Wikum: Bridging Discussion Forums and Wikis Using Recursive Summarization

Citation: Zhang, Amy X., Lea Verou, and David Karger. “Wikum.” Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing - CSCW ’17 (2017).

As Published: http://dx.doi.org/10.1145/2998181.2998235

Publisher: Association for Computing Machinery (ACM)

Persistent URL: http://hdl.handle.net/1721.1/112676

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Wikum: Bridging Discussion Forums and Wikis Using Recursive Summarization

Amy X. Zhang
MIT CSAIL
Cambridge, MA, US
axz@mit.edu

Lea Verou
MIT CSAIL
Cambridge, MA, US
leaverou@mit.edu

David Karger
MIT CSAIL
Cambridge, MA, US
karger@mit.edu

ABSTRACT

Large-scale discussions between many participants abound on the internet today, on topics ranging from political arguments to group coordination. But as these discussions grow to tens of thousands of posts, they become ever more difficult for a reader to digest. In this article, we describe a workflow called *recursive summarization*, implemented in our *Wikum* prototype, that enables a large population of readers or editors to work in small doses to refine out the main points of the discussion. More than just a single summary, our workflow produces a *summary tree* that enables a reader to explore distinct subtopics at multiple levels of detail based on their interests. We describe lab evaluations showing that (i) Wikum can be used more effectively than a control to quickly construct a summary tree and (ii) the summary tree is more effective than the original discussion in helping readers identify and explore the main topics.

Author Keywords

online discussion; comments; threaded discussion; wikis; summarization; deliberation; collaboration; crowdsourcing

ACM Classification Keywords

H.5.3. Group and Organization Interfaces: Asynchronous interaction; Web-based interaction

INTRODUCTION

Large online discussions involving many participants are pervasive on the web. News and entertainment sites offer comment systems that support discussion of primary content (articles, videos, blog posts) while on other sites the discussion is itself the primary content (Google Groups, forums). These discussions contain a diversity of rich information, including differing opinions on an issue, anecdotes, humor, explanations, coordination, and deliberation, and may continue to be consulted long after the discussion has died down. Over the course of thousands of comments, even open mathematics problems can be solved [10] and bitter battles on Wikipedia over controversial edits settled [27].

CSCW '17, February 25–March 01, 2017, Portland, OR, USA.
Copyright © 2017 ACM. ISBN 978-1-4503-4335-0/17/03 $15.00.
DOI: http://dx.doi.org/10.1145/2998181.2998235

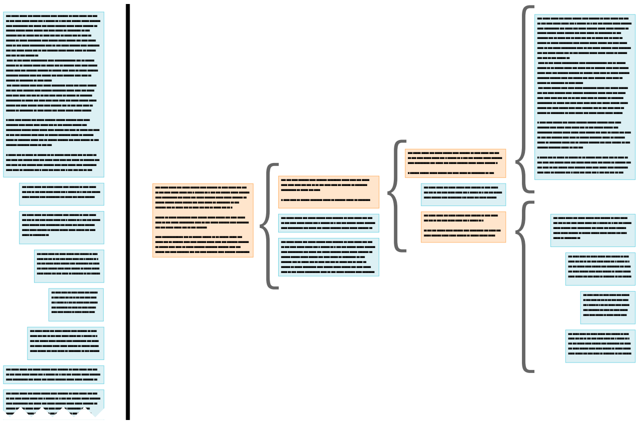
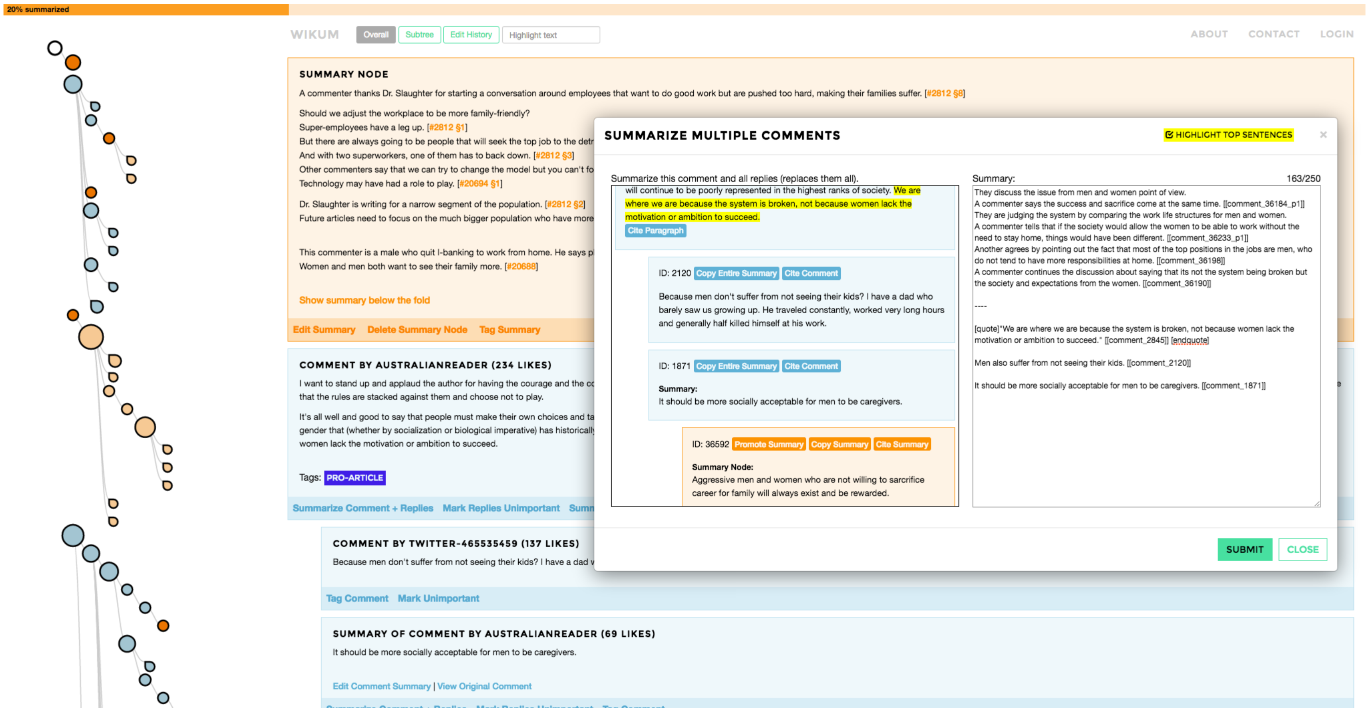


Figure 1: **Left:** Discussions are often long and difficult to get an overview. **Right:** Recursive summaries can be constructed to enable progressive hierarchical exploration.

On the downside, such discussions are often "append only." They simply grow, without any kind of organization or summarization. Readers, especially latecomers, need to invest significant time and effort reading to understand a discussion. Though there may be thousands of prior readers, each new reader must individually dig through the same threads of conversation to achieve understanding. There can also be too many tangents and nested layers of discussions to easily navigate. This is so much work that new readers often don't bother, and proceed to post redundant discussion.

Current techniques of sorting, filtering, and moderating comments can reduce but not solve these problems. These techniques only select a subset of the comment *texts*; they do not digest or organize their *ideas*. A large number of high quality, popular comments may be upvoted that are all saying much the same thing. Such *redundancy* in discussions may arise independent of quality, making it laborious for participants to identify all facets of the discussion. Similarly, an issue may be argued back and forth and ultimately *resolved*, or incorrect statements may be *refuted*. But these obsolete arguments and incorrect statements remain part of the discussion that a user must wade through to get to the conclusions.

For those seeking a general overview, a short textual *summary* is the traditional solution. But writing a summary of a large discussion will be a massive task, unlikely to appeal to the many readers who do not even bother to *read* the entire



SUMMARY NODE

A commenter thanks Dr. Slaughter for starting a conversation around employees that want to do good work but are pushed too hard, making their families suffer. [#2812 §8]

Should we adjust the workplace to be more family-friendly?
Super-employees have a leg up. [#2812 §1]
But there are always going to be people that will seek the top job to the detr
And with two superworkers, one of them has to back down. [#2812 §3]
Other commenters say that we can try to change the model but you can't fo
Technology may have had a role to play. [#20694 §1]

Dr. Slaughter is writing for a narrow segment of the population. [#2812 §2]
Future articles need to focus on the much bigger population who have more

This commenter is a male who quit I-banking to work from home. He says pl
Women and men both want to see their family more. [#20688]

COMMENT BY AUSTRALIANREADER (234 LIKES)

I want to stand up and applaud the author for having the courage and the co
that the rules are stacked against them and choose not to play.

It's all well and good to say that people must make their own choices and ta
gender that (whether by socialization or biological imperative) has historically
women lack the motivation or ambition to succeed.

COMMENT BY TWITTER-465535459 (137 LIKES)

Because men don't suffer from not seeing their kids? I have a dad v

SUMMARY OF COMMENT BY AUSTRALIANREADER (69 LIKES)

It should be more socially acceptable for men to be caregivers.

SUMMARIZE MULTIPLE COMMENTS

Summarize this comment and all replies (replaces them all).

will continue to be poorly represented in the highest ranks of society. We are where we are because the system is broken, not because women lack the motivation or ambition to succeed.

ID: 2120

Because men don't suffer from not seeing their kids? I have a dad who barely saw us growing up. He traveled constantly, worked very long hours and generally half killed himself at his work.

ID: 1871

Summary:
It should be more socially acceptable for men to be caregivers.

ID: 36592

Summary Node:
Aggressive men and women who are not willing to sarcrifice career for family will always exist and be rewarded.

Summary: 163/250

They discuss the issue from men and women point of view.
A commenter says the success and sacrifice come at the same time. [[comment_36184_p1]]
They are judging the system by comparing the work life structures for men and women.
A commenter tells that if the society would allow the women to be able to work without the need to stay home, things would have been different. [[comment_36233_p1]]
Another agrees by pointing out the fact that most of the top positions in the jobs are men, who do not tend to have more responsibilities at home. [[comment_36198]]
A commenter continues the discussion about saying that its not the system being broken but the society and expectations from the women. [[comment_36190]]

[quote]"We are where we are because the system is broken, not because women lack the motivation or ambition to succeed." [[comment_2845]] [endquote]

Men also suffer from not seeing their kids. [[comment_2120]]

It should be more socially acceptable for men to be caregivers. [[comment_1871]]

Figure 2: The Wikum interface. Orange nodes are summaries, blue and light orange nodes are original comments. Two of the summaries are expanded, to uncover the comments they are summarizing. An editing window is open to summarize a subthread.

discussion. Also, a typical summary offers no way to dive deeper into specific areas based on the reader's interest level or refer back to individual comments.

Contribution

To address these problems, we consider how a *group* of people could individually contribute small amounts of work to refine a large discussion into a *dynamic* textual summary that can be explored at *varying levels of detail*. In this work, we present the concept of a *summary tree*, an artifact that is a tree of short summaries of distinct subtopics of a discussion. The summaries are made at different levels of detail so that a higher-level summary covers a greater portion of the discussion. It reflects the paradigm of a good article, where an abstract gives a brief summary of the whole, the introduction summarizes at greater detail, and then individual sections (with their own high level introductions) cover subtopics at even greater detail. By leveraging its online nature, the summary tree is an *expandable* artifact that empowers readers to explore multiple levels of detail, including diving all the way down into original comments. The tree is also akin to topical taxonomies or hierarchical clusterings of items, but in this case each node contains its own substantive information summarizing all nodes nested within.

We design a workflow to create a summary tree using the idea of *recursive summarization* of a discussion, where users build summaries of small sections of the discussion, small sets of those summaries are then aggregated and summarized, and so on until the entire discussion is incorporated into the layered summary tree. Each unit of work requires only writing a short summary of a small number of unsummarized comments or lower-level summaries, so no editor need contribute excessive effort. This way, a group of participants can each do small amounts of work to collectively convert an unwieldy discussion into a short summary of the entire discussion.

To explore the design space of this process, we developed **Wikum** (a portmanteau of wiki and forum), a system for creating summaries and reading a discussion overlaid with summaries. As seen in Figure 2, Wikum combines a directly-manipulatable node-link tree visualization with a view that shows the summaries and comments in focus, as well as a wiki-like editing modal. Readers can explore the discussion, starting at a root summary and drilling into summaries that eventually expand to the original discussion. Editors can edit summaries or contribute additional summaries of unsummarized portions of the discussion.

We performed a lab evaluation to determine the feasibility of our recursive workflow, or how easy it would be to collectively summarize a large discussion using Wikum. As a control, we used a Google Doc file with track changes turned on, mimicking an unstructured wiki editing text box. Studying the contributions of 20 participants, we found that the same groups of users working in both Wikum and Google Docs were faster at summarizing the discussion in Wikum and also rated it as easier to use. In the Google Doc condition, we saw that users were reluctant to edit other people's work, choosing to append to ever-growing summaries, which ultimately defeated the purpose of summarization. This pitfall was avoided in Wikum as a higher-level summary overlays but does not tamper with other people's work. We performed a second lab evaluation of the created summary trees to understand readers' perceptions of their quality and usefulness. We found evidence from 13 additional participants that Wikum was helpful for quickly getting an overview of the discussion.

RELATED WORK

Filtering and Moderation

Filtering is at present the dominant approach to reducing large discussion volume. Many discussion interfaces today have some form of collective social moderation using voting. However, there are documented problems, including underprovision [17] and negative feedback loops [7]. Better mechanisms for personalization or recommendation can filter down some of the noise but also may lead to "filter bubbles" when only one point-of-view is represented [40]. Additionally, many discussion spaces now employ moderators to filter comments [30] or use community mechanisms such as voting or flagging [11]. Finally, many sites and researchers have experimented with automated filtering, such as for detecting spam [35] and trolls [8]. As we argued above, filtering can only go so far. Social moderation may surface only "popular" points. Comments may still be too numerous, have many tangents, and be redundant. Wikum goes a step further by considering how people can not just filter comments out but instead synthesize major points made.

Visualizing Discussions and Opinions

Many online discussions on the web today arrange comments in a linear fashion ordered chronologically. Those that are *threaded* often use indentation of the comments to indicate reply structure. Researchers have developed novel alternative presentations to help navigate threads and get an overview of a discussion. For instance, FlashForums provided a thumbnail view of the discussion that users could highlight portions of to see the full comments [12]. Other systems tried mixed-modal visualizations that show threaded conversations in both a tree and sequential way [48]. More recently, visualizations such as ConVisit [23] take this a step further, allowing users to perform interactive topic modeling over a thumbnail view. We make use of a graphical tree view as well in the Wikum interface to allow readers to see the shape of the discussion. There are also researchers that have explored more abstract visual representations of conversations to convey mood, temporal activity, activity by individuals [13], high level content [50], or reply structure [24].

A visualization that goes in a different direction is Opinion Space [15], an interface that maps users' opinions on a two-dimensional space. This provides a visual representation of the diversity of opinions and encourages exploration of divergent points of view. However, this interface does not allow for any actual discussion of the opinions presented. Similarly, ConsiderIt [28] allows users to build up pro-con lists on different issues, including remixing lists written by other people, but has no support for people to hash out disagreements or argue for their point-of-view. In contrast, Wikum aims to support getting an overview of a back-and-forth discussion.

Coordinating Wikis and Discussion

There are communities and systems that have tried to combine a forum for discussions with a community-maintained wiki or other repository for collecting knowledge [1]. Research on community wikis found that they were useful for managing frequently asked questions [21]. Examples include ExpertNet, a coupled forum and wiki system for government officials to solicit feedback from public experts [38], and Polymath, a successful large scale math collaboration which used a combination of comments, blog posts, and wikis [19]. In Polymath, the two leaders chose to summarize all discussions, a task they found time-consuming but also rewarding. Still, there were issues with newcomers feeling overwhelmed by the discussion. Wikum incorporates some of the design suggestions raised by studies of Polymath [10], including linking from wiki to primary content and citing comments.

Today, many social Q&A websites, especially for technical support communities, have overtaken mailing lists and other discussion forums as a place for knowledge sharing [47]. However it is unclear how well these systems perform for contentious or subjective issues. One such system that has gained popularity, Quora, has experimented with a feature called Answer Wikis [41] that aim to allow readers to synthesize the answers provided in a Quora question post. However, Quora only permits "uncontroversial, factual information" in the wiki space and has no process or structure for integrating the wiki with the discussion or ensuring the wiki covers the discussion well.

In the other direction are Wikipedia talk pages, where Wikipedia editors deliberate and coordinate their activity on a Wikipedia page [51]. These discussions can be sprawling, with discussions reaching tens of thousands of comments [31]. They are also difficult to make sense of, as there is little support for threading or collapsing of subthreads. Finally, the talk pages have little to no connection to the wiki article they are discussing, for instance to link the outcome of a deliberative discussion to the action made within the wiki.

Summarizing Discussion

Some researchers have worked on tools to provide a textual overview or summary of a discussion [42]. Currently, automatic summarization techniques have mostly focused on extractive summarizations [37] which select important sentences from a body of text. This method cannot provide a synthesis of points, such as when paraphrasing multiple redundant comments or determining a resolution from a debate. More recently, researchers have worked on abstractive summarization models [16], which seek to produce novel sentences not present in a body of text. However, most methods are not built for summarizing discussions but instead are for long documents or unconnected user reviews. Also, most techniques require massive sets of labeled training data [43] which do not exist for summaries of discussions. While automatic techniques cannot approach human efforts as of yet for our task, we consider ways they can augment editors' work.

Some systems similar to Wikum [2, 36] have been proposed that use human work to summarize discussions incrementally, but none of these systems have had formal user evaluations.

Additionally, these systems aim only for a "flat" set of top-level summaries of different topics; unlike Wikum, they do not produce summaries that can expand to reveal different levels of detail to let users drill into specific subtopics. We also *evaluate* our system on both the editing process and the reading experience. Another system explores paraphrasing individual comments within a discussion for the purpose of encouraging reflection [29], but does not have a mechanism for summarizing entire discussions. Deeper reflection can be important benefits of synthesizing conversation, and we are interested in studying how Wikum advances these goals in the future.

Scaffolding Complex Tasks Among Many Participants

A separate line of work in recent years has explored how to coordinate crowds of people doing small amounts of work to complete complex informational tasks. Much of this work has focused on breaking down large tasks into small parts and then providing scaffolding to integrate the parts. Researchers have developed workflows for tasks like summarizing books and movies [49], extracting categories and clusters from complex data [3], perform scheduling with many constraints [5], as well as taxonomy creation [9]. For most of these workflows, the intermediate steps of the workflow are discarded towards producing a final static artifact. In contrast, our work explores using summaries of portions of original content as part of the final artifact that can dynamically grow. To some extent, creating an outline [33] or article [20] does this in a limited setting, as there is often a beginning that summarizes and links to sections in the main document, which sometimes references original content. We build on this by considering how summaries could be tightly integrated with original content and have deeper levels of expansion.

In contrast with these crowdsourcing systems, our approach focuses on synthesizing online discussions and takes advantage of the existing discussion structure. As a result, our system has less workflow scaffolding, and users have the flexibility to decide how to participate. Additionally, in this work, we did not target anonymous crowd workers such as on Mechanical Turk explicitly. In particular we defer critical issues of quality control to future work.

DESIGN

We begin by outlining the major motivations that informed the design decisions around the summary tree artifact as well as the recursive summarization workflow.

Summary Tree Design

Our artifact and its implementation in the Wikum system aims to combine wikis and forums to address their respective drawbacks. Forums offer no way for someone with little time to get an overview of the discussion, while the condensation required of wikis necessarily drops much of the original detail. To address these complementary drawbacks we could directly combine the two artifacts, as in Quora Answer Wikis [41], providing a wiki page where a short summary of the entire discussion can be edited. These two components do not connect well though. There is no way to dig down into the summary in order to unpack its origins from the original discussion. The wiki also offers no support for *incremental* summarization—there's no way (aside from reading the entire discussion) to see what has already been summarized and what needs to be added.

We propose a *summary tree* as a more effective bridge that summarizes the discussion forum at *multiple* levels of detail. Summaries of small portions of the discussion can be authored, which can then be incorporated into a meta-summary. These meta-summaries can be similarly summarized, until everything is incorporated into a "root" summary of the entire conversation that serves as a starting point for hierarchically exploring the conversation. While other systems have explored creating a "flat" set of summaries of topical portions of a discussion [36], our proposed process of *recursive summarization*, which allows summarization at different depths of the discussion, provides additional benefits. A reader seeking more information can *expand* the root summary into the comments and summaries it summarizes, then choose interesting sub-summaries to expand further. They can dive down as deeply as they like, eventually reaching individual comments. Ideally, the sub-summaries of a summary will cover distinct topics, permitting a reader to focus exploration on topics of interest. As another pathway to accessing "primary source" comments in summaries, our summary tree can include (i) citations to individual comments (and lower level summaries) and (ii) quotes of text from them.

Workflow Design

Making our target artifact a summary tree also suggests a natural approach to constructing it. Starting with the original discussion tree, an editor working alone can choose an appropriately-sized group of related comments to summarize. Wikum then replaces those comments in the discussion with their summary, treating the summary much like any other comment. Editors can then continue to create new summaries that can distill both previously-written summaries and unsummarized comments, until we are left with a summary of the entire discussion. A reader can reverse this distillation process, expanding interesting summaries to arbitrary depth to acquire more detail.

An important challenge with this process is finding related comments and summaries to bring together and summarize. In the case of threaded discussion, there is a natural grouping heuristic as comments are already organized in a tree structure by reply. Editors can simply pick a small subtree to summarize, where all comments are likely discussing the same topic. Thus, the levels of the reply tree can scaffold the creation of the summary tree. However, even threaded discussions sometimes have comments that have too many replies. Also, given initially threaded comments, the recursive summarization process eventually distills each separate discussion to an individual "root"; these root summaries still need to be gathered and summarized. Likewise, non-threaded discussions have all comments at a single level. To address this, the Wikum system also allow editors to group similar comments at the same level to summarize, using methods like topic clustering, or selecting of adjacent comments (useful for chronological non-threaded discussions).

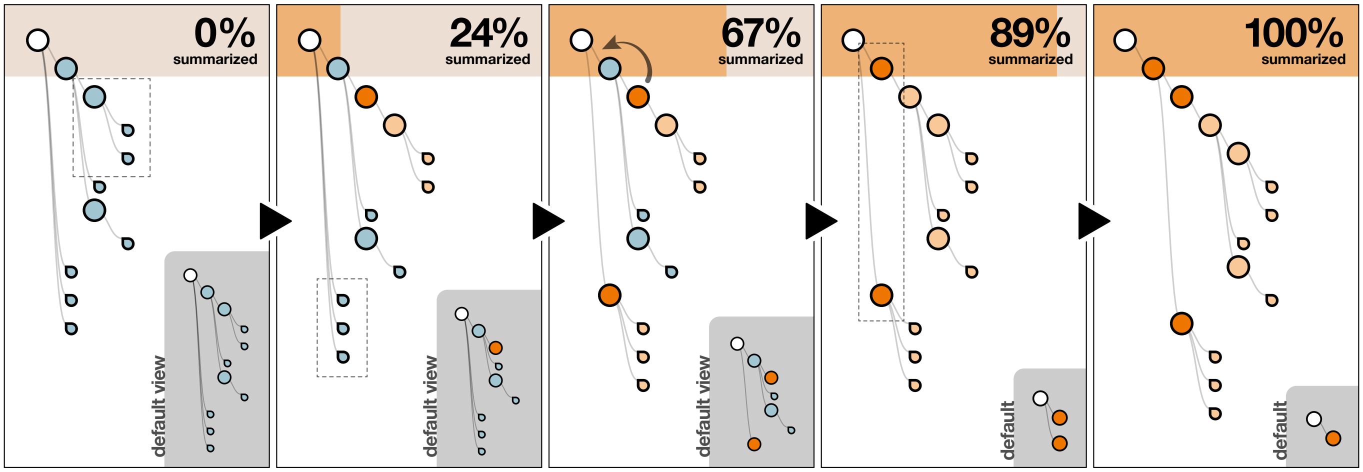


Figure 3: Summarization progress for a discussion with 10 comments. Shown here is a fully expanded view of each summary tree state, for illustrative purposes. The bottom right of each panel shows the initial (default) view when the summary tree is in the given state. 1) Initial discussion. 2) Summarizing a comment and its two replies. 3) Grouping & summarizing three root-level comments. 4) Promoting a summary one level up. 5) Summarizing the two root summaries.

Even before the summary tree is complete, the summaries that people write in Wikum are *embedded* in the original discussion and contribute towards making the discussion easier to read. In threaded discussions, the summary of a subtree (comment and its replies) lives "between" the comment and its parent. Upon reading the summary, one can expand it to see the comments it summarizes or move on. This can be beneficial to readers because it puts the summaries into context and also provides sensemaking capabilities for exploration of the discussion. Embedding the summaries into the discussion threads also makes it obvious which comments they cover and produces a visual distinction helpful for editors between summarized and unsummarized content.

We designed the summary tree with the goal of supporting effective reading, but our user studies, discussed further, revealed a second benefit. Wikum provides *additive summarization*, augmenting the underlying discussion with summaries. But ability to expand those summaries to reveal the content they summarize, as well as the ability to cite and quote original comments within a summary, makes clear that the material being summarized is *still present*. Thus, the majority of editors' work is *enriching* as opposed to deleting or editing other people's work. This mitigates some of the issues prior research has uncovered around people's reluctance to edit others' work in wiki-like environments [4].

Workflow Efficiency

Recursive summarization permits summarization to be done in small units. But one might worry that the recursive approach significantly increases the overall work requirement as content must be read and summarized at multiple levels. But this is not the case: when each summarization step causes a constant-factor decrease in the amount of as-yet-unsummarized text, the total work done will be little more than that required for one-shot summarization. To see this, suppose that any summary is shorter than the text it is summarizing by a factor of 5. We can therefore conclude that any time an editor reads w words to summarize them, the total text remaining loses $4w/5$ words. If the text starts with W words then it cannot lose more than this before it is fully summarized. Thus, the editors in total will need to read at most $5W/4$ words (of original content or summaries) before the summarization task is complete. And the total number of words written, at $1/5$ of that read by the editors, is only $W/4$. Since comments had to be written once, and are presumably being read many times, the summarization work is proportional to the work users were clearly willing to invest in the discussion in the first place. This suggests that summary tree creation requires only a scalable amount of work.

WIKUM SYSTEM DESIGN

The Wikum web interface consists of a tree visualization of the discussion and summaries made so far on the left and a display of selected comments and summaries on the right (Figure 2). Tree nodes are ordered chronologically (within threads when they exist) and can be sorted in other ways. The area of each node corresponds to the length of the corresponding text. Users can select comments by clicking nodes in the tree, which results in the right pane displaying the selected comment and any replies. Users can also select and display disjoint parts of the tree by dragging or Control-clicking. Clicking on a selected node expands or collapses its reply subtree. User-generated summaries are bright orange nodes. Unsummarized comments are displayed as light blue, while summarized comments are light orange to show they have been summarized above. Summaries are collapsed by default and clicking on them reveal the nodes they summarized.

Building the Summary Tree

For readers of a discussion, Wikum lets them see a visual overview, differentiate between summaries and comments, explore into summaries, and jump between conversations.

For editors, we provide the same interface with additional affordances for summarization. Wikum enables a number of possible edits to create the summary tree (Figure 3):

- **Mark as unimportant.** Hides the comment from view. Used for content with no information or interest value.
- **Summarize comment.** Summarizing a longer individual comment is possible. The comment then is replaced with the summary and a link to toggle the original text.
- **Summarize comment & replies.** Summarizes an entire subtree of a threaded discussion into a single summary node. Clicking on the summary node expands it to display the thread subtree.
- **Group & summarize.** Absent threads, we need a way to choose a group of posts to summarize. Even with threading, sometimes a single node may have so many children that it is too much work for one person to summarize. The group & summarize operation lets the editor select a few nodes, then group and summarize them to collapse them down to one node.
- **Promote summary.** If a summary of a subthread has been written, a person writing a summary at a higher level in the discussion thread can promote the lower summary to their position and build on the summary text; this lower summary can be a useful starting point for authoring the higher-level summary.

At the outset, as shown in Figure 3, editors may be mostly summarizing a comment and all replies (from a threaded discussion), leaving embedded summaries as signposts to future readers about whether to go down that thread. For non-threaded discussion and later stages of a threaded discussion, grouping and summarizing nodes at the same level that are topically related may be more used.

Creating High Quality Summaries Efficiently

We made additional design decisions to encourage higher quality summary writing. Clicking to summarize one or more comments causes an editing window to pop into view (Figure 2). This window displays the comment(s) to be summarized on the left, with a text area for the summary on the right.

Important sentence highlighting. We use an automatic extractive summarizer to identify and then highlight important sentences in the content, though this feature can be turned off. This was added to make it easier for people to skim content, though we do not pre-populate the text box with the sentences or allow 1-click transference, due to concerns that it would encourage low quality summaries.

Maximum length restriction. As we noticed people writing lengthy summaries in pilot sessions, Wikum enforces that each summary can be at most 250 words (about half a page) or half the length of the summarized text, whichever is smaller.

Cluster view for comments at the same level. For cases where there are too many adjacent nodes, we provide a clustered view which groups comments that are similar, to help a user select a good group to summarize. This makes it easier to group and summarize topically related comments.

Affordances for citations and quotes. Every node and paragraph within a summarized node can be *cited* in the text summary, which produces a clickable citation when browsing the discussion. Text from original comments can be *quoted* verbatim in the summaries by selecting it and clicking on "Quote". This inserts both the quoted text and a citation to its originating comment. These features were added to encourage summaries that stick to the points made in the discussion. The citations and quotes can also "bubble up" a deeper comment or quote that is interesting or well-written, useful for when readers want to quickly get to high quality comments.

Tag comments and filter by tag. Adding tags to comments is a lightweight task and can also help future summarizers by classifying topics or viewpoints expressed across multiple threads. Comments can also be filtered by specific tags.

SYSTEM IMPLEMENTATION

The Wikum system is comprised of a front end web interface built using D3, Javascript, HTML, and CSS. It also has a backend component built using the Django web framework and a MySQL database. The homepage of Wikum allows people to paste in URLs to different discussions that kick off a backend ingestion process that adds all the comments to the database. The system currently supports ingesting comments from Disqus, Reddit, and email threads in mbox format. The important sentence highlighting feature was incorporated via sumy[1], a python package implementing the LexRank algorithm for extractive summarization [14]. This algorithm was chosen after experimenting with several unsupervised extractive summarization techniques. The clustered view for comments at the same level processes the comments and clusters them by first converting each comment into a bag-of-words vector representation that has been TF-IDF normalized. Then the k-means algorithm is used to cluster the vectors. In the cluster view, the cluster with the smallest average distance between pairs of comments is shown first. There is a slider to adjust the size of the cluster, which affects the parameter of number of clusters inputted into k-means.

EVALUATION

We conducted two studies of Wikum to evaluate the process of creating a summary tree as well as the experience reading a summary tree artifact, respectively. In the first study, we sought to understand how long it would take and how easy it would be for a group of people to collectively summarize a large discussion using Wikum versus an alternative system. The second study evaluated the usefulness of the summaries created in the previous stage towards getting an overview as well as people's preferences and strategies around reading discussions using Wikum and our control settings.

Study 1: Summarization

In the first study, we evaluated how people summarized content with Wikum compared to more traditional methods to understand the feasibility of the recursive summarization workflow. We recruited 20 participants (mean age 24.9, SD 10.8; 55% female, 45% male) through campus mailing lists and

[1] https://pypi.python.org/pypi/sumy

social media and paid $15 for around one hour of their time. All participants reported reading at least one type of online discussion regularly.

Discussion Data

We were interested in seeing how people would summarize content from different discussion topics and types. Thus we selected three different discussions for our study: the comments on an article from the Atlantic called "Why Women Still Can't Have It All" (Social), a deliberative discussion among members of an academic department about a controversial political event involving their university (Political), and a discussion from the "Explain It Like I'm Five" subreddit seeking to understand a major scientific discovery (Science). Each of these discussions was among the most popular of its category, received many comments from its respective community, and is deeply threaded with many subdiscussions. For the purpose of our study, we pruned the discussions for each condition to roughly equal sizes (removing some of the top level posts and all their replies), aiming for 7,000-8,000 total words or 35-40 minutes of reading given an average reading speed of 200 words per minute [46]. In the end, Social had 84 comments comprising 7,532 total words with the deepest comment 15 levels deep; Political had 67 comments of 7,415 words, with a maximum depth of 14 levels; and Science had 104 comments, 7,375 words, and a maximum depth of 10 levels.

Experiment Design

There were three discussion types, as described earlier, and two system conditions. One system was Wikum, while the control condition was a Google Doc containing the raw discussion text. The text was indented up to 4 levels to indicate threading and then flattened at the 4th level for readability. Google Docs was chosen as a decent approximation to wiki environments. Track changes were turned on to distinguish summaries from original comments so that editors could see each other's work and any text that was deleted by a previous editor. Both conditions included metadata: poster username, number of upvotes, and a unique ID for each comment.

We created three groups and randomly assigned participants to one of them. Each group worked on summarizing two different discussions, one in Wikum and one in the Google Doc, with order counterbalanced. Thus at the end of the study, the three groups produced 3 Wikum summaries and 3 Google Doc summaries, with 2 summaries created per discussion. We chose this experiment design so that we could both compare Wikum versus Google Doc summaries from the same discussion, which controls for that topic of discussion, as well as summaries from the same group, which controls for individual differences in writing ability.

Procedure

User studies were one-on-one, in person, and conducted over a period of two weeks. After completing a short interview and survey about their habits related to online discussions, participants were asked to perform two tasks, limited to 20 minutes each. The goal of each task was to advance the collaborative summarization of one of the two conditions they were assigned, so that at the end, there is a summary of the entire discussion at 250 words or less (half a page). We asked users to work for 20 minutes and no more. Rather than assessing the "natural duration" of an individual's work, we wished to evaluate the *total work* required for summarization, which will likely be distributed among a large number of participants. We kept the time to 20 minutes per task so that each user study would take an hour.

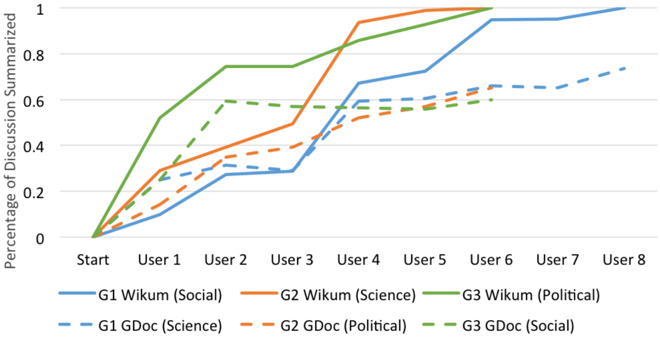


Figure 4: Amount of work completed by each successive user in the Summarization stage, by group. Each user amounts to 20 minutes of working time. All Wikum summaries were completed while none of the Google Doc summaries were finished, even with the same group of users editing both.

In the Wikum condition, users were first given a 5 minute tutorial on the interface. During the task, we did not give users any particular direction but let them spend their 20 minutes working on what they preferred. In the Google Doc condition, we likewise did not provide directions to users on how to summarize the content. We allowed users to write summaries how and wherever they liked but also encouraged users to be consistent and somehow indicate what was left to summarize to future user study participants. After completing each of the tasks, users filled out surveys on their perceived task load [22]. After both tasks were completed, they filled out a survey comparing the systems and answered some open-ended questions about their experience.

Results

Summaries were completed faster in Wikum than Google Docs by the same group. For each user study condition, we computed the *initial text size*—the number of words in the unsummarized comments plus number of words in the summaries—both at the start of the user task and after its completion. The difference tells us by how many words the user was able to shrink the total amount of initial text. Which comments had been summarized was easily defined in Wikum. In the case of Google Docs, we asked users to delineate comments they had summarized in the document, such as using strikeout or marking it "done". We declared a discussion to be fully summarized at the point where the amount of unsummarized content (comments and top-level summaries) totaled 250 words or less. Thus, at the start of our user study, all discussions are at 0% completion, and they reach 100% completion when enough original comments have been summarized so that there are only 250 words to read at the outset.

In Figure 4, we show the productivity of the different groups over the course of the study. As can be seen, each group had overall forward progress towards completion in both system conditions but the Wikum condition overall was faster. In total, two Wikum summaries each took a total of 120 minutes, while one took 160, to be completed. The average summarization rate (words reduced per minute) in Wikum was 51.9 while in Google Docs it was 36.3. Thus, in each of the groups, the Wikum summarization of the discussion was completed while the Google Doc summary was still not complete. We chose to stop subjects working on *both* tasks after each Wikum summary was completed because we wanted to use our other user study participants to provide feedback on the Wikum summary qualities as opposed to spending all their study time finishing the Google Docs summaries.

Comparing the 52 word-per-minute Wikum summarization rate with the 200 word-per-minute reading rate we cited earlier shows that summarization is a rapid activity that would demand only a small fraction of the total person-hours devoted to reading a popular discussion.

Users were reluctant to edit others' summaries in both conditions. In the Google Doc condition, 12/20 users chose to only append to an ever-growing single summary that quickly became longer than the 250-word maximum we set. Out of the remaining 8 users, 6 users wrote their summaries interleaved in the comments but did not delete or edit any existing summaries. If users mostly added to summaries and did not delete anything, this would make full summarization impossible since eventually the summary will be larger than the remaining comments. Indeed, as more users participated, we saw overall progress in the Google Doc condition shrink and even plateau in some of the groups, as Figure 4 indicates. However, this decline was avoided in Wikum, perhaps because recursive summarization has users summarize other people's summaries *without* destroying their work.

Users spent more time reading in the Google Doc condition. Perhaps as a result of ever-growing summaries in Google Docs, we noticed in the later Google Doc tasks that most users spent almost all the time reading instead of summarizing. As more people edited the document, they spent more time reading the existing summary to determine what was covered, skimming through the comments to find unsummarized content, and figuring out how to incorporate their findings back into the summary. One editor said "*Using the Wikum was so much easier...I knew what people had done...With the Google Doc it was this massive 40 page document. I got lost on what people had summarized and what needed to be summarized.*" Some editors did not bother to read previous summaries and then accidentally summarized portions that had already been summarized. Like Google Docs, wikis also lack this kind of scaffolding for summarization. However, some of these issues might potentially be mitigated with a more defined style guide or set of instructions.

	Group 1	Group 2	Group 3
Wikum	1037 (Social)	1310 (Science)	497 (Political)
GDoc	769 (Science)	1073 (Political)	771 (Social)

Table 1: Total number of summary words written by users.

	Social-G1	Science-G2	Political-G3
Summary Nodes	13	20	6
Citations	25	36	4
Quotes	0	7	0
Tags	6	1	5

Table 2: Total number of times each item was used or created in each of the three Wikum summaries.

Users overall wrote more summary text in the Wikum condition. Perhaps as a result of needing to spend less time coordinating other people's edits in the Wikum condition, users overall wrote more in the Wikum condition, as can be seen in Table 1. Though the amount of time spent and the people were kept constant per group, users overall wrote 2,844 words in Wikum versus 2,613 words in Google Docs. As described in the earlier Workflow Efficiency section, this additional summarization did not add much work compared to the 7-8,000 words in the original discussion. In the case of Group 3, the one group where Wikum users wrote less, the Wikum condition had one early participant who chose to summarize a large subthread in one summary. As readers complained about this in the second study, this suggests that in the future we should only allow editors to summarize limited chunks of discussion at a time.

In the case of summarization, more may not always be better. A thousand words of summary is around two pages long, which may be more than someone is willing to read. However, because of recursive summarization in the Wikum case, users can read a 250-word summary of the entire discussion and drill in to get more detailed summaries.

Earlier editors set the norms for later editors in Wikum. We noticed during the user study that the decisions made by early editors in Wikum, such as to use citations or quotes, set the norms for future editors, echoing prior work on norm setting in communities [25]. This led to different styles of summarization emerging in different groups. For instance, early use of citations and quotes led to more use of these features in the SCIENCE Wikum summaries, while it was not used at all by early POLITICAL editors (Table 2). The same was true for the case of adding tags. In the future, this could be more scaffolded, for instance by requiring some number of citations per number of comments being summarized.

The convergence of norms happened to a lesser extent in the Google Doc conditions. For instance, people would use different ways of signaling they finished summarizing a comment in the same document. Some users also chose to write their summary of a particular sub-discussion interleaved among the comments even if others had been contributing to a single summary at the top of the document. Later contributors tended to do this as the single summary got more unwieldy, and unsummarized comments were further from the summary at the top of the page.

Editors made use of the citation and quoting features. Many users chose to add citations in the summaries (Table 2). Several users liked the ability to cite, saying: "*The way in which you can cite paragraphs and posts is very useful to have that kind of chain of custody, like from where does this information come from?*" However, the quoting feature was used less often, possibly because it was less discoverable, as one needed to drag-and-select text before a "Quote" button showed. In the future, we could add "Quote" buttons next to highlighted sentences. Some editors used quoting and citing as a way to minimize editorializing and deflect lack of understanding of the content: "*Obviously someone who has a physics background would be better over me. Me summarizing this comment, I don't know if I would trust me. That's why I tried to quote a lot and really cite what was going on.*" The same user went on to say, "*...People might only read my summary, they might not read the actual comments, so I felt pressure to make sure you've accurately summarized the comment.*" For her, citing and quoting was also a way to point readers to original content and to also self-check that she was summarizing the comments faithfully.

Users reported that summarizing content they disagreed with took more effort. Some users expressed frustration with comments they disliked, with one editor saying, "*What I really wanted to be like was, this comment is stupid because it said this, rather than writing an unbiased thing. I think some of my summaries were a little snarky.*" A different editor mentioned working harder but also that she was more interested: "*It was more interesting to summarize comments that I disagreed with because it requires you to try to understand their point of view as much as possible...I already know my own point of view.*" Reflection and learning gained from summarizing other people's opinions [29] could be an additional side benefit of Wikum. As in Wikipedia, there may be value in educating editors about maintaining a so-called *Neutral Point of View (NPOV)* during summarization work [34].

Overall feedback on summarization. Users overall felt that the recursive summarization process helped to break the task down to something manageable, with one editor saying "*A lot of times I would look at a comment and all its sub-comments and be like, well I can't summarize all that, it's really overwhelming. But then I was able to drill down into the sub-sub-comments and...get the whole comment [subtree] and sub-comments into my head at the same time, write a summary, and then go a level up.*" From the post-study survey, users indicated that they preferred conducting summarizing using Wikum over Google Docs (t=3.02, p<0.01). Users also found Wikum easier to use. Survey results related to task load [22] revealed a significant difference when it came to physical demand, with Wikum overall causing lower physical demand (t=2.07, p=0.05, paired t-test). This may be because many users complained about needing to scroll more in the Google Doc condition. Likewise, Wikum showed lower temporal demand (feeling hurried or rushed during the task) (t=3.11, p<0.01), possibly because it look less time to get started editing in Wikum as opposed to Google Docs. Editors in Wikum also self-reported higher performance on the task (t=2.37, p<0.05).

Study 2: Reading and Exploration

In the second part of the user study, our goal was to assess whether a Wikum summary tree is a useful tool for quickly getting an overview of a discussion. We recruited 13 more participants (mean age 28.0, SD 9.7, 72.2% male, 27.8% female) via the same methods described in the previous stage. As before, all participants said they read at least one type of online discussion regularly. Participants were compensated $10 for around 40 minutes of their time.

Experiment Design

Before seeing any summaries of the discussion, the first author of this paper read over the three discussions and extracted a list of main points made in each. Care was taken to include points made throughout the discussion including in subthreads that were deeply nested. As we only showed editors a subset of the original discussion in Study 1, the author also looked over the comments that were pruned from the original discussion in order to come up with another list of points that were not in the study, but that could plausibly have been.

We designed a 2-factor user study where each participant was given three tasks, each limited to 20 minutes. For each task, the participant was given one of the three discussions and one of three interface conditions. One condition was the Wikum interface with the embedded summaries that users made in the prior stage. A second condition (DocSummary) was a Google Doc containing the summaries and the original comments also created in the prior stage. Summary text was colored purple, while deleted comments were faded gray. Original comments that had not been processed by the first stage participants were colored black. Summaries were left wherever users placed them in the preceding stage, whether that was at the top of the document or interspersed throughout the discussion. We also provided easier navigation to the different summaries using the Google Docs outline feature. The third condition (NoSummary) was a control, consisting of a Google Doc containing only the raw discussion with no summaries. The assignment of the discussion topics and interface conditions as well as the order was counterbalanced.

Procedure

In each task, the participant was given 10 minutes to try to get an overview of the discussion. During this time, the authors observed how participants chose to explore the discussion in the different interfaces. Then, without the discussion in front of them, they were presented with a list of 12 points, 6 of which had been mentioned in the discussion and 6 of which had not. Participants were not told the number of points that were false. They were asked to select points they remembered being brought up in the discussion. At the end, participants completed a survey about their experience and discussed their experience reading using the different interfaces.

Results

Most explored the Google Doc linearly, while there was a mix of strategies using Wikum. For the NoSummary condition, almost all participants read linearly down the page, with most running out of time before they read even half of the discussion. For the DocSummary condition, most users also read linearly down the page, though some users chose

Conditions	Precision	Recall	F1
Wikum	0.90	0.67	0.78
Google Doc Summary	0.88	0.63	0.72
Google Doc No Summary	0.81	0.58	0.65

Table 3: The results of Study 2 between the three conditions.

to focus on reading the summaries and skip over or skim the comments that were in gray. Others chose to read original comments, even if they already read the summary.

In the Wikum condition, people had a mix of strategies. Several users (5/13) chose to expand the discussion tree fully and read linearly down the discussion on the right, sometimes scrolling past some subthreads, but overall treating the Wikum interface exactly how they would a Google Doc. Others (4/13) chose a breadth-first approach from the root, reading summaries at each level and only expanding summaries when they deemed it necessary. Some users chose to expand everything at the outset but then focus on the summary nodes using the tree visualization, going from the root to the leaves (3/13) or from the leaves to the root (1/13). Many of the users who focused only on the summaries chose to stop reading well before the 10-minute cutoff, suggesting they had already achieved full comprehension.

Users recalled points made in the discussion more accurately in the Wikum condition. From the recall test, as seen in Table 3, Wikum performed slightly better than DocSummary on the measures of precison, accuracy, and F1 score, and both summary conditions performed better than NoSummary. However, none of the differences in scores between the three different conditions yielded a statistically significant difference (with p<0.05), likely due to the small sample size and the variations in topic, quality of summary, order of conditions, and different reading strategies and speeds. Thus, these results suggest that summaries are indeed helpful for getting an overview in a short amount of time, and that users were able to get an overview using Wikum as least as well as using Google Docs. Though the difference between Wikum and Google Docs with summaries was not significant, recall that all users were familiar with the Google Docs interface but had only a few minutes to learn the new Wikum interface. One user said "*A big chunk of the time went into understanding the Wikum interface itself - more than half. If I had seen this interface 5 or 10 times I would be familiar with it.*"

Some people preferred reading linearly while others enjoyed drilling in. The Wikum interfaces defaults to hiding comments underneath a summary. Some people disliked needing to click to open up a summary, saying "*[I would like to] have more control about what I was going to read, as well as look at the scrollbar to know the amount of content ahead of me.*" As a related issue, some people enjoyed the tree visualization, while other people found it overwhelming. While the tree visualization seems a useful feature for editors, it may be less necessary for readers of a summary tree.

People opened summaries to read comments for different reasons. Some people said they would read comments below a summary if it was poorly written or too short because they did not trust it. For instance, one person said "*That's the scientist in me. I need to see, is this comment really saying that? I didn't want the summaries to influence my take.*" Other times, readers actually thought the summary was well written and thus it piqued their curiosity: "*I was more likely to read the individual comments on the good summaries. The summaries went into depth, so I figured there was more discussion there. Good = interesting, so I wanted to learn more.*"

Overall feedback on reading and exploration. When it came to their experience reading and exploring the comments using the different interfaces, users rated Wikum the highest (4.2 on average on a 7-point Likert scale from 0 to 6), with DocSummary second best (3.6), and NoSummary the worst (2.5). The difference between the Wikum and DocSummary was not statistically significant (p<0.05) while the difference between those two conditions and the unsummarized one was significant (Wikum: t=-3.04, p<0.005, DocSummary: t=-3.05, p<0.005). Users were also asked to grade summary quality on a 7-point Likert scale. Overall users felt the Wikum summaries were of higher quality than the Google Doc ones (4.5 versus 3.5 on average respectively), though this difference was also not statistically significant. Thus our results suggest but do not conclude that Wikum provided benefits for readers over the Google Doc, and affirms that summaries are a useful way for readers to get an overview of a discussion.

From post-study interviews, users mentioned that the Wikum summaries were more succinct while Google Doc summaries went on for too long. This is despite the fact that the total text in all the Wikum summaries was actually greater for those users. One user said of the Wikum summaries: " *It felt good on a few comments - it was very noticeable...that there was a large amount of text just swirling around a few simple ideas, and the summary got it simple. Like into a tweet. That was really, really nice. I wish everything could be summarized like that.*" Another user said "*I felt it was helpful for Wikum but not really in the Google Doc. There, there were people rambling...It was kind of a mess. Because the summaries were right there in Wikum and directly related to the comments, [they were] much smaller summaries and a lot more helpful.*" A different user echoed that the Wikum summaries were shorter, and complained that the highest-level Wikum summary was too abstract so that he had to dig deeper to understand portions of the discussion (which Wikum is specifically design to support). This could be related to the preference some people had for reading linearly.

DISCUSSION

Design Implications

During the summarization stage of our user studies, we saw that Google Docs was too underconstrained so there were many opportunities for editors to go astray and set poor norms. However, even though Wikum has more constraints, we realized that some additional scaffolding could guide editors towards creating better summaries while still maintaining Wikum's flexibility. One editor was worried about too much rehashing, saying, "*If you encourage a summary every time you have a parent or child, you'll just have crummy* *summary on top of crummy summary...Trying to encourage only summaries when you have a certain depth or breadth to the tree would go a long way.*" In the other direction, one user chose to summarize a large portion of the discussion at once, producing a low quality summary. Later readers of this summary tree were surprised to find so many comments under that summary. This indicates that there may be an optimal range of discussion size that should be summarized in a recursive summary. Too small and the recursive summaries feel too incremental and repetitive to a reader. Too big and the summaries have poor coverage and hide a great deal of discussion. Wikum could also suggest groups of comments to target for summarization via heuristics or machine learning. These could include the start of a self-contained subthread, a clear shift in topic or participants, or a discussion devolving into arguing.

Another issue that came up was around the difficulty of summarizing opinionated content, especially content the editors disagreed with. Computation techniques in detecting language that is objective versus subjective [53] or determining opinionated or emotional sentences [54] could be a useful addition to a summary editing box to help editors monitor the language they use.

When it came to the reading experience, many readers in the study talked about trust as an important factor while reading the summaries. If they did not trust that the summaries were accurate or had good coverage, they felt they needed to read more of the original content. Distrust of wikis and other crowd-editable content can sometimes be mitigated with design [26]. This was one reason for our emphasis on citations and quoting. Other ways to improve trust could involve showing information such as number of edits, total time spent writing a summary, number of contributors, or percentage of original discussion cited. We could also introduce a form of social moderation, allowing readers to rate summaries on accuracy.

Finally, our study reveals future areas for experimentation with different presentations of the summary tree. Some readers liked the information that the tree visualization provided but others felt it was overwhelming or too disconnected from the text. Some ideas to explore include trying to integrate information that the tree provides directly into the discussion text, such as toggle controls, breadcrumbs, or even simplified subtree thumbnails. Views were also mixed on the preference for an expandable versus linear reading experience, echoing prior work in the hypertext literature around jumping around using links [18, 44]. Unlike a graph-structured hypertext however, which can pose significant navigation challenges [39], Wikum is likely easier to navigate since it is hierarchical. Additionally, one can ignore the expandable nature of Wikum and pre-expand everything, as we saw a few readers do, and read linearly. In the future, we could make this even easier by allowing readers to set how much of the summaries they wish to have autoexpanded upon load.

A Tension Between Summary Goals

From the user studies of both creating and reading summaries, we learned what users perceived was useful about Wikum as well as what they desired in a summary. Some users were interested in getting an overview of the topic of the conversation, with points organized in pros and cons and grouped by topic. Other users saw summaries embedded in the discussion structure as useful signposts for readers to decide whether to go down that particular path to find interesting comments. In the current state of Wikum, earlier contributions to an original discussion work much like signposts, as users are mostly summarizing small groups or threads of discussion. This aligned with what users reported, with one user stating, "*I was trying to summarize it in a way that would make someone looking go, do I bother reading this or not? Just what it's about rather than details, to decide whether to go down it.*" But as most of the original discussion gets summarized until one is only summarizing summaries, users need to begin to organize higher-level concepts into a coherent story.

These two modes suggest slightly different design decisions for both readers and editors. So far, Wikum has avoided edits to the discussion that break the original discussion threading structure. However, this runs into issues when different subthreads far away from each other have a redundant discussion. In that case, it would be useful to be able to merge those two discussions together under a single summary, breaking the discussion structure. As another case, an outline that wants to separately organize pros versus cons would likely break reply threading structure, since many arguments would have pro comments and con comments interleaved. While there are benefits to breaking discussion structure, there are pitfalls as well. For users more interested in following a thread of conversation, it would be important to still be able to see comments in their original context. We noticed in a pilot study that when editors had the ability to move comments and threads to different places in the discussion, they were reluctant to break the original discussion structure out of concern about altering original commentators' intents. This further suggests that future designs should allow the mapping of the summary tree back to the original discussion structure.

This difference in goals also suggests that we consider supporting multiple summaries of overlapping pieces of content for different purposes. This offers opportunities to meet more needs, but also significantly complicates the structure of the summary tree—turning it into a more general hypertext document. This would raise new challenges for navigating, such as deciding which summaries to use.

Who Summarizes?

We can see a system such as Wikum used in a number of different scenarios. For instance, a single individual working to summarize a large discussion could derive benefit from some of the scaffolding and breaking down of summaries, much like the self-sourcing literature envisions [45]. The subtask structuring means the individual need only consider a limited scope of discussion at any one time, so they can summarize without comprehending the entire discussion at once (at which point their finished summary can provide them with that full comprehension).

Wikum could also be used by the small skilled groups of *moderators* already managing many discussion sites. These moderators currently focus on flagging and removing inappropriate content, and may well be interested in Wikum's alternative approach to curation.

Additionally, analogous to *social moderation* we envision contributions by a larger number of community members. After reading a deep thread, readers could summarize the content for future readers. Commentators could be required or encouraged to contribute short summaries of their comment (already common practice in some communities as a "TL;DR") or summarize a back-and-forth conversation in which they just participated. As argued above, only a moderate fraction of users' time need be spent on summarization in order to "keep up" with the arrival of new content.

If any user can edit or add a summary, more sophisticated tools for tracking, observing, and reverting changes are necessary. We may wish to permit multiple users to author competing summaries on a single topic, then support voting to let the community select the best summary. We can also consider the role of the original commentators in the discussion being summarized as they may have more incentives. Other work has chosen to give commentators greater moderation power over the summaries of their comments [29], but in our case they may be overly biased.

Crowd workers who have been tasked to summarize a discussion could also use this summarization workflow. As in the community case, we would need to build in robust spam filtering and verification, processes which have been explored in the literature [6]. Finally, Wikipedia talk pages could be an interesting application of Wikum, as it is a place where people already familiar with wiki editing have lengthy arguments, and editors must make decisions that draw from such arguments. However, there is little structure for organizing the discussions, collaboratively summarizing discussions, or showing the resolutions to newcomers.

LIMITATIONS AND FUTURE WORK

In our user study, we only examined discussions that had a threaded structure. Also there was not an overwhelming number of replies to any one comment from the discussions, so the clustering feature was not heavily used. While Wikum can be used in a non-threaded discussion by clustering and grouping related comments, threading certainly provided a powerful grouping heuristic. Future work should study the use of Wikum on non-threaded discussions. More techniques could be added to make the process of finding related comments within a large space easier for editors. For instance, many non-threaded discussions actually have implicit threads of conversation as users reply to each other. Prior work on predicting reply structure from examining the text and chronology of unthreaded discussion could be useful here [52].

As we evaluated Wikum using a lab study, we do not yet have empirical evidence about how such a system would work in the wild. There are other use cases mentioned previously, such as using a paid crowdworker platform, that also warrant additional study. While our lab study helped to understand how editors and readers would use Wikum and suggested design directions, an in-the-wild study would clarify new aspects, such as determining possible incentives towards participation and dealing with bad actors.

Wikum can be used to summarize a static discussion but does not currently support incorporating new comments. One interesting future line of work would be adapting Wikum to ongoing discussions. Thus a subthread that has been summarized may need to be updated when a user contributes a new comment to the discussion. Also, we could consider how people may want to "reply to" previously written summaries.

We incorporated automatic summarization techniques to help editors skim comments. There are other opportunities to incorporate machine learning. Techniques such aspect summarization of product reviews [32] could be repurposed towards grouping comments and providing default summaries of those groups to build upon. Users can also provide training data in a human-in-the-loop process to improve the quality of models. For instance, could machine learning help determine where to segment the discussion into discrete subparts? The data produced by this system could also be used to better build and train automatic summarization techniques for discussions.

CONCLUSION

In this work, we designed, developed, and evaluated a workflow called recursive summarization for summarizing discussions and a system called Wikum that bridges discussion forums and wiki summaries. By bridging the two mediums of wiki and forum through embedding wiki summaries into a discussion structure at varying levels, we provide a process for editors to summarize portions of discussion and build upon each other's work. We also explore design decisions around an interface for readers to interactively explore a discussion, drilling deeper into a summary to get more information. From our evaluations, we found that editors created summaries productively using the Wikum interface and that the created embedded summaries were effective for helping readers get an overview of the discussion.

REFERENCES

1. Mark S Ackerman, Juri Dachtera, Volkmar Pipek, and Volker Wulf. 2013. Sharing knowledge and expertise: The CSCW view of knowledge management. *Computer Supported Cooperative Work (CSCW)* 22, 4-6 (2013), 531–573.

2. Mark S Ackerman, Anne Swenson, Stephen Cotterill, and Kurtis DeMaagd. 2003. I-DIAG: from community discussion to knowledge distillation. In *Communities and Technologies*. Springer, 307–325.

3. Paul André, Aniket Kittur, and Steven P Dow. 2014a. Crowd synthesis: Extracting categories and clusters from complex data. In *Proceedings of the 17th ACM conference on Computer Supported Cooperative Work & Social Computing*. ACM, 989–998.

4. Paul André, Robert E Kraut, and Aniket Kittur. 2014b. Effects of simultaneous and sequential work structures on distributed collaborative interdependent tasks. In

Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 139–148.

5. Paul André, Haoqi Zhang, Juho Kim, Lydia Chilton, Steven P Dow, and Robert C Miller. 2013. Community clustering: Leveraging an academic crowd to form coherent conference sessions. In *First AAAI Conference on Human Computation and Crowdsourcing*.

6. Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. 2015. Soylent: a word processor with a crowd inside. *Commun. ACM* 58, 8 (2015), 85–94.

7. Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2014. How Community Feedback Shapes User Behavior. In *Eighth International AAAI Conference on Weblogs and Social Media*.

8. Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. 2015. Antisocial behavior in online discussion communities. In *Ninth International AAAI Conference on Weblogs and Social Media*.

9. Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. 2013. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1999–2008.

10. Justin Cranshaw and Aniket Kittur. 2011. The polymath project: lessons from a successful online collaboration in mathematics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1865–1874.

11. Kate Crawford and Tarleton Gillespie. 2014. What is a flag for? Social media reporting tools and the vocabulary of complaint. *New Media & Society* (2014).

12. Kushal Dave, Martin Wattenberg, and Michael Muller. 2004. Flash forums and forumReader: navigating a new kind of large-scale online discussion. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. ACM, 232–241.

13. Judith Donath, Karrie Karahalios, and Fernanda Viégas. 1999. Visualizing conversation. *Journal of Computer-Mediated Communication* 4, 4 (1999), 0–0.

14. Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* (2004), 457–479.

15. Siamak Faridani, Ephrat Bitton, Kimiko Ryokai, and Ken Goldberg. 2010. Opinion space: a scalable tool for browsing online comments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1175–1184.

16. Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 340–348.

17. Eric Gilbert. 2013. Widespread underprovision on reddit. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*. ACM, 803–808.

18. Sallie Gordon, Jill Gustavel, Jana Moore, and Jon Hankey. 1988. The effects of hypertext on reader knowledge representation. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 32. SAGE Publications, 296–300.

19. Timothy Gowers and Michael Nielsen. 2009. Massively collaborative mathematics. *Nature* 461, 7266 (2009), 879–881.

20. Nathan Hahn, Joseph Chang, Ji Eun Kim, and Aniket Kittur. 2016. The Knowledge Accelerator: Big Picture Thinking in Small Pieces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2258–2270.

21. Derek L Hansen, Mark S Ackerman, Paul J Resnick, and Sean Munson. 2007. Virtual community maintenance with a collaborative repository. *Proceedings of the American Society for Information Science and Technology* 44, 1 (2007), 1–20.

22. Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in Psychology* 52 (1988), 139–183.

23. Enamul Hoque and Giuseppe Carenini. 2015. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM, 169–180.

24. Bernard Kerr. 2003. Thread arcs: An email thread visualization. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*. IEEE, 211–218.

25. Sara Kiesler, Robert Kraut, Paul Resnick, and Aniket Kittur. 2012. Regulating behavior in online communities. *Building Successful Online Communities: Evidence-Based Social Design. MIT Press, Cambridge, MA* (2012).

26. Aniket Kittur, Bongwon Suh, and Ed H Chi. 2008. Can you ever trust a wiki?: impacting perceived trustworthiness in wikipedia. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*. ACM, 477–480.

27. Aniket Kittur, Bongwon Suh, Bryan A Pendleton, and Ed H Chi. 2007. He says, she says: conflict and coordination in Wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 453–462.

28. Travis Kriplean, Jonathan Morgan, Deen Freelon, Alan Borning, and Lance Bennett. 2012a. Supporting reflective public thought with considerit. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*. ACM, 265–274.

29. Travis Kriplean, Michael Toomim, Jonathan Morgan, Alan Borning, and Andrew Ko. 2012b. Is this what you meant?: promoting listening on the web with reflect. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1559–1568.

30. Cliff Lampe and Paul Resnick. 2004. Slash (dot) and burn: distributed moderation in a large online conversation space. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 543–550.

31. David Laniado, Riccardo Tasso, Yana Volkovich, and Andreas Kaltenbrunner. 2011. When the wikipedians talk: Network and tree structure of wikipedia discussion pages.. In *Fifth International AAAI Conference on Weblogs and Social Media*.

32. Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of the 18th international Conference on World Wide Web*. ACM, 131–140.

33. Kurt Luther, Nathan Hahn, Steven P Dow, and Aniket Kittur. 2015. Crowdlines: Supporting Synthesis of Diverse Information Sources through Crowdsourced Outlines. In *Third AAAI Conference on Human Computation and Crowdsourcing*.

34. Sorin Adam Matei and Caius Dobrescu. 2010. Wikipedia's Neutral Point of View: Settling Conflict through Ambiguity. *The Information Society* 27, 1 (2010), 40–51.

35. Gilad Mishne, David Carmel, Ronny Lempel, and others. 2005. Blocking Blog Spam with Language Model Disagreement.. In *AIRWeb*, Vol. 5. 1–6.

36. Kevin K Nam and Mark S Ackerman. 2007. Arkose: reusing informal information from online discussions. In *Proceedings of the 2007 International ACM Conference on Supporting Group Work*. ACM, 137–146.

37. Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining Text Data*. Springer, 43–76.

38. Beth Simone Noveck. 2009. *Wiki government: how technology can make government better, democracy stronger, and citizens more powerful*. Brookings Institution Press.

39. Malcolm Otter and Hilary Johnson. 2000. Lost in hyperspace: metrics and mental models. *Interacting with Computers* 13, 1 (2000), 1–40.

40. Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you*. Penguin UK.

41. Quora. 2016. Quora Answer Wikis. (2016). Retrieved August 5, 2016 from https://www.quora.com/topic/Answer-Wikis-Quora-content.

42. Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. 2004. Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*. Association for Computational Linguistics, 105–108.

43. Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* (2015).

44. Amy Shapiro and Dale Niederhauser. 2004. Learning from hypertext: Research issues and findings. *Handbook of Research on Educational Communications and Technology* 2 (2004), 605–620.

45. Jaime Teevan, Daniel J Liebling, and Walter S Lasecki. 2014. Selfsourcing personal tasks. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2527–2532.

46. Susanne Trauzettel-Klosinski and Klaus Dietz. 2012. Standardized Assessment of Reading Performance: The New International Reading Speed Texts IReSTStandardized Assessment of Reading Performance. *Investigative Ophthalmology & Visual Science* 53, 9 (2012), 5452–5461.

47. Bogdan Vasilescu, Alexander Serebrenik, Prem Devanbu, and Vladimir Filkov. 2014. How social Q&A sites are changing knowledge sharing in open source software communities. In *Proceedings of the 17th ACM conference on Computer Supported Cooperative Work & Social Computing*. ACM, 342–354.

48. Gina Danielle Venolia and Carman Neustaedter. 2003. Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 361–368.

49. Vasilis Verroios and Michael S Bernstein. 2014. Context trees: Crowdsourcing global understanding from local views. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

50. Fernanda B Viégas, Scott Golder, and Judith Donath. 2006. Visualizing email content: portraying relationships from conversational histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 979–988.

51. Fernanda B Viégas, Martin Wattenberg, Jesse Kriss, and Frank Van Ham. 2007. Talk before you type: Coordination in Wikipedia. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*. IEEE, 78–78.

52. Hongning Wang, Chi Wang, ChengXiang Zhai, and Jiawei Han. 2011. Learning online discussion structures by conditional random fields. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 435–444.

53. Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 486–497.

54. Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* 39, 2-3 (2005), 165–210.