ConVisIT: Interactive Topic Modeling for Exploring Asynchronous Online Conversations

Enamul Hoque and Giuseppe Carenini
Department of Computer Science
University of British Columbia, Canada
{enamul, carenini}@cs.ubc.ca

ABSTRACT

In the last decade, there has been an exponential growth of asynchronous online conversations thanks to the rise of social media. Analyzing and gaining insights from such conversations can be quite challenging for a user, especially when the discussion becomes very long. A promising solution to this problem is topic modeling, since it may help the user to quickly understand what was discussed in the long conversation and explore the comments of interest. However, the results of topic modeling can be noisy and may not match the users current information needs. To address this problem, we propose a novel topic modeling system for asynchronous conversations that revises the model on the fly based on user's feedback. We then integrate this system with interactive visualization techniques to support the user in exploring long conversations, as well as revising the topic model when the current results are not adequate to fulfill her information needs. An evaluation with real users illustrates the potential benefits of our approach for exploring conversations, when compared to both a traditional interface as well as an interactive visual interface that does not support human-in-the-loop topic model.

Author Keywords

Interactive topic modeling; asynchronous conversation; text visualization; computer mediated communication;

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces
I.2.7 Natural Language Processing: Text analysis

INTRODUCTION

With the proliferation of Web-based social media, asynchronous conversations have become very common for supporting online communication and collaboration. An asynchronous conversation such as a blog may start with a news article or an editorial opinion, and later generate a long and complex thread as comments are added by the participants [2]. Consider a reader who opens a blog conversation about Obama's healthcare policy. She wants to know why people are supporting or opposing 'ObamaCare'. However, since the conversation is quite long (e.g., 100 comments) and some other related discussion topics like 'student loan' and 'job recession' were introduced, she finds it hard to keep track of the comments about 'ObamaCare', which end up being buried in the long thread. This and similar scenarios often lead to information overload, where the reader gets overwhelmed, starts to skip comments, and eventually leaves the conversation without satisfying her information needs [17].

Integrating Natural Language Processing (NLP) and Information Visualization (InfoVis) techniques has been proposed as a promising solution to this and similar problems [34, 36]. It has been found that when the results of topic modeling and sentiment analysis are presented in a visual interface, they can support most of the reading tasks identified in the domain of asynchronous conversations [15]. For instance, topic modeling can group sentences into different semantically coherent clusters and then assign descriptive keyphrases to each of them [18], such as 'ObamaCare' and 'student loan'. When these topics are presented within an interface, the user can select the one(s) she is interested in (e.g., 'ObamaCare') and then quickly navigate through its related comments. In addition, if information about sentiment (e.g., positive vs negative opinions) is visually encoded along with the topics, the user can assess what comments were in favor or against a particular issue.

While topic models can provide attractive solution in understanding large conversations, they may not be always useful to the end users [2, 15, 16]. This could be due to three different reasons. First, sometimes the current information seeking tasks may require a topic model at a different level of granularity, e.g., if the user needs more specific information about 'ObamaCare' she might be interested in exploring its potential sub-topics such as 'health insurance', 'healthcare cost' and 'drugs'. Second, the interpretation of topics may vary among users according to their expertise and mental model. In fact, in a topic annotation study humans sometimes disagreed on the number of topics and on the assignment of sentences to topic clusters [18]. For instance, for one of the conversations from their corpora, one annotator produced 22 topics, while another annotator reported only 8 topics. Finally, in some cases the results of topic modeling can be simply incorrect, in the sense that the generated topics would not make sense to any user [5, 18]. For example, two semantically different topics 'Obama health policy' and 'job reces-

IUI'15, March 29 - April 01, 2015, Atlanta, GA, USA.
Copyright 2015 ACM 978-1-4503-3306-1/15/03$15.00
http://dx.doi.org/10.1145/2678025.2701370

sion' might be wrongly grouped together with the misleading topic 'Obama recession'.

In this paper, we present an interface in which the user can explore a conversation by relying on topics that make sense to her, that are semantically coherent and match her expertise, mental model and current task. Our solution is to support the user in revising the topic model, while she is exploring the conversation. To achieve this, user feedback is incorporated within the topic modeling loop in real-time through a visual interface, named ConVisIT. The interface is designed by extending ConVis [15], which was developed to satisfy a comprehensive set of the user requirements in the domain of asynchronous conversation. ConVis presented topics, sentiment and a set of metadata to support the user in exploring and navigating through the conversation. However, a preliminary evaluation of ConVis suggested that the user could benefit from a greater control over the topic modeling process [15]. This was particularly evident from the interviews and observational data, where users expressed a pressing need for enhancing their ability to revise the topic model according to their own information needs. Motivated by this experience, we aim to support users in taking an even more active role in exploring conversations through an interactive topic modeling approach.

Figure 1 illustrates our proposed topic modeling framework. Given an asynchronous conversation (e.g., blog), the system generates the initial set of topics, which are presented in the visual interface along with other conversational data. The interface then supports the user in exploring the conversation. However, whenever the user realizes that the current topic model is not helping her, she can provide topic revision feedback to the system through interactions. Subsequently, the system updates the topic model and the new results are presented in the interface.

The primary contributions of our work are three-fold:

1) A novel interactive topic modeling system specifically designed for asynchronous conversation. Existing systems (e.g., [4, 16, 22]) are mainly devised for generic documents without considering the unique features of conversations. On the contrary, we analyze the information seeking tasks in our target domain to select a minimum set of topic revision operations that are critical to the user. Then, we devise computational methods for each of these operations to be performed by the system.

2) ConVisIT, a visual interface which provides a set of interactive features that allow the user to revise the current topic model. In response, the interface updates and re-organizes the modified topics by means of intuitive animations, so that the user can better fulfill her information needs.

3) A user study to understand how the visual interfaces for exploring conversations (ConVis and ConVisIT) may influence user performance and subjective opinions when compared to more traditional interfaces. This evaluation also provides insights into the potential advantages and relative tradeoffs of interactive topic visualization approaches (i.e., ConVis vs. ConVisIT).

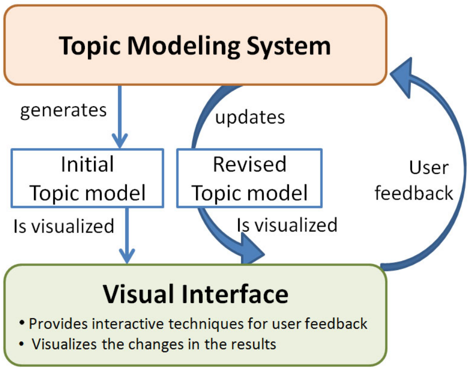


Figure 1. Interactive topic modeling framework for exploring asynchronous conversation.

RELATED WORKS

Visualizing asynchronous conversations

Earlier works on visualizing asynchronous conversations primarily investigated how to reveal the thread structure of a conversation using tree visualization techniques, such as using a mixed-model visualization to show both chronological sequence and reply relationships [33], thumbnail metaphor using a sequence of rectangles [35, 20], and radial tree layout [26]. However, such visualizations did not focus on analyzing the actual content (i.e., the text) of the conversations, therefore they are arguably inadequate to support users in most of the information seeking tasks identified in [15].

Recently, there has been more focus on performing content analysis of the conversations, such as identifying primary themes (or topics) within conversations [28, 9], and visualizing the content evolution over time [34, 36, 10]. Commonly, these approaches use probabilistic topic models such as Latent Dirichlet Allocation (LDA), where topics are defined as distributions of words and documents are represented as mixture of topics. For instance, the TIARA system applies the Themeriver metaphor [13], where each layer in the graph represents a topic and the keywords of each topic are distributed along time. From the height of each topic and its content distributed over time, the user can see the topic evolution. The underlying temporal topic segmentation of TIARA was further improved by applying a set of semantic, temporal, and visualization constraints simultaneously [25]. More recently, a hierarchical version of the Themeriver metaphor was also designed to explore the temporal changes of topics [10], by generating a topic tree based on computing the distance between the probability distributions of topics.

Although the aforementioned approaches do combine text analysis with InfoVis methods to support the user in making sense of conversational data, they suffer from two major limitations. First, often the visual encodings and interactive techniques are not derived from task and data abstractions based on a detailed analysis of specific user needs and requirements in the target domains, which has been identified as crucial according to the InfoVis design study methodology literature [23, 29]. Second, the text analysis methods employed by these approaches are not designed to exploit the specific characteristics of asynchronous conversations (e.g., use of quotation), while it has been shown that topic models are more accurate when these specific characteristics are taken into account [18]. In order to address these two limitations, ConVisIT was designed based on data and tasks analysis specific to the domain of asynchronous conversation [15], and by applying a topic modeling approach that takes advantage of the conversational feature [18].

Human-in-the-loop topic model

Since system-generated topic models can be noisy, some recent work investigate how user supervision can be introduced to improve the results. These works mainly focused on answering two research questions: 1) How to incorporate user feedback in the topic model? 2) How an interface can support the user in expressing such feedback? To answer the first question- in the dominant LDA topic modeling framework, the original unsupervised LDA method was modified to allow the introduction of human supervision [1, 16, 27]. For instance, Andrzejewski et al. incorporates user's domain knowledge in LDA by adding constraints in the form of must-link (enforces that sets of words must appear together in the same topic) and cannot-link (enforces that sets of words must be in different topics) using Dirichlet forest prior [1]. However, this method requires to rerun Gibbs sampling from scratch after a set of constraints is added, leading to high latency. Since, such latency is undesirable for real-time interactions, [16] proposes an efficient inference mechanism that aims to minimize user's waiting time. Unfortunately, all these approaches were designed for generic documents. In contrast, we devise a new interactive topic modeling framework that is designed to take advantage of conversational features.

The question of how a visual interface can support the user in expressing her feedback has been addressed in [6, 4, 22]. Chuang et al. extend Termite [7], which visualizes the term-topic distributions produced by LDA topic models, and allows the user to revise the model by clicking on words to promote or demote a words usage in a topic [6]. Similarly, [22] visualizes topic modeling results from LDA, and allows the user to interactively manipulate the topical keyword weights and to merge/split topic clusters. More recently, user feedback was incorporated through a scatter plot visualization, that steers a semi-supervised non-negative matrix factorization (NMF) method for topic modeling [4]. The authors show that the NMF-based approach has faster empirical convergence and offers more consistency in the results over traditional LDA-based approach. They visually present each topic cluster and then allows the user to directly manipulate the documents and keywords within each cluster to specify topic revisions. A fundamental limitation of most of these works is that the visual interfaces for interactive topic model were not evaluated with real users. Therefore, a set of critical research questions remained unanswered. For instance, would users be really interested in performing all the operations provided with such a complex interactive visualizations? What operations are actually useful to the users for performing exploratory tasks in a specific domain? To answer these questions, we applied a systematic approach, where we first devised a set of topic revision operations which are most useful according to our tasks analysis, and then performed a user study to measure the utility of these operations.

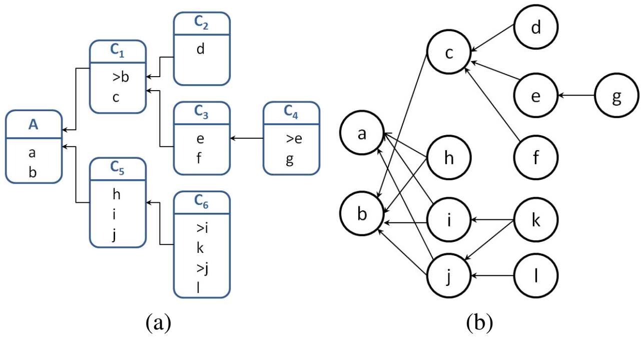


Figure 2. a) Reply-to relationships between the initial post A and the comments $C_1, C_2, ..., C_6$ (left). Here, '>' represents the quotation mark and each lowecase letter corresponds to a text fragment that may comprise one or more sentences. b) the corresponding FQG (right) where each node represents a text fragment and the edges represents replying relationships between fragments.

INTERACTIVE TOPIC MODELING SYSTEM

As illustrated in Figure 1, interactive topic modeling system performs two primary functions: 1) generating the initial topic model, 2) revising the topic model based on user feedback. Here, we discuss these two functions in details:

The initial topic model

Initially, we group the sentences of a conversation into a number of topical clusters (*segmentation*) and label each cluster by generating semantically meaningful descriptor (*labeling*). We adopt a novel topic modeling approach for asynchronous conversations that captures finer level conversation structure in the form of a graph called Fragment Quotation Graph (FQG) [18]. We extract all the distinct fragments (both new and quoted) within a conversation as the nodes of the FQG (see Figure 2). Then the edges are created to represent the replying relationship between fragments. If a comment does not contain any quotation, then its fragments are linked to the fragments of the comment to which it replies, capturing the original reply-to relation. Here we briefly describe how topic segmentation and labeling can take advantage of the FQG, interested readers are directed to [18] for a more detailed description.

Topic Segmentation

First, a Lexical Cohesion-based Segmenter (LCSeg) [12] is applied to find the segmentation boundary within each path (from roots to the leaves) of a FQG (see Figure 2). Then an undirected weighted graph $G(V, E)$ is constructed, where each node in V represents a sentence within the conversation, and each edge $w(x, y)$ in E represents the number of segments on different paths in which the two sentences appear together. If x and y do not appear together in any segment, their cosine similarity (always between 0 and 1) is used as edge weights. By construction, any subgraph of G whose nodes are strongly connected represent a set of sentences that should belong to the same topical segment.

To identify subgraphs whose nodes are strongly connected, a k-way min-cut graph partitioning algorithm is applied on the graph $G(V, E)$ with the normalized cut (Ncut) criteria.

No	Operation	Why?	Criteria			Reference
			Task relevancy	Topic Model relevancy	Redundancy	
1	Split a topic	This topic is too generic	high	yes	no	[4, 22]
2	Merge by joining	These topics are talking about similar things	high	yes	no	[4, 22]
3	Merge by absorption	A group of sentences are wrongly clustered into a different topic	high	yes	no	[4]
4	Split by keyword	This keyword should be separated into a new topic	medium	yes	yes	[4]
5	Change the overall granularity level of topics	Too few topics/ too many specific topics are generated	medium	yes	yes	-
6	Remove the topic from the display	This topic does not make any sense (i.e., off-topic)	low	yes	yes	[22]
7	Assign a label for this topic	The current label of this topic does not represent the actual topic	low	yes	yes	[11]
8	Increase the weight of this keyphrase	This keyphrase should be included in the topic label list	low	yes	yes	[11]
9	Apply must-link constraint	Those words **must be** in the same topic	low	no	no	[1, 16]
10	Apply cannot-link constraint	Those words **must not be** in the same topic	low	no	no	[1, 16]
11	Change keyword weights	This keyword is more related to the topic	low	no	yes	[4, 22]

Table 1. Different possible topic revision operations.

Since Ncut is a NP-complete problem, an approximate solution is found following an efficient method proposed by Shi and Malik [30]. At the end of this process, each sentence of the conversation is assigned to one of the topical segments.

Topic Labeling

Topic labeling takes the segmented conversation as input, and generates keyphrases to describe each topic in the conversation. The conversation is first tokenized and a syntactic filter is applied to select only nouns and adjectives from the text. Then a novel graph based ranking model is applied that exploits two conversational features: information from the leading sentences and the FQG. For this purpose, a heterogeneous network is constructed that consists of three subgraphs: the FQG; the word co-occurrence graph (Wc) generated from computing the co-occurrence of each word with respect to the leading sentence and the topical segment; and a bipartite graph that ties these two graphs together. A co-ranking method [37] is then applied to this heterogeneous network to generate the ranked list of words for each topic. The top-M selected keywords from the ranked list are then marked in the text, and the sequences of adjacent keywords are collapsed into keyphrases. Finally, to achieve broad coverage of the topic, the Maximum Marginal Relevance (MMR) criterion is used to select the labels that are most relevant, but not redundant.

Corpora and preprocessing

For our analysis and experiments, we used twenty blog conversations from the Slashdot [1] corpora [18]. For each conversation, we generated a topic model comprising of x topics, where x represents the average number of topics produced by the annotators for that conversation. To determine the sentiment polarity of each sentence of the conversation we used SoCAL [32], which has been shown to work well on user-generated content. We defined five different polarity intervals (-2 to +2), and for each comment we counted how many sentences fall in any of these polarity intervals to compute the polarity distribution for that comment.

Interactive topic revisions

Although the initial topic model is more accurate than models generated by traditional methods for non-conversational text [18], still the extracted topics may not always match the user's information needs. Depending on user's mental model and current tasks, the topic modeling results may not be adequate. For instance, more specific topics may be more useful in some cases, while more generic ones in other cases. Therefore, we incorporate a set of topic revision operations by which users can iteratively modify the initial topic model to better fulfill their information needs.

Since it may take some effort from the users to express different topic revision operations, it is important to devise the minimal set of operations that would be both intuitive and sufficient to support user's tasks [6]. For this purpose, we first identified eleven different possible topic revision operations (see Table 1) based on reviewing existing work on interactive topic model [1, 4, 16, 22]. Next, we prioritized the operations based on the following criteria ordered by their importance: 1) *Task relevancy:* To what extent this operation is relevant to the tasks involved in exploring conversation as identified in [15]? 2) *Topic model relevancy:* Is this operation applicable to our topic model approach? 3) *Redundancy:* Is this operation already covered by other operations, which are stronger on the previous two criteria?

The three operations at the bottom of Table 1 (9-11) are eliminated based on both task and topic model relevancy criteria. Not only these operations are designed to fix the term-topic distribution which is not applicable to our topic modeling approach; but more importantly they are arguably not very useful to support the high level exploratory reading tasks as identified in [15] and therefore the users may not be motivated to perform such operations. On the contrary, we selected the top three operations in Table 1 (i.e., 'split a topic', 'merge topics by join', and 'merge topics by absorption'), because we identified them as the most relevant to our exploratory reading tasks that require the user to dynamically change the granularity level of different topics. Also, by selecting them some

[1] http://slashdot.org

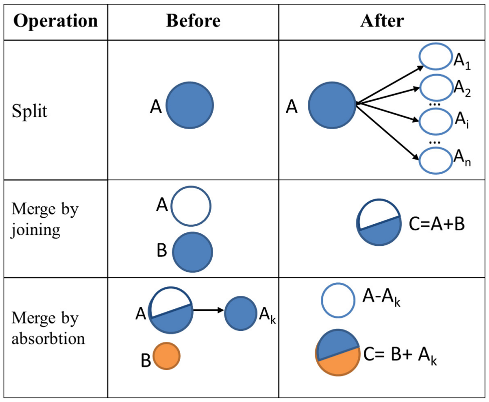


Figure 3. Three different user operations for topic revision

other candidate operations with lower task relevancy become redundant and therefore they are eliminated, such as 'change the overall granularity level of topics' (covered by topic splitting and merging) and 'split by keyword' (covered by topic splitting). In the reminder of the section, we describe how each of these operations support user's tasks, and how the underlying topic model is revised according to these operations.

Split a topic

Topic splitting allows the user to explore more specific subtopics of a given topic, thus changing the topic granularity to a finer level. Consider an example, where initially the system creates a topic named 'military security'. As the user starts exploring this topic, she finds it to be too generic with respect to her information needs and therefore she wants to split it into more specific sub-topics.

Method: Assume that the user wants to split a topic A into multiple sub-topics (see Figure 3). Upon user's request, the underlying topic model creates a sub-graph $G_A(V_A, E_A) \subset G(V, E)$ from the original graph $G(V, E)$ generated in the initial topic segmentation, where V_A represents the vertices (sentences) of topic A, and and each edge $w(x, y)$ in E_A represents the edge weights of topic A.

Next, the system splits the chosen topical cluster A into further n sub-clusters $A_1, A_2, ..., A_n$, by applying the same graph partitioning algorithm used in the initial topic segmentation phase, i.e., approximate solution to n-Cut [30] on $G_A(V_A, E_A)$. Here, n is the optimal number of sub-topics, which is automatically determined by finding the value of n for which an objective function Q is maximized according to the formula proposed by Newman and Girvan [24],

$$Q_n = \sum_{c=1}^{n} \frac{\sum_{x \in V_c, y \in V_c} w(x,y)}{\sum_{x \in V_A, y \in V_A} w(x,y)} - \Big(\frac{\sum_{x \in V_c, y \in V_A} w(x,y)}{\sum_{x \in V_A, y \in V_A} w(x,y)} \Big)^2 \tag{1}$$

Here, Q_n measures the quality of a clustering of nodes in the graph $G_A(V_A, E_A)$ into n groups, where $\sum_{x \in V_c, y \in V_c} w(x, y)$ measures the within-cluster sum of weights, $\sum_{x \in V_A, y \in V_A} w(x, y)$ measures the sum of all edge weights in the graph, and $\sum_{x \in V_c, y \in V_A} w(x, y)$ measures the sum of weights over all edges attached to nodes in cluster c. In essence, according to (1), the nodes in high quality clusters should have much stronger connections among themselves than with other nodes in the graph.

We apply equation (1) for the value of $n = 2, 3, 4, 5$ and select the value of n, for which Q_n is maximum. The highest possible value is capped to 5 because of time constraint imposed by the interactive nature of the operation. Notice however, that this limitation is not too penalizing. Our analysis of the Slashdot corpus shows that in 86% cases of splitting a topic, the best value of Q_n is with $n \leq 5$.

Once the parent topic is segmented into n different subclusters, representative keyphrases are generated for each sub-topic. This is done by running our topic labeling method only on the sub-conversation covered by A.

Merge by joining

This operation allows the user to aggregate multiple similar topics into a single one. As opposed to topic splitting, the result is a topic with coarser granularity level. Consider an example, where the initial topic model produces two different topics namely 'secure code' and 'simple sql server injection'. The user may find that both topics are too specific, therefore joining them into a more generic topic may help her to better perform the subsequent tasks.

Method: Assume that the user decides to merge by joining two topics A and B (see Figure 3). To perform this operation, the topic modeling system creates another topic C and assigns its vertices as $V_C = V_A \bigcup V_B$ and edges as $E_C = E_A \bigcup E_B$. After that, a label for C is generated. This is done by running our topic labeling method only on the sub conversation covered by C.

Merge by absorption

If a sub-topic is more related to a different topic than its current parent topic, merge by absorption allows the user to separate this sub-topic from its current parent and merge it with the one to which it is more related. Unlike the previous merge operation (which joins two independent topics), this operation allows a sub-topic that is already placed under a topic to be absorbed by a different parent topic. Consider an example, where the sentences related to two different topics, namely 'Obama health policy' and 'job recession' are wrongly grouped together under the topic 'Obama recession'. The user may realize that the sub-topic 'job recession' should be separated from its parent topic and merged with the 'unemployment' topic to which it is more related.

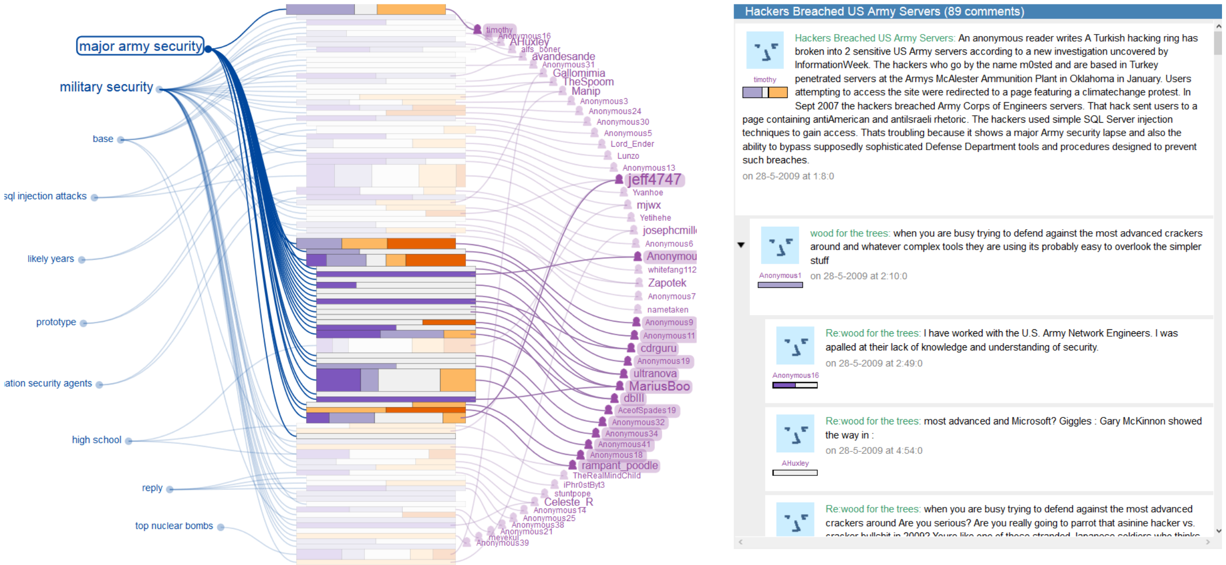
Method: Upon receiving a merge by absorption feedback from the user on A_k and B, the topic modeling system removes the sub-topic A_k from its current parent A and merge it with the topic B (see Figure 3). The system then creates a new parent topic C and then assigns vertices such that,

$$V_C = V_{A_k} \cup V_B, V_A = V_A \setminus V_{A_k} \tag{2}$$

and edges such that,

$$E_C = E_{A_k} \cup E_B, E_A = E_A \setminus E_{A_k} \tag{3}$$

After that, the topic labeling method takes the portion of the conversation that consists of the sentences in V_C, thus generating a label for C that potentially represents descriptive keyphrases from both topics A_k and B.



Hackers Breached US Army Servers: An anonymous reader writes A Turkish hacking ring has broken into 2 sensitive US Army servers according to a new investigation uncovered by InformationWeek. The hackers who go by the name m0sted and are based in Turkey penetrated servers at the Armys McAlester Ammunition Plant in Oklahoma in January. Users attempting to access the site were redirected to a page featuring a climatechange protest. In Sept 2007 the hackers breached Army Corps of Engineers servers. That hack sent users to a page containing antiAmerican and antiIsraeli rhetoric. The hackers used simple SQL Server injection techniques to gain access. Thats troubling because it shows a major Army security lapse and also the ability to bypass supposedly sophisticated Defense Department tools and procedures designed to prevent such breaches.

on 28-5-2009 at 1:8:0

wood for the trees: when you are busy trying to defend against the most advanced crackers around and whatever complex tools they are using its probably easy to overlook the simpler stuff

on 28-5-2009 at 2:10:0

Re:wood for the trees: I have worked with the U.S. Army Network Engineers. I was apalled at their lack of knowledge and understanding of security.

on 28-5-2009 at 2:49:0

Re:wood for the trees: most advanced and Microsoft? Giggles : Gary McKinnon showed the way in :

on 28-5-2009 at 4:54:0

Re:wood for the trees: when you are busy trying to defend against the most advanced crackers around Are you serious? Are you really going to parrot that asinine hacker vs. cracker bullshit in 2009? Youre like one of those stranded Japanese soldiers who thinks

Figure 4. A snapshot of ConVisIT showing a blog conversation from Slashdot: the Thread Overview visualizes the whole thread and how the sentiment is expressed for each comment (middle-left); the Facet Overview presents topics and authors circularly around the Thread Overview; and the Conversation View presents the actual conversation (right). Here, the user hovers the mouse over the topic element ('major army security'). This action highlights the connecting visual links, brushing the related *authors*, and providing visual prominence to the related comments in the Thread Overview.

CONVISIT: EXPLORING CONVERSATIONS USING INTERACTIVE TOPIC MODEL

In order to effectively support the user in exploring conversations, we enable the three interactive topic revisions by redesigning the ConVis interface [15], which was originally designed to present static topic model results as a linear list. In this section, we first provide an overview of the visual interface features common in both ConVis and ConVisIT, along with their justification in term of visual encoding and interactions. Next, we describe the interaction redesign made in ConVisIT to incorporate topic revision operations.

Features common to both Interfaces

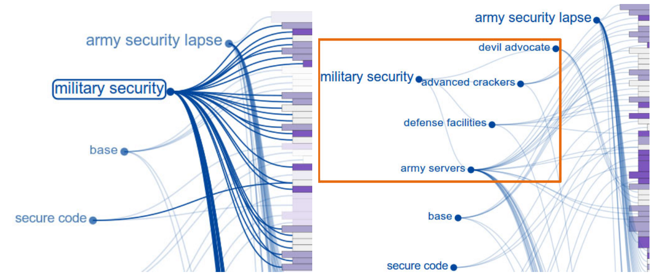
The interface starts with visualizing the initial topic model, the sentiments being expressed and a set of metadata of the given conversation (See Figure 4). It is primarily an overview + details interface, since this design has been found to be more effective for text comprehension tasks than other approaches such as zooming and focus+context [8]. The overview consists of the whole thread as well as the topics and authors of the conversation. The Thread Overview visually represents each comment of the discussion as a stacked bar, where each stacked bar encodes three different metadata (comment length, position of the comment in the thread, and depth of the comment within the thread). A set of five diverging colors was used to visualize the distribution of sentiment orientation of a comment in a perceptually meaningful order, ranging from purple (highly negative) to orange (highly positive). Thus, the distribution of colors in the Thread Overview can help the user to perceive the kind of conversation they are going to deal with. For example, if the Thread Overview is mostly in strong purple color, then the conversation has many negative comments.

The primary facets of the conversations, namely topics and authors are presented in a circular layout around the Thread Overview (Figure 4). Both topics and authors are positioned according to their chronological order in the conversation starting from the top, allowing the user to understand how the conversation evolves as the discussion progresses. The font size of facet items helps the user to quickly identify what are the mostly discussed themes and who are the most dominant participants within a conversation. To indicate topic-comment-author relationship, the facet elements are connected to their corresponding comments in the Thread Overview via subtle curved links. These visual links allow the user to perceive the related entities more quickly and with greater subjective satisfaction than plain highlighting [31]. Finally, the Conversation View displays the actual text of the comments in the discussion as a scrollable list.

The user can start exploring the conversation by hovering the mouse on topics, which highlights the connecting curved links and related comments in the Thread Overview. As such, one can quickly understand how topics may be related to different comments and authors. Then, if the reader becomes further interested in specific topic/author, she can click on it. As a result, a thick vertical outline is drawn next to the corresponding comments in the Thread Overview. Such outlines are also mirrored in the Conversation View. Besides exploring by the topics/authors, the reader can browse individual comments by hovering and clicking on them in the Thread Overview. In particular, when the user hovers over a comment its topic is highlighted, while when the user clicks on a comment, the actual text for that comment is shown in the Conversation View (by scrolling). In this way, the user can easily locate the comments that belong to a particular topic.

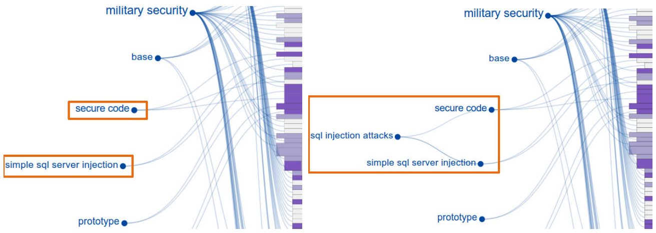
Interactive visualization for topic revisions

As the user explores the conversation, she may realize that the initial topic model is not helping her anymore, and may want to revise it. To support such situation, ConVisIT provides a set of interactive topic revision operations within the interface



(a) Before splitting (b) After splitting

Figure 5. An example showing: (a) The user hovers over the topic 'military security' and decides to perform the split operation. (b) As a result, the topic moves to its left while the rest of the topics are pushed along the perimeter of the circular layout to create space for the new children.



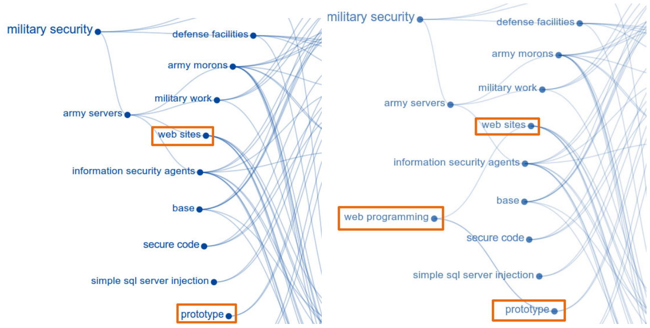
(a) Before merge by joining (b) After merge by joining

Figure 6. An example showing: (a) The user decides to merge two topics by joining (indicated by orange color). (b) As a result, ConVisIT updates the topic organization where these two topic nodes are merged under the parent topic 'subject sql injection attack'.

through some intuitive direct manipulation methods. As the user performs these operations, the system updates the topic model and changes the visual encoding of the topic list from the initial flat list of topics into a multi-rooted tree organization. Such updates to the topic organization becomes visible to the user through perceptually meaningful animations, following the design guidelines of effective animation presented in [14]. In particular, we have designed staged animation for each operation, i.e., we break up the corresponding transition into a set of simple sub-transitions, allowing multiple changes to be easily observed.

For instance, when the user splits a topic by double clicking on it, the following sub-transitions occur. First, the clicked topic A moves to the left along with its parent node(s) (if any), while existing nodes at the deepest level are pushed towards their new positions (up/down) around the circular layout to create angular spaces for the new sub-topics. Second, the new sub-topics $A_1, A_2, ...A_n$ appear and move from their parent's position (A) to their new positions. Third, labels appear for these sub-topics (see Figure 5(b)). Double clicking on A again causes it to collapse by following the exact reverse order of animation, i.e., the labels of the children move from their current positions to their parent and fade away, and then the parent moves to its previous position while other nodes move closer to the parent node to fill the gaps left by the removed children nodes.

Merging of two topics can be performed by dragging a topic A over another topic B, which causes the system to update the topic model. As a result, a new parent topic C appears to the left and curved links are drawn from C to A and B to indicate parent-child relationship (see Figure 6). The user can subsequently double click on C to collapse it, which hides its sub-topics. Finally, if a child topic A_k is discovered to be wrongly placed under a topic A instead of under a more appropriate topic B, the user can drag A_k over B. As a result, the link of A_k with its parent A is removed and then a new parent node C appears that connects both A_k and B (see Figure 7).



(a) Before merge by absorption (b) After merge by absorption

Figure 7. An example showing: (a) The user decides to perform merge by absorption on two topics 'web sites' and 'prototype'. (b) ConVisIT updates the topic organization where the previous link from 'web sites' to 'army server' is removed, and then 'web sites' is absorbed into a more generic parent topic 'web programming' along with 'prototype'.

As the user continues to perform interactive topic revisions, the topic organization can potentially grow quickly to multiple levels of hierarchy due to iterative splitting and merging. The current implementation can reasonably show a topic organization having a tree depth up to four levels, when the visualization is used on a 1920 x 1080 screen. This seems adequate for conversations with no more than a few hundreds comments, because the number of sub-topics grows exponentially with the depth of the topic hierarchy, and topics at the bottom of a hierarchy of depth four becomes so specific (i.e., cover so few sentences) that further splitting would be inappropriate. For instance, if we assume that the avg. branching factor in a single-rooted topic hierarchy is 3 and the conversation contains 300 sentences, each leaf of the topic hierarchy of depth 4 will contain on avg. $(300/3^4) = 3.7$ sentences.

IMPLEMENTATION

A server side component (in php) communicates with the topic modeling system (in python) to produce the updated results. The visualization component, on the other hand, is implemented in Javascript (using D3 and JQuery library), which is sufficiently fast to respond in real time to the user actions. The system runs on a laptop computer with a 2.4 GHZ processor and 16 GB RAM. The average processing time for topic splitting operation is 6.92 sec. and for topic merging operation is 2.74 sec. (over the initial set of topics in our corpora). In order to increase the response time, topic split results were cached by the system for all the topics in the initial topic model, as well as for the sub-topics as soon as they were created upon topic revision operations.

USER STUDY

The goal of the study is to understand how the introduction of visual interfaces for exploring conversations may influence the user performance and subjective opinions compared to more traditional interfaces. In this paper, we have introduced ConVisIT, which is highly interactive, providing the capability to revise topic models. Its precursor, ConVis [15] is also an interactive visualization for exploring conversations, however it does not support the topic revision operations. Finally, as a traditional interface for exploring conversation, we have re-implemented the interface to the popular Slashdot blog. The user study aims to answer the following two questions:

(1) When we compare ConVisIT, ConVis and the Slashdot interface, is there any difference in user performance and subjective reactions?

- Does one interface help to find more insightful comments in a conversation?
- Is one interface perceived as more useful and easy to use?
- Is reading behavior influenced by the interfaces? If the answer is 'Yes' then how?

(2) What specific visualization features/components of the three different interfaces are perceived as more/less beneficial by the potential users (e.g., interactive topic revision, Thread Overview, and relations between facets)?

Methodology

Since the above research questions require comparisons among different user interfaces, we perform a summative evaluation through controlled experiments [21]. The study was designed with three interfaces as conditions: a) *Slashdot*, b) *ConVis*, and c) *ConVisIT*. The Slashdot interface follows a typical blog reading interface design and it serves as a suitable baseline for our experiments. It provides an indented list based representation of the whole conversation as well as common functionalities of blog interfaces such as scrolling up and down, collapsing a sub-thread, and searching for terms. The primary reason for comparing between ConVis and ConVisIT was to verify whether any potential influence in performance and user behaviour are due to the visualization features common between them, or due to the interactive topic revision feature (which is only present in ConVisIT). For fair comparison, different interface parameters such as screen size and font size were kept the same across all the interfaces. Moreover, a within-subject design was used for this experiment with interface as the within-subject factor, allowing us to directly compare the performance and subjective measures of each participant with respect to all three interfaces. Finally, all study aspects, including instructions and setup, went through several iterations of evaluation and pilot testing with two users, who did not participate in the actual study.

Participants

20 subjects (aged 19-43, 8 females) with considerable prior experience of reading blogs participated in the study. They held a variety of occupations including engineer, software developer, and students at undergraduate, masters, and PhD levels, mostly with strong Science background. They were recruited through emails and Reddit posts.

Task and procedure

At the beginning, a pre-study questionnaire was administered to capture demographic information and prior experience with blog reading. Then, the user went through the following steps for each of the three interfaces: 1) In a warm-up session, the interface was introduced to the participant using a sample conversation; 2) The participant was then asked to perform a task on a given conversation (a different conversation was provided for each interface). Rather than asking some specific questions, we provided an open-ended task to reflect the exploratory nature of blog reading. We asked the participant to explore the conversation according to her own interests using the given interface and write down a summary of the keypoints found while exploring the conversation. The study lasted approximately 90 minutes and each participant was paid $20 to participate.

We carefully selected three different conversations from the Slashdot blog corpora having similar number of comments (89, 101, and 89) to avoid potential variations due to the conversation length or complexity of the thread. Also, to counterbalance any potential learning effects due to the order of exposure to specific interfaces and conversations, the order was varied using a 3 x 3 Latin square.

During the study, we collected both quantitative data such as task completion time and qualitative data such as observations and questionnaires. After completing the task with each interface, participants rated the following measures on a 5 point Likert scale: 1) *usefulness*: 'I found this interface to be useful for browsing conversations'; 2) *easeofUse*: 'I found this interface to be easy to use'; 3) *enjoyable*: 'I found this interface enjoyable to use'; and 4) *findInsightfulComments*: 'This interface enabled me to find more insightful comments'. At the end of the study, post-study questionnaires followed by and semi-structured interviews were administered regarding the individual features and overall interfaces. The recorded interviews were transcribed and coded to facilitate analysis. Finally, we logged interface actions to better compare the usage patterns of the three different interfaces.

Results analysis

Quantitative analysis

The results of the in-study questionnaires are presented in Figure 8, showing the average rating expressed by the participants on four different measures. Since the data was collected using a standard 5 point Likert scale, standard parametric analysis is not suitable due to the lack of normality [19]. Instead we perform nonparametric analysis i.e., Mann-Whitney's U tests on the responses for each of these measures. Finally, all reported pairwise comparisons are corrected with the Bonferroni adjustment.

The analysis reveals that the interfaces significantly affected *findInsightfulComments*, with pairwise comparisons showing that ConVisIT was perceived to help them in finding more insightful comments than the ConVis and the Slashdot interfaces. This is an important result because it supports our intuition that by allowing the user to dynamically modify the topic organization (in ConVisIT), we enable her to find more insightful comments. There were also significant effects of interface on *usefulness* (see Table 2), with pair-wise tests showing that ConVisIT and ConVis were perceived to be significantly more useful than the Slashdot interface (see Table 2). Also, ConVisIT was rated slightly more useful than ConVis, although the difference was not significant. Similar

Measures	Slashdot vs ConVis	Slashdot vs ConVisIT	ConVis vs ConVisIT
usefulness	$Z = -3.423; p < 0.001$	$Z = -3.725; p < 0.001$	$Z = -0.560; p = 0.575$
easeofUse	$Z = -0.646; p = 0.518$	$Z = -0.657; p = 0.511$	$Z = -0.144; p = 0.885$
findInsightfulComments	$Z = -3.855; p < 0.001$	$Z = -4.792; p < 0.001$	$Z = -2.003; p < 0.05$
enjoyable	$Z = -3.770; p < 0.001$	$Z = -3.888; p < 0.05$	$Z = -.652; p = 0.24$

Table 2. Statistical analysis (Mann-Whitney's U test) on *usefulness*, *easeofUse*, *enjoyable* and *findInsightfulComments* measures.

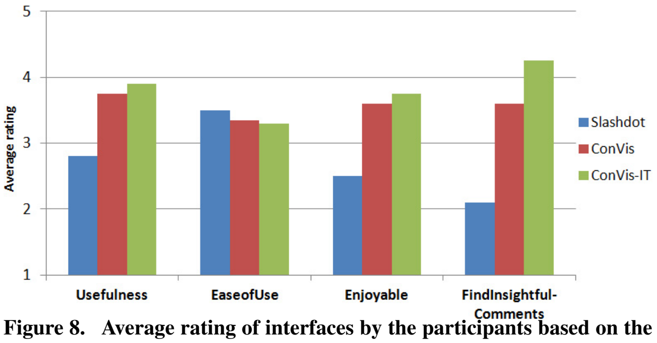


Figure 8. Average rating of interfaces by the participants based on the following measures: *usefulness*, *easeofUse*, *enjoyable* and *findInsightful-Comments* . Longer bars indicate better rating.

results were obtained on *enjoyable* measure, where ConVis and ConVisIT were rated significantly higher than Slashdot (see Figure 8). Finally, the *easeofUse* measure is not significantly affected by the interfaces, indicating that none of the interfaces was superior on this measure. However, this is a favorable outcome for ConVisIT in that even though its interactive features are more complex than in ConVis, the participants did not report ConVisIT as being significantly more difficult to use. Similarly, it is also a favorable outcome for both ConVisIT and ConVis, since, in spite of their complexity, they were found to be as easy to use as the simpler traditional blog interface.

Interface Features

The in-study questionnaire also included a number of questions regarding the usefulness of specific features of the three interfaces. To complement this data, we also analyzed the interaction log data of ConVis, ConVisIT, and Slashdot. The quantitative results of the subjective responses are provided in Figure 9. We can readily see that the majority of the responses regarding features of the Slashdot interface range from strongly disagree to neutral. In contrast, responses regarding ConVis and ConVisIT features are dominated by strongly positive to neutral ratings.

Regarding topic revision operations, *Split* was found to be more useful (35 % strongly agree and 40 % agree) than *Merge* (20 % strongly agree and 25 % agree). This is also evident from the usage of these operations, as the split operation was used more frequently (5.3 times on average) than merge (1.6 times on average). Moreover, 16 out of 20 users performed split operation prior to performing any merge operation. A possible explanation is that participants generally found the initial topic model results to be coarse grained with respect to their information needs, expertise and metal model, and therefore they tended to apply split operation both earlier on and more frequently than the merge operation so that they could read at finer topic granularity.

An interesting observation from the log data is that even though some features were common in both ConVis and ConVisIT, they were used more frequently with ConVisIT. For example, participants hovered and clicked on topics significantly more times on average using ConVisIT (60.4 and 82.0 respectively) than using ConVis (11.6 and 19.1 respectively). A possible explanation is that due to the presence of interactive topic revision features, the participants could create topics that were more useful to them and therefore they relied on topics more frequently in their exploration.

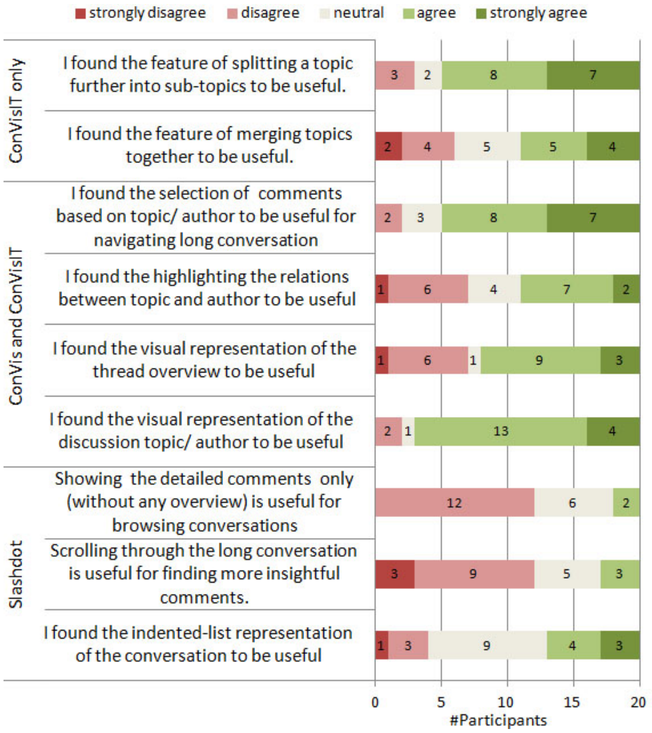


Figure 9. Responses to statements regarding specific features of the three interfaces under investigation.

Time

Interestingly, the average time required to complete the tasks was not significantly affected by the interfaces, with Slashdot, ConVis, and ConVisIT requiring 1056, 1240 and 1159 secs respectively. This result is rather promising, because it indicates that participants were not slowed down by the fact that they were unfamiliar with the topic revision operations and by the overhead involved in performing those operations. On the contrary, they perceived to find more insightful comments as reported before.

Qualitative Analysis

Overall Preference

At the end of the study, participants were asked to indicate their overall preference for a blog reading interface and then justify their choice. 60 % of the participants indicated a preference for ConVisIT, 25 % for ConVis, and 15 % for Slashdot. Most of the participants who chose ConVisIT felt that the topic revision operations were very helpful in finding relevant comments: *"ConVisIT is the most convenient interface because of its splitting and merging features. Using this interface to understand the conversation, I really did not have to go through all the comments" (P19).* It was also evident that when the granularity level of the topics did not match the user's information needs, ConVisIT was especially helpful for navigation: *"Sometimes the first-level keywords are way too generic, so it's better to navigate via second-level categories (P11)".* However, participants did become frustrated in a few cases, when ConVisIT could not accurately split the topics into meaningful list of sub-topics as mentioned by P13: *"...I enjoyed the ability to split apart topics, though I think it would benefit from better categorization of topics as I felt like some were misclassified".*

Those participants who chose the ConVis interface over its counterparts emphasized the utility of its visual components, i.e., the visual representation of the thread and highlighting the relations between topics and comments, which *"...makes it easier to find out which comments are more interesting"* , and *"...allowed me to see more of what was going on, how comments were inter-related, as well as kept me interested and focused on the thread as a whole." (P4).* The primary reason for preferring ConVis over ConVisIT was that sometimes the revised topic organization became too cluttered or made the navigation too complex: *"... drilling down to a subtopic made the graph look too cluttered up. Sometimes, it was harder to figure out if two topics were at the same level or not based on the layout." (P7)*, and *"It felt like a good mix, others were too complex (ConVisIT) or too simple (Slashdot)." (P2).*

Three participants who preferred the Slashdot interface felt that it was easier to use, although *"...it is not giving me the structural information that I am interested about " (P16).* Another reason was that they were so much familiar with this interface: *"Scrolling through the conversation was good enough for me to find important topics in it, maybe because I am used to reading things this way." (P15).*

DISCUSSION

Summary of findings

Based on our analysis of the study results, we now revisit our research questions. Our first research question was whether there is any difference in user performance and subjective reactions due to the interface condition. We found that overall ConVisIT was the most preferred interface, and was rated higher over its counterparts on the *findInsightfulComments* measure. On the contrary, Slashdot was the least preferred interface, and it received significantly lower rating on three different measures. As for ConVis, it seems to provide a middle ground between the other two interfaces and its topic organization, although static, was found to be visually less cluttered than the one of ConVisIT. In general, this shows that while interactive topic model can be beneficial to the user, such feature may introduce visual clutter and interaction costs at least for some users. Finally, there were no significant differences among the interfaces in terms of *easeOfuse*, in spite of the higher complexity of ConVis and ConVisIT.

The second key research question was what specific visualization features/ components of the interfaces are perceived as more/less beneficial by the potential users (e.g., interactive topic revision, Thread Overview, and relations between facets)? We found that in general, the visualization features of ConVis and ConVisIT received higher rating than the ones of Slashdot. Interestingly, we found that subjective reactions about different features of the interfaces such as split, merge, and clicking on topic directly correlates with their frequency of use. More importantly, we found that not all interactive topic revision operations were equally received. For example, the split operation was used more frequently than its counterparts. This issue needs to be further investigated.

Study limitations

Even though a controlled study is suitable for comparing different interfaces, it may not accurately capture real world uses scenario [21]. Although we carefully recruited participants who were frequent blog readers, still different settings were controlled to make a fair comparison among interfaces (e.g., they were not allowed to choose a conversation according to their own interest). In this context, a more ecologically valid evaluation [3] would be to allow the users to explore their own conversations of interest over an extended period of time. Such longitudinal study would provide valuable insights regarding the utility of the interface. For instance, one possible way of enhancing the ecological validity could be to make the system publicly available and invite potential participants to carry out their usual reading activity with ConVisIT interface.

CONCLUSIONS AND FUTURE WORK

This paper presents a novel human-in-the-loop topic modeling approach combined with a visual interface to support the exploration of large conversations. With ConVisIT, users can explore and revise the topic model to better fulfill their information needs. Our evaluation suggests that ConVisIT can enhance the user's ability to find more insightful comments, even if they are buried in a long conversation. Remarkably, ConVisIT was preferred by the majority of the participants over the other two interfaces (i.e., ConVis and Slashdot) that do not support interactive topic revision, indicating that users benefit from getting more control over the topic modeling process while exploring conversations.

There are several important avenues for extending ConVisIT. While our approach has been found to be useful for Slashdot conversations about technology, we plan to investigate its potential utility in other domains, ranging from political to health related blogs, to education forums for Massive Open Online Courses (MOOC). In this regard, we plan to conduct longitudinal case studies with real users for each of these domains. Finally, exploring a large set of conversations is arguably even more challenging task than when with only one conversation, because the volume and complexity of the textual data may drastically increase and the information overload problem could be even more prevalent among users [17]. Therefore, we aim to extend our interactive topic modeling approach to handle a large collection of asynchronous conversations, where the user will be able to explore topics that are discussed over many different threads.

REFERENCES

1. Andrzejewski, D., Zhu, X., and Craven, M. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proc. International Conf. on Machine Learning* (2009), 25–32.
2. Carenini, G., Murray, G., and Ng, R. *Methods for Mining and Summarizing Text Conversations*. Morgan Claypool, 2011.
3. Carpendale, S. Evaluating information visualizations. In *Information Visualization*. Springer, 2008, 19–45.
4. Choo, J., Lee, C., Reddy, C. K., and Park, H. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Trans. Visualization & Comp. Graphics 19*, 12 (2013), 1992–2001.
5. Chuang, J., Gupta, S., Manning, C., and Heer, J. Topic model diagnostics: Assessing domain relevance via topical alignment. In *Proc. Conf. on Machine Learning* (2013), 612–620.
6. Chuang, J., Hu, Y., Jin, A., Wilkerson, J. D., McFarland, D. A., Manning, C. D., and Heer, J. Document exploration with topic modeling: Designing interactive visualizations to support effective analysis workflows. In *NIPS Workshop on Topic Models: Computation, Application and Evaluation* (2013).
7. Chuang, J., Manning, C. D., and Heer, J. Termite: Visualization techniques for assessing textual topic models. In *Proc. International Working Conf. on Advanced Visual Interfaces* (2012), 74–77.
8. Cockburn, A., Karlson, A., and Bederson, B. B. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR) 41*, 1 (2008), 2.
9. Dave, K., Wattenberg, M., and Muller, M. Flash forums and forumreader: navigating a new kind of large-scale online discussion. In *Proc. ACM Conf. on CSCW* (2004), 232–241.
10. Dou, W., Yu, L., Wang, X., Ma, Z., and Ribarsky, W. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE Trans. Visualization & Comp. Graphics 19*, 12 (2013), 2002–2011.
11. Endert, A., Fiaux, P., and North, C. Semantic interaction for visual text analytics. In *Proc. Conf. CHI* (2012), 473–482.
12. Galley, M., McKeown, K., Fosler-Lussier, E., and Jing, H. Discourse segmentation of multi-party conversation. In *Proc. Annual Meeting on Association for Computational Linguistics* (2003), 562–569.
13. Havre, S., Hetzler, E., Whitney, P., and Nowell, L. Themeriver: visualizing thematic changes in large document collections. *IEEE Trans. Visualization & Comp. Graphics 8*, 1 (2002), 9–20.
14. Heer, J., and Robertson, G. G. Animated transitions in statistical data graphics. *IEEE Trans. Visualization & Comp. Graphics 13*, 6 (2007), 1240–1247.
15. Hoque, E., and Carenini, G. Convis: A visual text analytic system for exploring blog conversations. *Computer Graphics Forum 33*, 3 (2014), 221–230.
16. Hu, Y., Boyd-Graber, J., Satinoff, B., and Smith, A. Interactive topic modeling. *Machine Learning 95*, 3 (2014), 423–469.
17. Jones, Q., Ravid, G., and Rafaeli, S. Information overload and the message dynamics of online interaction spaces: A theoretical model and empirical exploration. *Information Systems Research 15*, 2 (2004), 194–210.
18. Joty, S., Carenini, G., and Ng, R. T. Topic segmentation and labeling in asynchronous conversations. *Journal of Artificial Intelligence Research 47* (2013), 521–573.
19. Kaptein, M. C., Nass, C., and Markopoulos, P. Powerful and consistent analysis of likert-type ratingscales. In *Proc. Conf. CHI* (2010), 2391–2394.
20. Kerr, B. Thread arcs: An email thread visualization. In *IEEE Symposium on Information Visualization* (2003), 211–218.
21. Lam, H., Bertini, E., Isenberg, P., Plaisant, C., and Carpendale, S. Empirical studies in information visualization: Seven scenarios. *IEEE Trans. Visualization & Comp. Graphics 18*, 9 (2012), 1520–1536.
22. Lee, H., Kihm, J., Choo, J., Stasko, J., and Park, H. ivisclustering: An interactive visual document clustering via topic modeling. In *Computer Graphics Forum*, vol. 31, Wiley Online Library (2012), 1155–1164.
23. Munzner, T. A nested model for visualization design and validation. *IEEE Trans. Visualization & Comp. Graphics 15*, 6 (2009), 921–928.
24. Newman, M. E., and Girvan, M. Finding and evaluating community structure in networks. *Physical review E 69*, 2 (2004), 026113.
25. Pan, S., Zhou, M. X., Song, Y., Qian, W., Wang, F., and Liu, S. Optimizing temporal topic segmentation for intelligent text visualization. In *Proc. International conf. on Intelligent User Interfaces)* (2013), 339–350.
26. Pascual-Cid, V., and Kaltenbrunner, A. Exploring asynchronous online discussions through hierarchical visualisation. In *IEEE Conf. on Information Visualization* (2009), 191–196.
27. Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proc. Conf. on EMNLP* (2009), 248–256.
28. Sack, W. Conversation map: an interface for very-large-scale conversations. *Journal of Management Information Systems 17*, 3 (2000), 73–92.

29. Sedlmair, M., Meyer, M., and Munzner, T. Design study methodology: reflections from the trenches and the stacks. *IEEE Trans. Visualization & Comp. Graphics 18*, 12 (2012), 2431–2440.

30. Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence 22*, 8 (2000), 888–905.

31. Steinberger, M., Waldner, M., Streit, M., Lex, A., and Schmalstieg, D. Context-preserving visual links. *IEEE Trans. Visualization & Comp. Graphics 17*, 12 (2011), 2249–2258.

32. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. Lexicon-based methods for sentiment analysis. *Computational linguistics 37*, 2 (2011), 267–307.

33. Venolia, G. D., and Neustaedter, C. Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In *Proc. Conf. CHI* (2003), 361–368.

34. Viégas, F. B., Golder, S., and Donath, J. Visualizing email content: portraying relationships from conversational histories. In *Proc. Conf. CHI* (2006), 979–988.

35. Wattenberg, M., and Millen, D. Conversation thumbnails for large-scale discussions. In *extended abstracts on CHI* (2003), 742–743.

36. Wei, F., Liu, S., Song, Y., Pan, S., Zhou, M. X., Qian, W., Shi, L., Tan, L., and Zhang, Q. Tiara: a visual exploratory text analytic system. In *Proc. ACM Conf. on Knowledge Discovery and Data Mining* (2010), 153–162.

37. Zhou, D., Orshanskiy, S. A., Zha, H., and Giles, C. L. Co-ranking authors and documents in a heterogeneous network. In *Seventh IEEE International Conf. on Data Mining* (2007), 739–744.