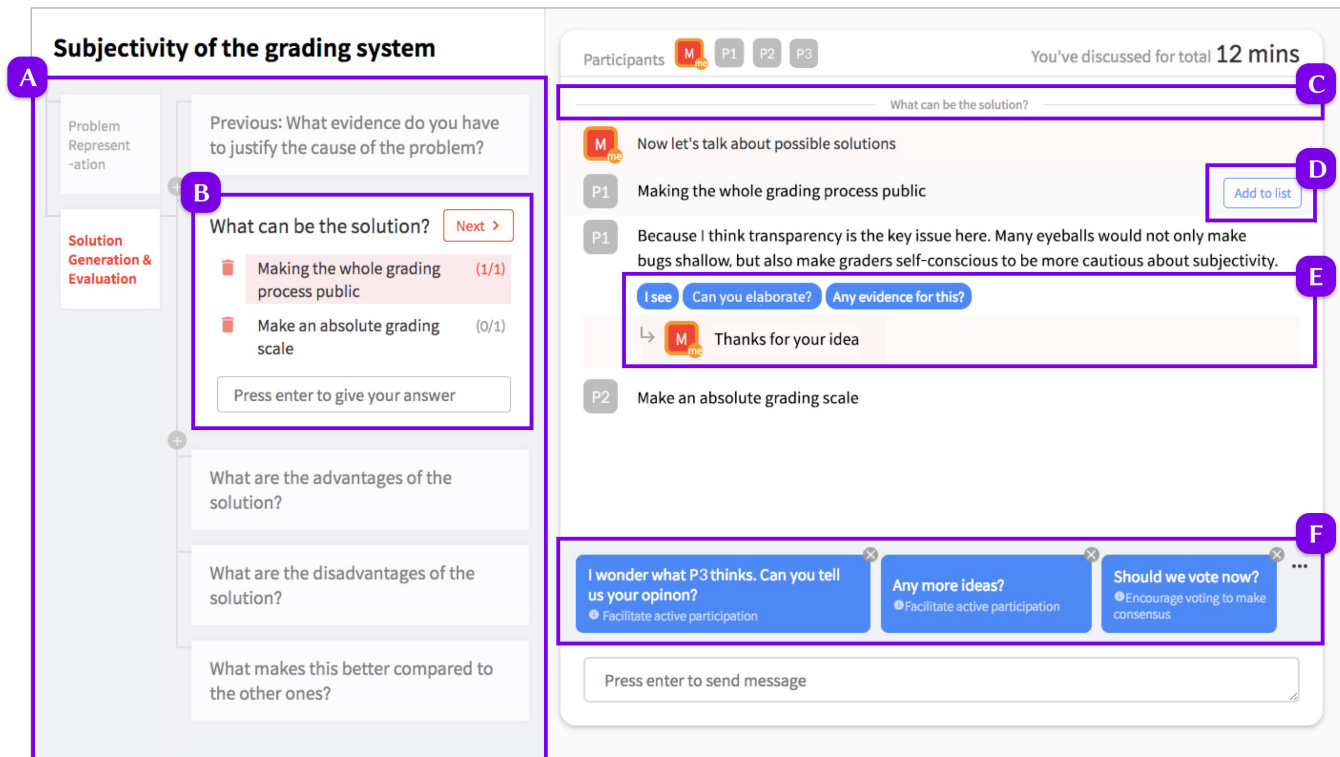


Figure 1. The overview of SolutionChat: (A) agenda panel for showing the discussion structure, (B) current discussion stage and featured opinions, (C) stage divider, (D) button for add a message as a featured opinion, (E) inline message recommendations for moderators to add short reactions to discussants' opinions, and (F) block message recommendations with generic facilitation messages for moderators to use.

decide to move the discussion to the next or any stage, and stage updates are synchronized between all participants.

Participants can label any chat message as a featured opinion (Figure 1D), which then gets listed in the AP and serves as a summary of main ideas discussed for each stage. For the remainder of the paper, we refer marking a message as a featured opinion as *summarization*.

SolutionChat also provides real-time message recommendations (MR) to moderators to lower their task load (Figure 1E and 1F). SolutionChat monitors the discussion context in real-time, including chat messages, current stage, featured opinions, and elapsed time for the current stage, and recommends appropriate facilitation messages in real-time (e.g., “Are there any other opinions?” and “Can you elaborate further?”). The moderator can simply click a recommended message to send to all participants.

To evaluate the effect of SolutionChat on supporting structured discussion in online chat, we ran a controlled lab study with 12 groups of university students ($n=55$). Three conditions compared were the baseline chat interface, AP, and AP with MR (AP+MR). Results show that participants with AP report a higher level of awareness on the discussion structure. Moderators who used AP sent fewer summarization messages than the baseline while participants were able to track main ideas better. Moderators with AP+MR sent significantly more managerial messages compared to the other conditions.

This paper makes the following contributions:

- Design goals derived from the formative study that identify ways to support structured discussion in online chat
- SolutionChat, an online chat system for structured discussion designed to improve structure awareness and moderator support
- Results from the controlled study showing the value of the agenda panel and real-time message recommendations in supporting structured discussion

RELATED WORK

In this section, we review previous work on the role of moderators in discussion, moderation in online communities, structured community participation, discussion summarization, and real-time message recommendation.

The Role of Moderators

A rich body of education research has focused on improving the quality of students' discussion, as it is shown to be linked to students' learning gain [3]. These studies highlight the importance of students' engagement in turn-taking of conversations [7], production and development of students' ideas [23, 42], and argument-oriented discussions [1, 2, 6, 11, 37].

An effective solution for moderators to promote quality discussion and stimulate students is to use generic prompts [3], such as questions like “Can you add something here?” and “Can you provide any evidence for it?”. A limitation of this

intervention technique is that it requires adequate training for moderators prior to the discussion.

Moderators coordinate the discussion in various ways while managing discussion structure. Lund [30] suggests four types of discussion support for human argumentation: pedagogical support to provide educational interventions (e.g., generic prompts and summarization), social support to maintain a bright and safe atmosphere, interaction support to react to participation activity, and managerial support to cover task design, task completion, monitoring, and technical support.

When discussing issues in the community, moderators should provide managerial support with discussion structure in mind and pedagogical support to lead participants to have productive conversations. Since social and interaction support are closely associated with group dynamics of a community and relationships between its members, these two types of support are relatively less connected to the structure of discussion.

Moderation in online communities

Online communities often moderate content and user behavior to preserve and strengthen their organizational norm and processes. For content moderation, algorithms are increasingly used to fully automate the moderation process [21] or collaborate with humans [20]. For process moderation, systems can manage group processes. For example, a chatbot guided repetitive discussion stages via participants' command or voting [41]. However, algorithmic moderation in response to the dynamics of group discussion remains a challenge due to limited NLP performance [25] and a high cost of failed interactions. Inspired by previous studies that have demonstrated the usefulness of meta-talk prompts generated by newcomers [31] and the value of generic prompts [3], this work suggests a way to assist untrained moderators in the context of dynamic online chat discussion by enabling algorithmic suggestions with human control.

Structured Community Participation

While the form of structure varies, many community participation platforms have chosen asynchronous communication to run their structured discussion.

A common design pattern used by discussion platforms is to introduce multiple stages. Multiple stages drive participants' discussion effort to evaluate various predefined aspects of proposals. Mood [43] is a solution-centered discussion platform with four stages (problem proposal, solution proposal, feasibility, and moral check). LiquidFeedback [10] is a platform for discussing policy with four stages (admission, discussion, verification, and vote).

In synchronous discussion, however, there are fewer cases of supporting structured participation than in asynchronous settings. Leadline [14] supports structured discussion by providing a pre-selected script with structure information. Loops [13] scaffolds semi-structured discussion with five major stages (set goals, generate ideas, elaborate ideas, critique and rank ideas, and archive the results) and creates a sub-room for each stage for users to freely join.

While inspired by these systems, SolutionChat differs in that it provides more flexible structure where the moderator can use customized structure and has control in stage transitions.

Discussion Summarization and Real-time Message Recommendation

Discussion summarization is an important area of support for online discussions, as it provides an accessible overview to participants to gain shared awareness and understanding of the progress of discussion.

Previous work has introduced summarization techniques for discussion using algorithms [15, 48, 35] or crowdsourcing. Wikum [47] proposes a multi-level and recursive summarization workflow. Deliberatorium [24] demonstrates the usefulness of summarization in the form of a tree-structured network. For online chat, Collabot [38] suggests a way to cluster online chat messages based on topics, and Tilda [46] allows participants to markup chat messages for summarization. Other systems have focused on summarizing the consensus state to help communities' decision-making process. ConsensusUs [29] visualizes participants' consensus for multi-criteria decision making, and ConsiderIt [26] visualizes the level of agreement between users for a policy.

Real-time messaging recommendation can be helpful to reduce a load of communication. Many commercial mobile keyboard applications recommend appropriate emojis based on the typed words [17, 39, 32, 40]. Gmail's Smart Reply [18] suggests appropriate short responses to email and Gmail's Smart Compose [5] suggests words and phrases as the user types in email text. These techniques attempt to capture the communication context to make recommendations.

To the best of our knowledge, SolutionChat is the first moderator support system for online chat discussion to combine summarization and real-time messaging recommendation.

FORMATIVE STUDY

To better understand the challenges in managing and participating in chat-based structured discussion, we conducted a formative study. We sought to (1) understand how participants perceive and track the discussion structure and (2) investigate the role and behavioral patterns of moderators, especially with respect to Lund's four types of human support [30] in a structured online chat discussion.

Participants

We recruited 18 participants (Male: 9, Female 9), all of whom were students at our institution in South Korea. We divided 18 participants into six groups of three. Each participant received 15K KRW (~12.5 USD) for joining an hour-long study session.

For the remainder of the paper, we refer to discussion participants who are not a moderator as *discussants*. This is to avoid confusion with study participants, which include both moderators and non-moderators.

Tasks and Procedures

We asked participants to discuss the issue of requiring faculty advisors' physical signatures for various administrative pro-

cesses at our institution. This issue was heavily discussed in online communities at our institution at the time of the study.

Considering community problems are often ill-structured (i.e., problems have no one optimal answer) and the problem could lead to many possible solutions [8], we prepared a discussion structure by adopting the ill-structured problem-solving process proposed by Ge [45] (Problem representation, Generating or selecting solutions, Making justifications, Monitoring and evaluating). The final discussion structure included cause, evidence for the cause, solution, and advantages and disadvantages of the solution. Participants were asked to follow this structure in their discussion.

We designed three experimental conditions to compare discussion activity qualitatively in terms of moderator presence and predefined discussion structure: moderator+structure, moderator-only, and structure-only. For groups in the moderator+structure condition, we randomly designated a moderator and presented the discussion structure to the moderator (but not discussants). For groups in the moderator-only condition, we also randomly designated a moderator but nobody in the group was presented with the discussion structure. For groups in the structure-only condition, we did not designate a moderator but presented the discussion structure to all discussants. We prepared a web-based chat interface with a message log, a message input field, and a user ID display with a special marker for the moderator. Participants were instructed to discuss via the chat interface for 40 minutes and share their experience in a group interview session.

Results & Design Goals

We organize results by extracting contrasts that may be derived by the presences of the structure and moderator.

Discussion Structure: Groups with discussion structure (structure-only and moderator+structure) generally followed the structure and discussed ideas within the scope of each stage. Structure-only groups sometimes skipped certain discussion stages without clear consensus or made frequent shifts between the stages (e.g., discuss solutions in the cause stage), perhaps due to having no moderator. Moderator-only groups, while not having predefined structure, developed their own custom structure during discussion. One group mainly discussed the pros and cons of the current practice and the other group came up with a two-stage workflow (cause and solution).

Groups with the discussion structure covered various aspects that the structure suggested, while groups without the structure tended to focus on fewer aspects and held longer conversations on them. Discussants had mixed reactions to discussing with structure. A discussant in the moderator+structure condition expressed that “Somebody should ask these kinds of questions [U1]” and “The moderator tightly managed the discussion with the questions. I think it was good. For example, the first problem alone may take whole discussion time, however, the moderator made a transition fluently between the stages [U2].” However, U2 also expressed that they experienced time pressure during chat: “I felt I have been dragged by the moderator”. Also, U2 complained it was difficult to know what structure the group was following, as they had no explicit

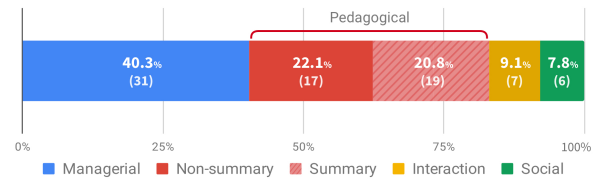


Figure 2. A distribution of moderation message counts per support type. Moderators mainly provided pedagogical and managerial support to discussants during chat-based discussion.

access to it: “I had hard time knowing where are we in the discussion because we don’t know about the question. [U2]”

Based on our findings related to discussion structure, we identify the following design goal:

- G1. Assist discussion stage management by exposing the discussion structure and highlighting the current stage to all participants.

Moderator Actions: To understand the types of support moderators provided to discussants, we labeled and counted moderator messages based on Lund’s human support categories [30] for the groups with moderator+structure and moderator-only. Previous work also used this scheme to categorize moderators messages in chat [3]

We conducted a discourse analysis to label moderators’ messages with their intent. Two researchers independently assigned a code to each message by developing their own coding scheme, and resolved conflicts through discussion. Based on the intents we distilled, we matched each intent to one of Lund’s four support categories (See Table 1).

The managerial support and pedagogical support take the major share of moderator messages (see Figure 2 and Table 1). Managerial messages were often associated with messages that repeated from stage to stage. For example, messages like “Shall we proceed to the next stage?” were repetitively observed across the stages. Among all managerial message types, the *stage introduction* message, in which the moderator reminds the goal of the discussion stage upon entering a new stage, was the most frequent.

In terms of pedagogical messages, moderators summarized the points raised by discussants, asked discussants to provide evidence or elaborate their points, and redefined the scope of discussion. Among all pedagogical message types, the *summarization* message was most common. In the interview, discussants expressed gratitude for the moderator’s summarization efforts while at the same time they were worried about the excessive burden of the moderator. They expressed needs for systemic support for discussion summarization. Similar to managerial messages, many generic pedagogical support messages occurred repetitively across the stages (e.g., “Can you elaborate more?”).

Since the summarization messages are specific to the content of discussants’ message while the other pedagogical messages are more generic (without requiring specific discussion content to be included in the message), we separated pedagogical

<i>Human support category</i>	<i>Moderator's intent</i>	<i>Example message</i>
Managerial	Introduce a new stage	"This time, let's talk about the pros of the solution."
	Proceed to the next stage	"Shall we go to the next stage?"
	Manage time	"Due to time constraints ..."
	Suggest voting	"Should we vote among our current candidates?"
Pedagogical	Summarize	"There was an opinion that taking a leave of absence for personal reasons should not require an advisor signature."
	Ask for elaboration	"Can you elaborate more?"
	Ask for evidence	"Can you give us an example?"
	Redefine discussion scope	"Let's talk about taking a leave of absence for personal reasons."
Interaction	Request participation	"U2, would you like to share your opinion?"
	Request new ideas	"Any ideas?"
Social	Acknowledge	"Thank you for your thoughtful opinions."

Table 1. Categories of moderator support and types of moderators' intent for each category. We developed a coding scheme to identify frequently used message intents and mapped them to Lund's human support categories [30]

messages into summary pedagogical support and generic pedagogical support, and set design goals for each (see below).

Based on our findings related to moderator actions, we identify the following design goals:

- G2. Reduce moderators' constant burden in summarizing throughout the discussion.
- G3. Facilitate moderators' managerial support by assisting with repetitive managerial messages.
- G4. Facilitate moderators' pedagogical support by assisting with repetitive pedagogical messages.

SOLUTIONCHAT

With these design goals in mind, we introduce SolutionChat, an online chat platform designed to support structured discussion with moderator assistance. With SolutionChat, we envision untrained moderators can apply structural discussion with a lower overhead while maintaining active managerial and pedagogical support, thereby contribute to supporting structured discussion to a broader range of groups and communities to engage in chat-based discussions.

The SolutionChat interface (Figure 1) consists of three main components. The agenda panel (AP) (Figure 1A) is designed to increase participants' structure awareness and decrease the summarization burden of moderators. AP displays the structure of the discussion and allows participants to collectively maintain featured opinions for the current stage. Message recommendations (MR) (Figure 1E & F) suggest contextually appropriate messages moderators can use in real-time. The chatroom displays messages with participants' ID. The moderator ID is denoted as M and highlighted in red for improved visibility. When the discussion stage changes, a stage divider is added to visually alert the start of a new stage to participants.

Moderators' SolutionChat Usage Scenario

We walk through a usage scenario for SolutionChat from a moderator's view. Once a group decides on a discussion topic, they can create a SolutionChat chatroom dedicated for the topic and choose an initial discussion structure to follow,

which could be either the default template provided by the system or any custom list of stages. The selected structure populates the AP. Once the chatroom is created, it gets a shareable URL and participants can join the room through the link.

The discussion starts from the first stage of the structure, and the moderator has the control of the flow: they can determine when to advance to a different stage and which stage to move to. Upon starting a new stage, the moderator receives message recommendations from MR to introduce the stage (e.g., "Let's talk about the pros of the solution now.", "Please don't mention the cons of the solution for now as we'll discuss them later."). The moderator can click a recommended message to send it in chat. At the same time, discussants can clearly notice the stage transition by the stage divider in the chatroom window (Figure 1C) and the highlighted current stage in the AP (Figure 1B). Now MR recommends messages to promote brainstorming (e.g., "Any more ideas?") and developing arguments (e.g., "Can you give an example?"). The moderator and discussants can freely label any message in the chat as a featured opinion by clicking on the 'Add to list' button next to a message (Figure 1D), which then lists the message in the AP under the current stage (Figure 1B). Once discussants suggest more than three featured opinions or exceed the recommended stage allocation time (both of which are customizable), SolutionChat recommends messages about voting or reaching a consensus. The optional voting feature allows the group to pick a winning opinion or an idea for the stage, which helps the discussion in the following stages to be more focused. If the voting finishes with a winner or the moderator manually clicks the Next button (Figure 1B), the current stage is marked as complete and the discussion moves on to the next stage.

Structure Awareness Support: Agenda Panel

To address G1 (Assist discussion stage management), SolutionChat provides a generic discussion structure inspired by the ill-structured problem-solving process [45] as a default discussion structure template. We assume most community discussions address ill-structured problems because social problems are often ill-structured with no clear solution [45].

<i>MR</i>	<i>Type</i>	<i>Condition</i>	<i>Recommended Message Example</i>	<i>References</i>
Inline	Social	NLU (opinion)	“I see.”, “Thank you for your opinion.”	[45, 34]
	Pedagogical	NLU (opinion)	“Can you provide some evidence?” “Can you elaborate?”	[45, 34]
Block	Managerial	Discussion stage change	“In this stage, we will set the goal of the discussion.”	[33]
		Every three minutes	“We have X minutes left for our discussion.”	[27]
			“Let’s talk about this for X more minutes.”	
			“Can we move faster since we are running out of time?”	
	Pedagogical	A majority vote on a FO	“Shall we proceed to the next stage?”	[34]
		3 min. after the last FO	“Shall we vote now?”	[34]
	Pedagogical	3 min. after the last FO	“Are there any other opinions?”	[34]
		FO added	“Can we try to find evidence for the featured opinions?”	[45]
			“Are there any other opinions?”	
		FO count > 3	“Let’s check if our featured opinions are biased.”	[34]
	Interaction	Every three minutes	“I wonder what XX thinks. Can you tell us your opinion?”	[34]

Table 2. The recommendation messages of Inline MR and Block MR. Inline MR recommends social and pedagogical messages for discussant’s opinion messages and Block MR recommends primarily managerial and pedagogical messages based on the information of AP.

The discussion structure starts with a goal statement (“Define a goal state in one sentence for this problem”), followed by causes of the problem, evidence of the problem, solution ideas, pros and cons of solutions, and reasons why the selected solution is the best solution. To accommodate more flexible discussion structure for diverse discussion scenarios, the moderator can use a custom structure that suits the need of the group. Once the moderator selects a structure template and as the structure gets updated, all participants can see the entire discussion stages on the left side of the screen (Figure 1A).

To address G2 (Reduce the burden of summarization), SolutionChat helps participants recognize the current stage of discussion. It highlights the current stage in the AP (Figure 1B) and adds a stage divider in the chat window whenever the stage changes (Figure 1C). The moderator can flexibly decide which stage to direct the discussion to, by proceeding to the next stage, jumping to any stage in the structure, or creating a new stage. The moderator can proceed to the next stage by pressing the Next button in the AP, which could be useful when the moderator wants to follow the given structure upon finishing up a discussion in the current stage. The moderator can jump to a specific stage at any time by clicking on a stage in AP, which could be useful if the moderator wants to skip certain stages or go back to a previous stage of the discussion. The moderator can create a new stage in any location of the structure by clicking the plus button in the AP, which could be useful if the need for a new stage arises during a chat.

To address G4 (Facilitate pedagogical support), SolutionChat allows participants to add any opinion as a featured opinion and vote among the featured opinions. By listing the featured opinions for the current stage, the AP serves as a collectively maintained summary and an archive of the discussion. Participants can keep track of main ideas discussed and easily follow up with the discussion even if they missed some parts of it. All in all, this allows everyone in the group to have a shared

understanding of the status of the discussion. The moderator can edit the list to keep it organized and up-to-date.

Moderator Action Support: Message Recommendations

MR is designed to reduce the cost of moderator’s discussion facilitation efforts. To address G3 and G4 (Facilitate managerial support), MR recommends various managerial and pedagogical support messages in real-time.

SolutionChat displays recommendation messages in two different areas depending on whether the recommendation can be linked to a specific discussant’s message. If the recommended messages can be linked to a specific message, they are displayed inline right below the discussant’s message (Inline MR, Figure 1E). For more general recommendations that are not specific to a discussant’s message, they are displayed right above the chat input box (Block MR, Figure 1F).

To determine which types of recommendation messages might be helpful for structured discussion, we did a literature survey in organizational behavior and CSCL (Computer Supported Collaborative Learning): Organizational behavior literature suggests interventions for productive discussions and CSCL literature suggests interventions for stimulating the reasoning process. The resulting categories along with relevant references are presented in Table 2.

With **Inline MR**, the moderator can add short reactions to discussants’ opinions. To promote socially safe atmosphere for freely sharing opinions (social support) and stimulate discussants to develop arguments, Inline MR recognizes whether discussants’ messages are opinions, and if so, recommends appropriate social (e.g., “I see.” and “Thank you for your idea.”) and pedagogical messages (e.g., “Can you elaborate?” and “Any evidence for this?”).

To detect discussants’ opinions in real-time, we use natural language understanding technology. We trained a Snips NLU module [9] for this purpose, and used two sources of training

data: (1) chat data from the formative study and (2) manually crafted phrase templates. For (1), a discussant's utterance "I don't think it's going to work very well." is labeled with the intent of "adding an opinion", as the discussant is expressing their opinion. A possible reply for this utterance could be "Can you elaborate?". For (2), we created templates that commonly appear in opinion messages (e.g., "I think the problem is ", "We need ", "it seems "). We configured Snips NLU's deterministic intent parser (regular expression based) as our primary parser for the training data and used probabilistic intent parser (logistic regression based) as a fallback. Due to the small size of training data, the accuracy of detection definitely has large room for improvement. While we leave improving the accuracy and training the NLU unit with larger training data as future work, we also note that our design choice of allowing the moderator to ignore recommendations mitigates the issue caused by low detection accuracy.

With **Block MR**, the moderator can facilitate the overall discussion more efficiently with the help of recommended messages for managing discussion stages. Block MR recommends two primary support types: managerial messages related to the task setting activity for the structure (e.g., "Shall we proceed to the next stage?") and pedagogical messages on featured opinions (e.g., "Let's check if our featured opinions are biased"). The full list of recommendation message categories are presented in Table 2.

MR continuously monitors the AP because it represents up-to-date information about the discussion. MR recommends managerial messages based on the stage information of AP. For example, when the stage changes, MR can pick stage introduction messages (e.g., "Let's talk about the con of the solution") that are specifically designed for the current stage. MR can also recommend pedagogical messages based on the information present in AP. For example, once a certain number of featured opinions are added, MR recommends a bias checking message (e.g., "Let's check if our featured opinions are biased.") to prevent group thinking.

The moderator can click a recommended message to send it to all discussants or manually enter any message they want. When pilot-testing Block MR, we noticed that many moderators were inspired by recommended messages but rather than clicking it chose to paraphrase and add their own flavor to write a new message. To support this implicit use of recommendations further, Block MR automatically dismisses any recommendation that is similar in content to the message manually entered by the moderator. We trained an NLU unit with canned messages of MR to check for the similarity.

EVALUATION

To evaluate the effect of AP and MR on supporting structured discussion in online chat, we ran a controlled lab study. The primary objective of the lab study is to see how AP affects participants' awareness of discussion structure and how MR affects moderators' facilitation activity. Specifically, we test the following hypotheses.

- H1. Participants hold higher awareness of the discussion structure with AP.

- H2. Moderators provide better summarization support with fewer summarization messages with AP.
- H3. Moderators provide better managerial support with more managerial messages with MR.
- H4. Moderators provide better pedagogical support with more pedagogical messages with MR.

Participants

We recruited 55 participants from two Korean universities. We posted participant recruitment flyers on online communities of each university. All participants agreed with up to two hours of participation in the experiment for 20K KRW (~18 USD). All participants were informed of the discussion topics prior to the experiment, and the recruitment explicitly stated that only participants with interest in the topics can apply.

We divided the participants into 12 groups. Each group was randomly formed at the time of study sessions. To mitigate acquaintance issue, we used partitions between the participants and randomly assigned usernames for each session. Each group consisted of 3-4 discussants and a moderator all from the same university.

Conditions

To evaluate AP and MR, the two key components of Solution-Chat, separately, we compared three conditions in the study: Baseline, AP, and AP+MR. Note that we cannot evaluate MR separately since MR depends on AP: many recommendations of MR are triggered by the events from AP.

We have provided the pre-selected discussion structure in all conditions to observe the effects that AP and MR provide on top of discussion structure itself. The baseline interface was configured to expose the discussion structure to participants during chat. In the AP condition, we enabled all features of AP on top of the baseline: the current stage is explicitly highlighted to all participants, participants can add featured opinions to the current stage, and participants can vote on the featured opinions. In the AP+MR condition, we enabled all features of MR on top of the AP condition. In this condition, moderators receive moderator message recommendations from Block MR and Inline MR.

Tasks and Procedures

Each group was asked to discuss one topic for each condition, a total of three topics in a fixed order. We counterbalanced the order of interface conditions across groups. The discussion topics were: 1) subjectivity in the academic grading system, 2) the inconvenience of the course registration system, and 3) low quality of cafeteria food.

A moderator was randomly chosen for each group with a draw at the beginning of the study. The designated moderator led the group's all three discussions. Each discussion session was 20 minutes, followed by a 10-minute break. We enforced a hard time limit for each discussion session.

For each experimental condition, we prepared an interactive tutorial with customized content for the moderator and discussants. All participants followed the tutorial prior to accessing

the actual interface. After each discussion session, we blocked the whole interface and asked participants to fill out a survey.

Measures and Analysis

We measured participants' perceived trackability of discussion and perceived moderation quality for each session with 7-point Likert scale questions. To measure the perceived trackability of discussion, we asked participants to rate how well they were able to 1) understand the overall discussion structure, 2) see what stage the discussion was currently at, and 3) keep track of main opinions for each stage.

To measure the perceived moderation quality from discussants' perspective, we asked discussants to evaluate their moderators for their managerial and pedagogical support. For the managerial support, discussants rated how well the moderator 1) introduced the stage, 2) managed the time, and 3) managed the discussion when stalled. For the pedagogical support, discussants rated how well the moderator guided them to provide 1) objective and 2) balanced opinions.

To analyze how AP and MR affect the actual moderator actions, we conducted a discourse analysis with the moderators' messages. We first categorized each message following Lund's discussion support categories [30]: managerial, pedagogical, social, and interaction. We then further distinguished summarizing messages (pedagogical-summary) from the other types of pedagogical messages (pedagogical non-summary) for their difference in the purpose of support. While summary messages are used to recap important ideas or decisions, non-summary pedagogical messages (e.g., asking for evidence or elaboration) are used to elicit and expand discussants' ideas. Two researchers independently coded the total of 639 moderator messages from 36 sessions into one of five categories (managerial, pedagogical-summary, pedagogical non-summary, social, and interaction). Inter-rater reliability was 0.66 (Cohen's κ). The two researchers resolved conflicts through discussion and finalized the labels.

RESULTS

We first present a summary of moderator messages for each condition. Figure 3 shows the moderator message counts of each condition. The total moderator message counts for each condition were 201, 158, and 280 for the baseline, AP, and AP+MR, respectively. For the AP+MR condition, 154 (55.0%) were user-generated comments and 126 (45.0%) were system-recommended comments. Out of 639 messages, there were 332 managerial, 124 pedagogical non-summary, 81 pedagogical-summary, 75 social, and 25 interaction messages. Figure 4 shows the moderator message counts of each category for each condition.

To test the hypotheses, we conducted pairwise comparisons among three conditions (baseline, AP, and AP+MR). We conducted the Wilcoxon signed-rank tests on 1) the number of messages of each type and 2) Likert scale scores. To handle the multiple comparisons issues, we adjusted p-values with the Holm-Bonferroni correction (indicated as p_A) and used them to determine the significance.

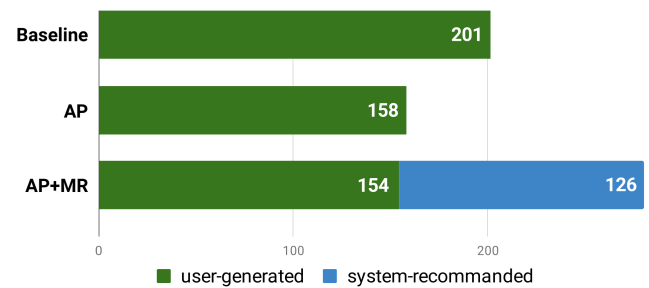


Figure 3. Moderation message counts per condition. For AP+MR condition, user-generated messages and system-recommended messages are distinguished.

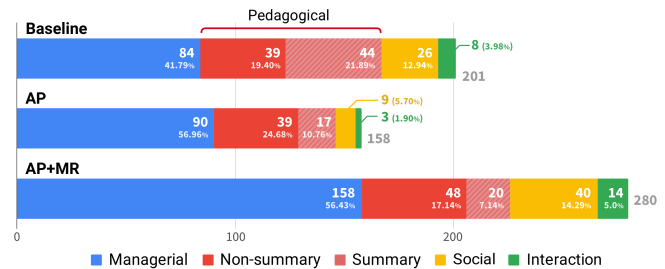


Figure 4. A distribution of moderation message counts per support type. Moderators mainly provided pedagogical and managerial support to discussants during chat-based discussion.

H1 (Participants hold higher awareness of the discussion structure with AP.) Participants showed higher perceived awareness of the discussion structure in AP and AP+MR conditions than in the baseline condition. For the question on the overall structure, the average scores were 4.91 (SD=1.36), 5.38 (SD=1.35), and 5.36 (SD=1.37) for the baseline, AP, and AP+MR conditions, respectively. Despite the difference in the average scores, however, the pairwise differences between baseline and AP and between baseline and AP+MR were not significant. For the question on the current stage, the average scores were 4.95 (SD=1.56), 5.73 (SD=1.24), and 5.62 (SD=1.45) for the baseline, AP, and AP+MR conditions, respectively. The pairwise differences between baseline and AP and between baseline and AP+MR were both significant ($p_A < 0.005$ and $W = 1080$ for baseline-AP and $p_A < 0.05$ and $W = 1899.5$ for baseline-AP+MR). There was no significant difference between AP and AP+MR.

H2 (Moderators provide better summarization support with AP.) Moderators did fewer summarizing actions in the AP and AP+MR conditions than in the baseline condition. The average number of summary prompts per session was 3.37 (SD=2.46), 1.42 (SD=1.08), and 1.67 (SD=1.77) for the baseline, AP, and AP+MR conditions, respectively. The pairwise differences between baseline and AP and between baseline and AP+MR were both significant ($p_A < 0.05$ and $W = 5$ for baseline-AP, $p_A < 0.05$ and $W = 1$ for baseline-AP+MR). There was no significant difference between AP and AP+MR condition.

Despite the reduced summarization actions by moderators, discussants said that they could keep track of main opinions for each stage. For a 7-point Likert scale question on this, the average scores were 4.93 (SD=1.33), 5.75 (SD=1.33), and

5.71 (SD=1.09) for the baseline, AP, and AP+MR conditions, respectively. The pairwise differences between baseline and AP and between baseline and AP+MR were both significant ($p_A < 0.005$ and $W = 956.5$ for baseline-AP, $p_A < 0.005$ and $W = 2032$ for baseline-AP+MR).

H3 (Moderators provide better managerial support with more managerial messages with MR.) Moderators did more managerial actions in the AP+MR condition than in the baseline and AP conditions. The number of managerial prompts per session was 7.00 (SD=2.98), 7.50 (SD=5.10), and 13.17 (SD=6.57) on average for the baseline, AP, and AP+MR conditions. The pairwise difference between baseline and AP+MR was significant ($p_A < 0.05$ and $W = 4.5$). There was no significant difference between baseline and AP conditions and AP and AP+MR conditions.

In accord with the increase in the moderators' managerial actions, discussants gave higher scores for their moderator's managerial support in the AP+MR condition than in the baseline. The average scores for stage introduction were 4.70 (SD=1.53), 5.23 (SD=1.49), and 5.51 (SD=1.43), for stalled discussion management were 4.77 (SD=1.64), 5.09 (SD=1.41), and 5.51 (SD=1.52), and for time management were 4.21 (SD=1.74), 4.74 (SD=1.54), and 5.09 (SD=1.50) for the baseline, AP, and AP+MR conditions, respectively. For the questions on stage introduction and discussion management, the differences between the baseline and AP+MR condition were significant with $p_A < 0.05$. For the question on time management, the difference between the baseline and the AP+MR conditions showed limited significance with $p_A = 0.058$. The difference between AP and AP+MR and baseline and AP were insignificant for all three questions.

H4 (Moderators provide better pedagogical support with more pedagogical messages with MR.) Our last hypothesis on MR's support in moderator's pedagogical actions was not supported by the data. First of all, the number of non-summary prompts per session showed no significant difference between groups although the mean count was slightly higher in the AP+MR condition (3.42 (SD=2.27), 3.33 (SD=2.46), and 4.50 (SD=2.50) for baseline, AP, and AP+MR, respectively). Also, there was no difference between the conditions in discussants' evaluation on the moderator's pedagogical support.

However, this is not a surprising result considering that MR is built upon AP that increases discussants' understanding of overall discussion structure and trackability of discussion. Some pedagogical messages (non-summary) are used to guide the direction or scope of the discussion, while discussants' need for such guide might have diminished with higher awareness and understanding of the discussion.

Ordering Effect

We acknowledge that our results might be influenced by the ordering of experiment conditions despite our counterbalancing. To address this issue and test our counterbalancing design, we also ran regression models controlling for the ordering effect as well as the discussion group/moderator fixed effect. We used the ordered probit model for 7-point Likert scale measures and the OLS model for the message count measures. The

regression results were generally consistent with our previous results, showing that our counterbalancing was effective.

DISCUSSION

In this section, we discuss design considerations for supporting chat-based structured discussion, with focus on enhancing structure awareness and lowering the moderator's overhead.

Structure Awareness Support

To improve participants' awareness on the discussion structure, the system should support the following.

First, the system should help discussants understand the overall discussion structure so they know how the discussion will proceed. Our results suggest that showing the structure to all discussants is effective in increasing overall structure awareness. While simply showing the structure achieves the goal to some extent, including feature opinions and highlighting the current structure have further improved structure awareness. All in all, we believe structure awareness could be achieved through multiple elements in addition to simple display.

Second, the system should help participants stay in sync with the current stage in a discussion. To achieve this goal, SolutionChat uses both AP and MR, and we believe they serve complementary roles. While AP served as a static place with always-on structure information, MR served as a way to dynamically remind discussants of the structure.

In our experiment, many moderators used SolutionChat's message recommendations for stage introduction at the beginning of a stage, which we believe positively impacted the discussants' evaluation of moderators. While discussants could always check the AP for up-to-date information, they appreciated the fact that moderators also additionally reminded them. This likely helped them because they could keep their visual focus on the chat window while following the chat flow, without having to look at the AP. It remains unclear, however, under what condition these two UI components work in a complementary manner. We believe future chat systems could incorporate more static components that are effective in reminding users of the structure, and we present one such model in this work via a combination of AP and MR.

Moderator Message Support

Based on the behaviors of moderators and discussants, we found four considerations in terms of suggesting repetitive messages, recommendation quality, accommodating diverse moderator styles, and fail-safe recommendations.

First, moderators found it useful to get recommendations for repetitive messages. Many moderators gave positive feedback on the efficiency of using MR. "As the recommended messages are often similar to what I intended to say, it was convenient to use MR [M12].", and "Within a short time, I was able to lead the discussion in the desired direction [M7].". Some moderators mentioned they had learned to lead the discussion better through MR. "At first, it was difficult to make facilitation messages and understand the tutorials, but MR made it 1000000 times easier [M9].", and "There were times when I was frustrated because I do not know how to lead the

next (stage) and MR helped me a lot [M10].” We believe our choice of focusing on supporting repetitive messages had positive effect on efficiency of moderation. We also believe interesting future opportunities exist for training moderators with system support, which we leave as future work.

Second, the quality of system-recommended messages and how they are presented and perceived should be considered together. Previous research on algorithmic aversion shows that humans lose trust on algorithms as they make errors, even with overall high accuracy [12]. While some moderators were positive about the quality of MR, others expressed distrust and found recommendations irrelevant at times. This is likely due to contextually misaligned recommended messages. But we also note that accuracy is one of the many defining factors for users’ trust on the system: the way recommendation is presented (e.g., via dismissable real-time recommendations) and the way explanations are provided (e.g., “this message helps because...”) could also heavily affect people’s usage. We observed that moderators enjoyed having contextual, real-time recommendations they could easily ignore. Because the goal of the system was not to make perfect recommendations, moderators also had lower expectations on its accuracy.

Third, the system should accommodate diverse messaging styles of moderators. Two moderators pointed out lack of diversity in MR: “I found no fun in the recommended messages because all the messages look the same. (...) I hope there is more variety in recommendations [M2].” Some moderators combined their own message with the recommendation. For example, before a moderator took a recommendation to proceed to the next stage, the moderator wrote their own message first (e.g., “To go faster.”) and took the system recommendation (“Shall we proceed to the next stage?”). This suggests an interesting way users adapt to system recommendations and we believe supporting more customizable and personalized messaging styles is promising future work.

Finally, the system should minimize the cost of inaccurately recommended messages. While we believe that the relatively low NLU accuracy was not a show-stopper because moderators rated 5 out of 7 on average about the relevance of the recommendation (average click rate of MR: Inline MR=9.35%, Block MR=10.19% per session). However, one moderator offered negative feedback on the accuracy: “It’s hard for a person to predict which direction the discussion will take. There is also a lot of discretionary judgment of the moderator in deciding the course of the discussion, and in this regard, MR seems to be useless [M8].” Earlier in the project our goal was to build an automated moderator but we soon realized this is not the right path even with perfect NLU. Human moderators serve much broader and more creative roles than sending predictable and canned messages. That is why we decided to make these boring parts of moderation easy so that moderators could focus on their more unique role as a human moderator who is also heavily invested in the community they care about.

Challenges in Live Deployment

To apply SolutionChat to real world communities, we suggest accommodating flexible problem-solving processes and

dynamic member composition online. Firstly, different communities may want to design their own discussion structure. For example, a community may want to add “consider moral acceptance of the solution” as a stage. While SolutionChat supports flexible editing of the structure, no group used this feature in our lab study, likely due to short and timed sessions. We plan to study various types of discussion structure existing communities employ and find ways to support them as templates. Secondly, the structure awareness support should consider constantly joining and leaving members. While AP shows a summary of each stage’s discussion status, for a new discussant, following up with the previous discussion is one thing but being able to actively contribute presents additional challenges. More interaction and social support could be a key to creating a more welcoming online chat discussion group.

LIMITATIONS

Our study has several limitations that should be recognized. First, our experiment design led to an ordering effect. As participants gained familiarity with the system over time, we suspect the learning effect has occurred. However, despite the learning effect, most of our results still hold as shown in our analysis. While we considered running a between-subjects study, uncontrolled group dynamics and individual styles may have introduced even larger confounds to the study. Second, 20 minutes of discussion time may be too short. Clearly, no group finished their discussion within 20 minutes. However, we believe that 20 minutes can still capture meaningful discussion and moderation behavior. Session behavior logs suggest adequate use of AP and MR was recorded for 20 minutes.

CONCLUSION

Despite the increase in communities’ problem-solving efforts in online chat platforms, the platforms provide limited support for structured discussion. This paper introduces a system that enhances structure awareness and provides real-time moderation support. For future work, we will explore ways to assist structured discussions for more diverse types of goals, such as collective action and event planning. Also, we plan to support real-time consensus making that considers the volatile member configurations of real-world communities.

ACKNOWLEDGEMENTS

This work was supported by the Office of Naval Research (ONR: N00014-18-1-2834), and by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2016-0-00564, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding).

REFERENCES

- [1] Christa SC Asterhan and Baruch B Schwarz. 2007. The effects of monological and dialogical argumentation on concept learning in evolutionary theory. *Journal of educational psychology* 99, 3 (2007), 626.
- [2] Christa SC Asterhan and Baruch B Schwarz. 2009. Argumentation and explanation in conceptual change: Indications from protocol analyses of peer-to-peer dialog. *Cognitive science* 33, 3 (2009), 374–400.

- [3] Christa SC Asterhan and Baruch B Schwarz. 2010. Online moderation of synchronous e-argumentation. *International Journal of Computer-Supported Collaborative Learning* 5, 3 (2010), 259–282.
- [4] R Bartlett and M Deseriis. 2016. Loomio and the Problem of Deliberation. *Open Democracy* (2016).
- [5] Mia Xu Chen, Benjamin N Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M Dai, Zhifeng Chen, and others. 2019. Gmail Smart Compose: Real-Time Assisted Writing. *arXiv preprint arXiv:1906.00080* (2019).
- [6] Christine Chin and Jonathan Osborne. 2010. Supporting argumentation through students' questions: Case studies in science classrooms. *The Journal of the Learning Sciences* 19, 2 (2010), 230–284.
- [7] Elaine B Coleman. 1998. Using explanatory knowledge during collaborative problem solving in science. *Journal of the Learning Sciences* 7, 3-4 (1998), 387–427.
- [8] Jeff Conklin. 2006. *Wicked problems & social complexity*. CogNexus Institute San Francisco, CA.
- [9] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, and others. 2018. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190* (2018), 12–16.
- [10] Fiorella De Cindio and Stefano Stortone. 2013. Experimenting liquidfeedback for online deliberation in civic contexts. In *International Conference on Electronic Participation*. Springer, 147–158.
- [11] Erica De Vries, Kristine Lund, and Michael Baker. 2002. Computer-mediated epistemic dialogue: Explanation and argumentation as vehicles for understanding scientific notions. *The journal of the learning sciences* 11, 1 (2002), 63–103.
- [12] Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. 2015. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General* 144, 1 (2015), 114.
- [13] Thomas Erickson, Christine Halverson, Wendy Kellogg, Mark Laff, Peter Malkin, and Tracee Wolf. 2001. Loops: Designing A Web-Based Environment for Persistent, Semi-Structured Conversation. *Yorktown Heights, NY* (2001).
- [14] Shelly Farnham, Harry R Chesley, Debbie E McGhee, Reena Kawal, and Jennifer Landau. 2000. Structured online interactions: improving the decision-making of small discussion groups. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 299–308.
- [15] Robert Farrell, Peter G Fairweather, and Kathleen Snyder. 2001. Summarization of discussion groups. In *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 532–534.
- [16] Fiona E Fox, Marianne Morris, and Nichola Rumsey. 2007. Doing synchronous online focus groups with young people: Methodological reflections. *Qualitative health research* 17, 4 (2007), 539–547.
- [17] Google. 2016. GBoard - the Google Keyword. (May 2016). <https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin> Accessed: 2019-09-20.
- [18] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652* (2017).
- [19] Kevin A Hill and John E Hughes. 1998. *Cyberpolitics: Citizen activism in the age of the Internet*. Rowman & Littlefield Publishers, Inc.
- [20] Shagun Jhaver, Iris Birman, Eric Gilbert, and Amy Bruckman. 2019a. Human-Machine Collaboration for Content Regulation: The Case of Reddit Automoderator. *ACM Trans. Comput.-Hum. Interact.* 26, 5, Article 31 (July 2019), 35 pages. DOI: <http://dx.doi.org/10.1145/3338243>
- [21] Shagun Jhaver, Amy Bruckman, and Eric Gilbert. 2019b. Does Transparency in Moderation Really Matter?: User Behavior After Content Removal Explanations on Reddit. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 150 (Nov. 2019), 27 pages. DOI: <http://dx.doi.org/10.1145/3359252>
- [22] Ian Kearns, Jamie Bend, and Beatrice Stern. 2002. *E-participation in local government*. Institute for Public Policy Research.
- [23] Alison King and Barak Rosenshine. 1993. Effects of guided cooperative questioning on children's knowledge construction. *The Journal of Experimental Education* 61, 2 (1993), 127–148.
- [24] Mark Klein. 2011. How to harvest collective wisdom on complex problems: An introduction to the mit deliberatorium. *Center for Collective Intelligence working paper* (2011).
- [25] Lorenz Cuno Klopfenstein, Saverio Delpriori, Silvia Malatini, and Alessandro Bogliolo. 2017. The rise of bots: A survey of conversational interfaces, patterns, and paradigms. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. ACM, 555–565.
- [26] Travis Kriplean, Jonathan Morgan, Deen Freelon, Alan Borning, and Lance Bennett. 2012. Supporting reflective public thought with considerit. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 265–274.

- [27] Lance B Kurke and Howard E Aldrich. 1983. Note—Mintzberg was right!: A replication and extension of the nature of managerial work. *Management science* 29, 8 (1983), 975–984.
- [28] Sasha Lekach. 2017. The huge TechHasNoWalls protest started on Slack. (Feb 2017). <https://news.yahoo.com/huge-techhasnowalls-protest-started-slack-213408329.html>
- [29] Weichen Liu, Sijia Xiao, Jacob T Browne, Ming Yang, and Steven P Dow. 2018. ConsensUs: Supporting multi-criteria group decisions by visualizing points of disagreement. *ACM Transactions on Social Computing* 1, 1 (2018), 4.
- [30] Kristine Lund. 2004. Human support in CSCL. In *What we know about CSCL*. Springer, 167–198.
- [31] Brian McInnis, Gilly Leshed, and Dan Cosley. 2018. Crafting Policy Discussion Prompts As a Task for Newcomers. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 121 (Nov. 2018), 23 pages. DOI: <http://dx.doi.org/10.1145/3274390>
- [32] Microsoft. 2017. Word Flow Keyboard. (2017). <https://www.microsoft.com/en-us/garage/profiles/word-flow-keyboard/> Accessed: 2019-09-20.
- [33] Michael D Mumford and Mary S Connelly. 1991. Leaders as creators: Leader performance and problem solving in ill-defined domains. *The Leadership Quarterly* 2, 4 (1991), 289–315.
- [34] Michael D Mumford and Sigrid B Gustafson. 1988. Creativity syndrome: Integration, application, and innovation. *Psychological bulletin* 103, 1 (1988), 27.
- [35] Gabriel Murray, Giuseppe Carenini, and Shafiq Joty. 2018. NLP for Conversations: Sentiment, Summarization, and Group Dynamics. In *Proceedings of the 27th International Conference on Computational Linguistics: Tutorial Abstracts*. 1–4.
- [36] Rachel Quednau. 2016. Top Strong Citizen Discussions this Week. (Apr 2016). <https://www.strongtowns.org/journal/2016/4/13/top-strong-citizens-discussions?rq=slack>
- [37] BB Schwartz, Y Neuman, and S Biezuner. 2000. Two wrongs may make a right... if they argue. *Cognition and Instruction* 18, 4 (2000), 461–494.
- [38] Naama Tepper, Anat Hashavit, Maya Barnea, Inbal Ronen, and Lior Leiba. 2018. Collabot: Personalized Group Chat Summarization. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 771–774.
- [39] TouchPal. 2008. TouchPal Keyboard. (2008). <http://www.touchpal.com/> Accessed: 2019-09-20.
- [40] TouchType. 2010. SwiftKey Keyboard. (2010). <https://swiftkey.com/en> Accessed: 2019-09-20.
- [41] Niels van Berkel, Jorge Goncalves, Danula Hettiachchi, Senuri Wijenayake, Ryan M. Kelly, and Vassilis Kostakos. 2019. Crowdsourcing Perceptions of Fair Predictors for Machine Learning: A Recidivism Case Study. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 28 (Nov. 2019), 21 pages. DOI: <http://dx.doi.org/10.1145/3359130>
- [42] Carla Van Boxtel, Jos Van der Linden, and Gellof Kanselaar. 2000. Collaborative learning tasks and the elaboration of conceptual knowledge. *Learning and instruction* 10, 4 (2000), 311–330.
- [43] Ilse Verdiesen, Martijn Cligge, Jan Timmermans, Lennard Segers, Virginia Dignum, and Jeroen van den Hoven. 2016. MOOD: Massive Open Online Deliberation Platform-A Practical Application.. In *EDIA@ ECAI*. 4–9.
- [44] Scott Wright. 2009. The role of the moderator: Problems and possibilities for government-run online discussion forums. *Online deliberation: Design, research, and practice* (2009), 233–242.
- [45] GE Xun and Susan M Land. 2004. A conceptual framework for scaffolding III-structured problem-solving processes using question prompts and peer interactions. *Educational technology research and development* 52, 2 (2004), 5–22.
- [46] Amy X Zhang and Justin Cranshaw. 2018. Making sense of group chat through collaborative tagging and summarization. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 196.
- [47] Amy X Zhang, Lea Verou, and David Karger. 2017. Wikum: Bridging discussion forums and wikis using recursive summarization. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, 2082–2096.
- [48] Liang Zhou and Eduard H Hovy. 2006. On the Summarization of Dynamically Introduced Information: Online Discussions and Blogs.. In *AAAI Spring symposium: Computational approaches to analyzing weblogs*. 237.