MultiConVis: A Visual Text Analytics System for Exploring a Collection of Online Conversations

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

ABSTRACT

Online conversations, such as blogs, provide rich amount of information and opinions about popular queries. Given a query, traditional blog sites return a set of conversations often consisting of thousands of comments with complex thread structure. Since the interfaces of these blog sites do not provide any overview of the data, it becomes very difficult for the user to explore and analyze such a large amount of conversational data. In this paper, we present MultiConVis, a visual text analytics system designed to support the exploration of a collection of online conversations. Our system tightly integrates NLP techniques for topic modeling and sentiment analysis with information visualizations, by considering the unique characteristics of online conversations. The resulting interface supports the user exploration, starting from a possibly large set of conversations, then narrowing down to the subset of conversations, and eventually drilling-down to the set of comments of one conversation. Our evaluations through case studies with domain experts and a formal user study with regular blog readers illustrate the potential benefits of our approach, when compared to a traditional blog reading interface.

Author Keywords

Asynchronous conversations; text visualization; visual text analytics, 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, there has been an exponential growth of asynchronous online conversations discussing a large variety of popular issues like 'ObamaCare', 'US immigration reform', and 'Apple iWatch release'. Asynchronous conversations, such as blogs, may start with a news article or editorial opinion, and later generate long thread with hundreds of comments, which readers may become interested in exploring and analyzing to seek variety of information and opinions. However, given a query, traditional blog sites only present the set of relevant blogs as a paginated list ordered by their recency, without providing any high-level summary of the conversations. This navigational support is often inadequate to explore a set of blogs that may be of great interest to readers [22].

For example, consider the issue of 'iPhone bending' that went viral on social media, when the iPhone 6 was launched in September 2014. Soon after the product was released, some people claimed that this new phone can easily bend in the pocket while sitting on it. This incident triggered a huge amount of discussions in Macrumors [3], a blog site that regularly publishes Apple related news and allows participants to make comments. Within a few days, more than a dozen conversations with thousands of comments were generated in Macrumors covering various related issues, such as 'what users reported about the bending issue', 'what Apple says to defend its new product', and 'what are the reactions from the rivals of Apple' etc. In this situation, we could imagine three different users who would like to explore this set of conversations. First, a potential customer, who intended to buy an iPhone may want to explore these conversations to verify whether the bending issue is really serious. Second, a journalist may want to publish a story about what people are saying about the 'bending issue'. Finally, an Apple marketing analyst may want to get a pulse from the online community to make informed decision about how to react to the rumors and possibly redesign the products. In all cases, given the large number of conversations/comments, it would be extremely difficult and time-consuming for a user to explore and analyze all this information with the current blog interfaces, that only provide sequential access to conversations/comments.

Integrating Natural Language Processing (NLP) and Information Visualization (InfoVis) techniques has been proposed as a promising solution to this and similar textual/information overload problems [14, 40, 41]. In this work, we tightly couple NLP techniques for topic modeling and sentiment analysis with interactive visualizations to support the exploration and analysis of large set of conversations by considering the specific characteristics of the conversational domain. While asynchronous conversations comprise emails, blogs, microblogs (e.g., Twitter), and messaging; in this paper we focus on the blog conversations. In fact, blog conversations exhibit several unique characteristics: unlike microblog or messaging [35], they do not have fixed length comments; furthermore they have finer conversational structure as participants often reply to a post and/or quote a fragment of other com-

IUI'16,, March 07-10, 2016, Sonoma, CA, USA

DOI: http://dx.doi.org/10.1145/2856767.2856782

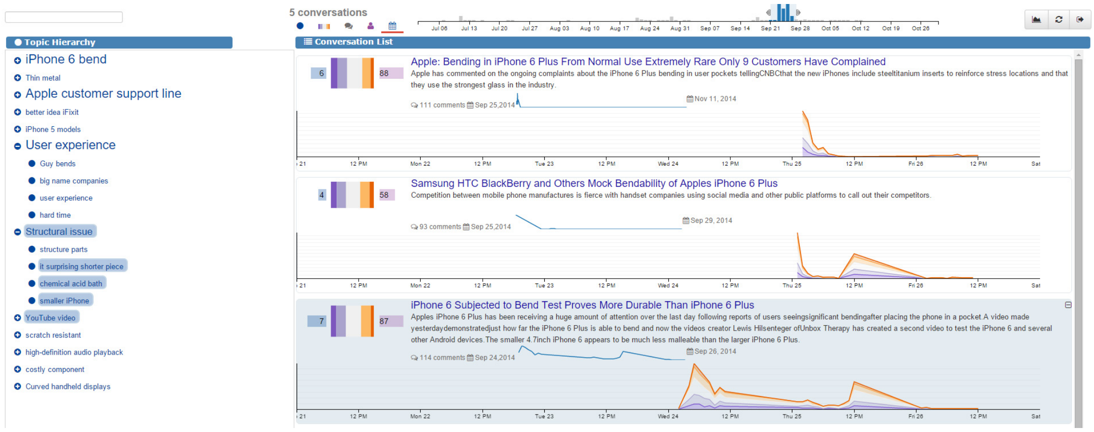


Figure 1. The MultiConVis interface, showing a subset of blog conversations returned by the query 'iPhone bending' from Macrumors in November 2014. Here, the user filtered out some conversations from the list using the Timeline located at the top, and then hovered on a conversation item (highlighted row in the right). As a consequence, the related topics from the Topic Hierarchy were highlighted (left).

ments [24]. In this paper, we consider these unique characteristics in devising both NLP and InfoVis techniques. Our work is closely related to a visual text analytic system named ConVis [20], which was also designed for blog conversations, but focused on supporting the user in exploring a single conversation. As we move from a single conversation to a collection of conversations, critical challenges emerge from the fact that users need to deal with a much larger amount of data, with different levels of granularity. For instance, the number of topics increases drastically for a set of conversations, therefore understanding and exploring these topics can be much more time consuming and cumbersome. Since some of these topics are similar in their semantic meaning, grouping them into a hierarchical topic organization may support the understanding and navigation of topics more effectively.

To address this challenge, we devise a hierarchical topic modeling technique that organizes the topics within a set of conversations into multiple levels, based on their semantic similarity. The resulting topic hierarchy is intended to better support user's understanding and navigation of the topics. We then design a visual interface that presents the hierarchial topic structure along with other conversational data as shown in Figure 1. The main contributions of this work are:

1) A **hierarchical topic modeling method** over a collection of conversations. While [20] describes how to effectively extract topics from a single conversation, here we propose a method which creates a topic hierarchy for a whole collection of conversations, by aggregating the topics extracted from each conversation in the collection.

2) The design and implementation of the **MultiConVis** interface, which supports exploration of a collection of blog conversations based on the topic hierarchy and sentiment. In essence, MultiConVis can be seen as an interface built on top of ConVis to allow the user to seemingly switch from exploring a collection of conversations to a single conversation. In particular, MultiConVis initially visualizes all the conversations in the whole collection, next supports the user in filtering out conversations that are irrelevant to her information needs, and then allows her to drill down to a specific conversation, which is visualized with the ConVis interface.

3) The evaluation of MultiConVis through a set of **case studies**, and a **user study** to investigate how the system influences user performance and subjective opinions when compared to a sample, traditional blog reading interface similar to existing interfaces, like Slashdot [4] and Macrumors [3].

RELATED WORK

Research prototypes that aim to support the exploration of a large collection of conversations can be categorized based on the information they extract and visualize: (a) metadata of the conversations (e.g., timestamps, tags, and authors), (b) the results of text analysis such as topic model and opinion.

Metadata visualization

Some earlier works have focused on how to support the exploration of a blog archive using only metadata, for example, by visualizing tags and comments arranged along a timeaxis [22], or by providing faceted visualization widgets for visual query formulation according to time, place, and tags [13]. While these works may assist users to find the blogs they are looking for, they are not designed to support users in understanding the actual content (i.e., the text) of these conversations. However, many tasks require the user to get overviews of the actual content of a collection of conversations, such as "Find out what are people feeling about *X* over time." Therefore, our goal is to visualize a combination of various metadata and textual analysis results that are identified as important in our user requirements analysis.

Topic modeling and visualization

In contrast to simply showing the metadata of the conversations, recently there have been some attempts to visualize the topics discussed within a collection of conversations [40, 41, 14]. Recent 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. Many of these works also consider the temporal aspects of topics by showing the evolution of topics over time. For example, *Themail* visualizes how topics in a collection of email conversations develop over time by arranging keywords selected based on term-frequency inverse document-frequency (TF-IDF) along a horizontal time axis [40]. *TIARA* [41] represents the temporal evolution of topics from an email collection by applying the ThemeRiver visualization [18], where each layer in the stacked graph represents a topic and the keywords of each topic are distributed over time. From the height of each topic and its content distributed over time, the user can see the topic evolution.

More recent works have tried to move beyond visualizing topics as a flat list, by organizing them into a hierarchy [15, 8, 29]. For example, *HierarchicalTopics* organizes a large number of topics into a tree structure by considering the distance between the probability distributions of topics [15]; and then utilizes a hierarchical ThemeRiver view to explore temporal trends of topics. Using the same algorithm, *TopicPanorama* builds topic hierarchies from multiple corpora (i.e., news, blogs, and microblogs), followed by matching these hierarchies using a graph-matching technique, so that the common and distinctive topics from different corpora can be visualized [29]. It combines a radially stacked tree visualization with a density-based graph visualization to facilitate the examination of the matched topic graph from multiple perspectives. Compared to these approaches that generate static topic hierarchies, *RoseRiver* focused on exploring the evolutionary patterns of hierarchical topics generated at different timeframes by conveying topic merging and splitting relationships over time using Sankey diagrams [8].

Organizing topics into a hierarchy can be very useful to our work as well, because the number of distinct topics in a collection of conversations may be quite high, compared to a single conversation. However, existing hierarchical topic modeling approaches are not designed specifically for conversational data. In contrast, MultiConVis creates a topic hierarchy for a collection of conversations by aggregating the topics of each conversation in the collection. And such topics are generated by taking specific characteristics of asynchronous conversations (e.g., reply-relationship) into account [20].

Opinion visualization

There is a growing interest in visualizing the opinions expressed in conversations, mostly focusing on microblogs [11, 31, 43]. Diakopoulos et al. presented *Vox Civitas* [11] that displayed sentiment and tweets volume over time for events discussed in microblogs to support the tasks of journalistic inquiry. *TwitInfo* [31] was also designed for visualizing microblogs with a focus on providing more accurate aggregation of sentiment information over a collection of tweets. Unlike these works, *OpinionFlow* focused more on visualizing the spreading of opinions about a particular topic (e.g., 'US government shutdown') among participants with a combination of a density map and a Sankey diagram [43]. Often the opinion information is summarized with other important aspects of information spreading such as temporal information, and the connections among conversation threads and authors [44].

A critical issue when abstracting data for sentiment analysis is how to aggregate sentiment information across sentences, comments and conversations. While all the works described above dealt with twitter data, in which tweets are only organized as a list, here we focus on a set of much more structured blog conversations, where each conversation consists of a set of comments organized in multiple threads with reply-relationships. We exploit this additional structure when we visually represent sentiment over multiple, different levels.

Facets \ Levels	**Collection of Conversations**	**One Conversation**
Topics	Hierarchy with all topics from all conversations	Conversation-level topics with explicit links to the topic hierarchy and to comments
Time	- Start day/time - Volume of comments over time	Ordinal time representations
Sentiment	- Sentiment distribution for each conversation - Sentiment evolution over time for each conversation	Sentiment distribution for each comment
Authors	Number of authors for each conversation	Conversation-level authors with explicit links to comments

Table 1. A summary of how facet elements are abstracted for a collection of conversations vs. one conversation.

USER REQUIREMENTS ANALYSIS

Why and how people read a collection of blogs has been studied extensively in the fields of computer mediated communication [46, 12, 26], social media [19, 10], human computer interaction [6, 33], and information retrieval [27, 30, 32, 37]. In essence, the primary goals of reading blogs include information seeking, fact checking, and opinion seeking [26, 9], which require the reader to understand what *topics* are discussed in the conversations and what *opinions* are expressed on those topics. Furthermore, users often exhibit a *variety seeking behaviour*, i.e., they tend to switch frequently from a topic to its sub-topics or to a completely different topic [37].

Blog readers also care about temporal aspects of the conversations [19, 10], for instance, the start and end time of a conversation, the chronological position of a comment with respect to the other comments within a conversation [6], and the volume of comments over time when exploring multiple conversations. Information about authors of the comments is also considered to be valuable [19], especially for blogs in which the same users participate frequently.

Table 1 summarizes our design choices for what information our interface should display, in light of current literature on blog readers. The row in the table corresponds to data facets and the columns to whether the facet is for multiple conversations vs a single one.

Since the number of topics for a collection of conversations is potentially much larger than for a single conversation, all the *topics* within a collection are organized into a hierarchy, while the topics of each single conversation are organized as a flat list and are explicitly connected to the comments of that conversation. To support the goals related to the *time* facet, the volume of comments over time is computed for each conversation in the collection of conversations, whereas within each conversation the chronological position of the comments is used. For the *sentiment* facet the distribution of sentiments across five polarity intervals (-2 to +2) is computed by counting how many sentences fall in each of these intervals. Here, for a collection of conversations, we compute the sentiment distribution for *each conversation*, whereas for one conversation, we compute this distribution at a finer level, i.e., for

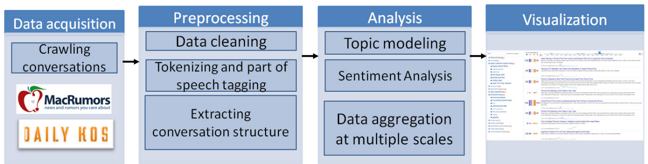


Figure 2. Overview of the MultiConVis system.

each comment. Finally, for the *authors* facet, while for a set of conversations only counts of authors are computed without providing the detailed list of authors, for one conversation the list of authors for that conversation is shown.

Current literature on blog reading not only inspired our data choices, but also guided the development of MultiConVis interactive visualization techniques. Considering the exploratory nature of blog reading, MultiConVis supports the user in browsing the set of conversations and comments by means of all the key facets (e.g., topics, sentiment and authors). Furthermore, the interface facilitate the exploration through the facets at different levels of granularity: from all conversations, to a subset of conversations, to one conversation. For consistency, elements of the same facet across different levels of granularity have similar visual mapping in terms of color, shape and other visual encoding channels. Finally, to facilitate the exploration and filtering of conversations, important attributes of each conversation (e.g., number of topics/ authors and overall sentiment distribution) are encoded as information scent [42].

SYSTEM OVERVIEW

The MultiConVis system consists of four major components as shown in Figure 2. Given a specified query (e.g., 'iPhone bending'), the *data acquisition module* invokes a blogsite such as Macrumors to crawl the set of conversations obtained from the first page of the search results returned by that site. Next, the *preprocessing module* performs data cleaning to retain only the conversational data in the crawled pages, followed by extracting the conversational structure, i.e., reply-relationships and quotation. We also use a state-of-the-art tagger [2] to tokenize text and annotate the tokens with their part-of-speech tags. After that, the *analysis module* performs topic modeling and sentiment analysis over the whole set of conversations. It then aggregates both metadata and results of text analysis at different granularity levels as described in the user requirements analysis. Finally, the *visualization module* displays the results obtained from the analysis module, and supports the user to interactively explore the conversations.

TEXT ANALYSIS

Topic hierarchy generation

Our topic modeling approach takes a collection of n blog conversations $C = \{c_1, c_2, ..., c_n\}$ that satisfies a user query, and generates a topic hierarchy following a bottom-up approach. In the resulting hierarchy, each node represents the cluster of sentences in the conversations that discuss the topic described by the label of the node. While one could think of a top-down approach to be more suitable for generating the topic hierarchy as it considers the whole set of conversations while generating the initial set of clusters (the roots of the hierarchy); we choose a bottom-up approach because in this

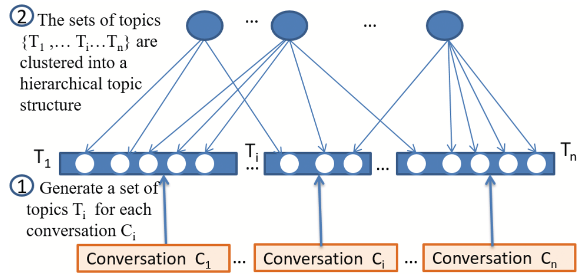


Figure 3. Hierarchical topic model generation.

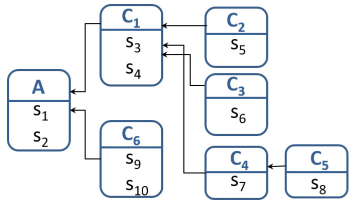
way we are able to take into account the conversational structure extracted from each conversation. In other words, we first generate a set of topic clusters for each conversation by taking advantage of its conversational structure, and then we organize these topic clusters from all the conversations into a hierarchy. More specifically, our topic hierarchy generation involves two primary steps as shown in Figure 3: 1) generate a set of topics T_i for each conversation $c_i \in C$; 2) aggregate all the T_i into a hierarchical topic structure for the whole collection.

For **topic Modeling over each conversation**, we adopt the method described in [20]. We briefly summarize it here, because our topic modeling method for a collection of conversations exploits similar data structures and techniques. Topic modeling of a single conversation starts by grouping the sentences of the conversation into a number of topic clusters (*segmentation*). Then, representative key phrases are assigned to each of these clusters (*labeling*).

In essence, topic segmentation applies a Lexical Cohesion-based Segmenter (LCSeg) [16] to each thread in the conversation as shown in Figure 4, where each thread represents a path from the initial message to a leaf message. Notice that after running the LCSeg algorithm, two sentences (e.g., s_1 and s_4) may appear together in the same segment in one thread $(A, C1, C2)$, while falling into different segments in another thread $(A, C1, C5)$. To consolidate all the (possibly conflicting) segmentation decisions made on each thread, we apply an efficient min-cut graph partitioning algorithm [36]. The optimal number of topics for each conversation is automatically determined by maximizing a clustering objective function proposed in [34].

Topic labeling takes the segmented conversation as input, and generates a set of keyphrases to describe each topic cluster in the conversation. This is done by adapting the co-ranking method proposed in [45], in which a list of the top keyphrases is extracted form a graph of words that captures the co-occurrence of each word in the topic cluster with respect to the words in the leading sentence of that cluster, as well as the position of each word with respect to the thread structure of the conversation.

Creating the topic hierarchy over the collection is the key computational contributions of this paper. Once the sets of topics T_i for each conversation c_i are generated, we organize all of them into a single topic hierarchy to create a structured overview of the whole collection of conversations. To achieve this, we have devised a graph-based method similar to the one that we apply to single conversations. The main difference



(a)

Thread	Sentences
A,C1,C2	s_1, s_2, s_3,s_4 | s_5
A,C1,C3	s_1, s_2, s_3, s_4 | s_6
A,C1,C4,C5	s_1, s_2 | s_3, s_4 |, s_7, s_8
A,C6	s_1, s_2, s_9, s_{10}

(b)

Figure 4. a)Reply-to relationships between the initial post A and the comments $C_1, C_2, ..., C_6$ of a conversation (left). Each post may comprise of one or more sentences as denoted by $s_1, s_2, s_3, ..., s_{10}$. b) the corresponding list of threads along with segmentation results after running the LCSeg algorithm on each of these threads. Here, the segmentation boundary is denoted by '|' (right).

here is that the nodes of the graph we create are not sentences anymore, but topics.

In particular, we create a weighted undirected graph $G(V_C, E_C)$, where the nodes V_C represent the union of all the topics T_i from the set of conversations $C = \{c_1, c_2, ..., c_n\}$ and the edge weight $w(x;y)$ in E_C, between any two given topic nodes x and y, are generated by computing the average similarity between all pairs of sentences, in which one sentence belongs to topic x and the other one belongs to topic y. More formally, consider S_x is a set of l sentences and S_y is a set of m sentences for topics x and y respectively. Then we compute the edge weight $w(x;y)$ as follows:

$$w(x;y) = \frac{1}{l \times m} \sum_{s_i \in S_x, s_j \in S_y} sim(s_i, s_j) \qquad (1)$$

Here, $sim(s_i, s_j)$ is the measure of similarity between a pair of sentences s_i and s_j. This measure is based on cosine similarity between s_i and s_j, if topic x and topic y belong to two different conversations c_x and c_y. Also, the same cosine similarity measure is used when s_i and s_j are from the same conversation, but never appear in the same segment in the segmentation results of the LCSseg algorithm. However, if s_i and s_j are both from the same conversation and they appear together in the same segment at least once, then the similarity is determined by k, where k is the number of times ($k >= 1$) in which s_i and s_j appeared in the same segment. This is based on the intuition that two topics that are from the same conversation and have stronger cohesion in the threads of that conversation should be more likely to be clustered together than those that do not. More formally,

$$sim(s_i, s_j) = \begin{cases} \{CosineSim(s_i, s_j) & \text{if } c_x \neq c_y \\ \begin{cases} k & \text{if } k >= 1 \\ CosineSim(s_i, s_j) & \text{if } k = 0 \end{cases} & \text{else} \end{cases} \qquad (2)$$

$$CosineSim(s_i, s_j) = \frac{\sum_{w \in s_i, s_j} tf_{w,s_i} \cdot tf_{w,s_j}}{\sqrt{\sum_{p \in s_i} tf_{p,s_i}^2} \cdot \sqrt{\sum_{q \in s_j} tf_{q,s_j}^2}} \qquad (3)$$

$$0 \leq CosineSim(s_i, s_j) \leq 1 \qquad (4)$$

Here, $tf_{a,b}$ denotes the term frequency of term a in the sentence b.

Once we have built the graph $G(V_C, E_C)$, we apply the same graph partitioning algorithm used in topic segmentation for single conversation, i.e., approximate solution to n-Cut [36] on $G(V_C, E_C)$. As a result, topic nodes that are mostly similar (i.e., strongly connected in $G(V_C, E_C)$ will form coherent clusters. Each of these clusters can be interpreted as a parent topic (in the topic hierarchy) of all the topic nodes that forms that cluster.

For the final step of topic labeling, we assign a set of keyphrases to each parent topic by taking all the sentences from all the children topic nodes under it, and by then extracting and ranking keyphrases from all those sentences. This process is similar to the topic labeling method described for single conversation, except that given the absence of a thread structure between multiple conversations, we modify the ranking process by creating a graph that only capture word co-occurence relationships.

Sentiment

For sentiment analysis, we apply the Semantic Orientation CALculator (SO-CAL) [39], which has been shown to work well on user-generated content. SO-CAL computes sentiment polarity as numeric values. At first, we generate the polarity for each sentence of the conversation using SO-CAL. We defined five different polarity intervals (-2 to +2), and aggregate the results at various levels. For instance, at the level of a single conversation for each *comment*, we count how many sentences fall in any of these polarity intervals to compute the polarity distribution for that comment. Similarly, when dealing with a set of conversations, for each *conversation* we count how many sentences fall in any of these five polarity intervals to compute the polarity distribution for that conversation.

MULTICONVIS

In order to explore various design choices, we carried out an iterative design process, starting from early mockups and prototypes, to a fully functional system. Throughout this process, we performed formative evaluations [28] to identify potential usability issues and to iteratively refine the prototype. We now present the final design of the MultiConVis interface[1], along with justifications for the key design decisions based on our user requirements analysis and the InfoVis literature.

Visual encoding

Facets: As mentioned earlier, a key design goal of MultiConVis is to facilitate the exploration of a set of conversations at multiple levels of granularity, while maintaining consistent visual mapping across different levels. We maintained consistency in the visual encodings across different levels as follows: 1) Sentiment distributions are represented in the same way (as a stacked bar) for a conversation, for a topic as well as for a comment (see Figure 5(a)). A set of five diverging colors was used in a perceptually meaningful order purple (highly negative) to orange (highly positive) to visualize the distribution of sentiment orientations at all the three different levels of granularity[2]. 2) For all the attributes related to topics/authors facet, the same color coding was used across different levels (see Figure 5(b)).

[1] See also video demo in supplementary material.

[2] The orange and purple colors were selected instead of the standard green and red to avoid the color blindness effects.

(a) Sentiment distribution (b) Topic in four different states with respect to user interaction (c) Conversation

Figure 5. The main visual encodings in MultiConVis: a) Sentiment distribution is shown as stacked bar; b) Visual encoding of topics changes according to different user interactions; c) Visual encoding of a set of aggregated metadata and text analysis results for a conversation.

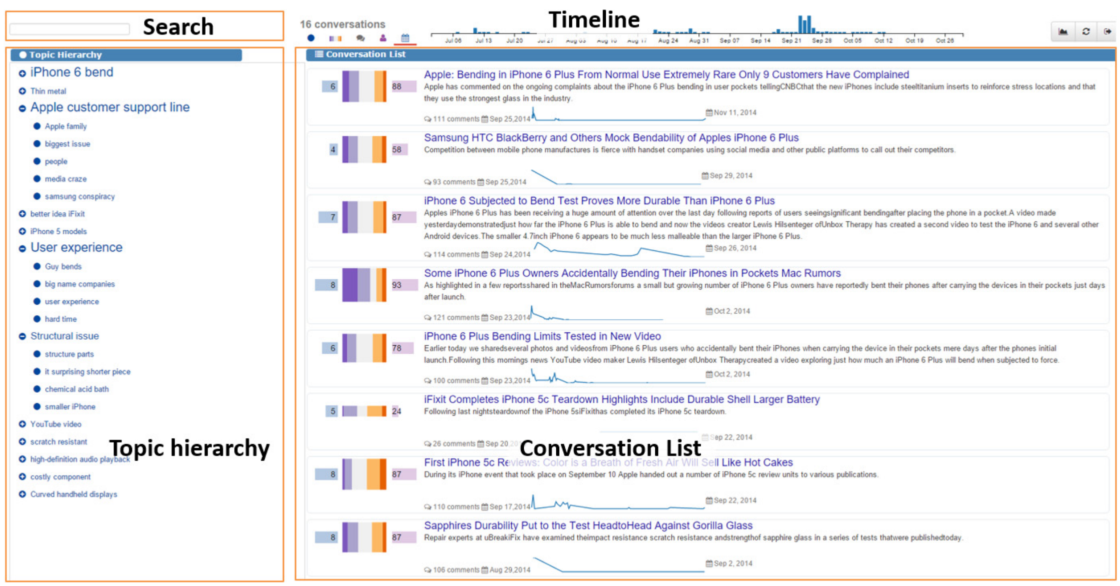


Figure 6. A snapshot of MultiConVis for the 'iPhone bending' dataset: the Topic Hierarchy represents the set of topics and their sub-topics as an indented tree (left); the Conversation List shows a set of aggregated metadata and text analysis results for each conversation (row); the Timeline at the top shows the volume of comments over time for all conversations.

All conversations: Initially, when the user starts exploring the whole collection of conversations MultiConVis displays three components as shown in Figure 6: 1) a Topic Hierarchy; 2) an overview of the set of conversations (Conversation List); and 3) a Timeline View showing the volume of comments of the whole collection over time. These three components are interactively coordinated, so that any operation in one view is reflected in the other views.

The **Conversation List** shows the current set of conversations, where each item in the list represents a set of aggregated metadata and the results of text analysis for the corresponding conversation (See Figure 5(c)). In particular, we encode the following attributes of each conversation: 1) the overall sentiment distribution using a stacked bar, 2) the number of comments, which is encoded as the height of this stacked bar, 3) the count of topics and authors as horizontal bars, and 4) a sparkline that represents the volume of comments over time in a more space efficient way [17]. In addition, the title and a text snippet of the conversation are shown to the right side of its visual summary. Overall, these attributes summarize the set of conversations, facilitating the discovery of interesting subsets of conversations that are of interest to the user.

The **Topic Hierarchy** visually conveys all the topics in the whole collection of conversations using an indented tree representation. Here, topics are sorted chronologically within each level of the hierarchy. Each topic node is represented by its top keyphrase label returned by the topic modeling method, however, when the user hovers on a topic additional keyphrases are also shown to provide more context about that topic. The font size of a topic node represents how much it has been discussed compared to other topics. We present the Topic Hierarchy as an indented tree, where the parent-child relationship is represented by relative vertical position along with horizontal position. We made this choice because an indented tree representation is much more compact than explicitly showing hierarchical links between topic nodes.

Multi-level exploration

From the whole collection to subsets of conversations: While the user initially gets an overview of all the conversations in the collection, her subsequent goal is to find the subset of conversations that are more interesting or relevant, given her current information needs. We support this goal by providing a set of interactive features: linked highlighting, selection, filtering and reordering. The Timeline View, shown in Figure 6, allows the user to quickly filter out conversations that do not fall within the time range in which the discussions were more active or relevant. In addition to filtering, the user can reorder the set of conversations based on the following attributes: number of topics/ authors/ comments, sentiment distribution, and date of the first post of a conversation.

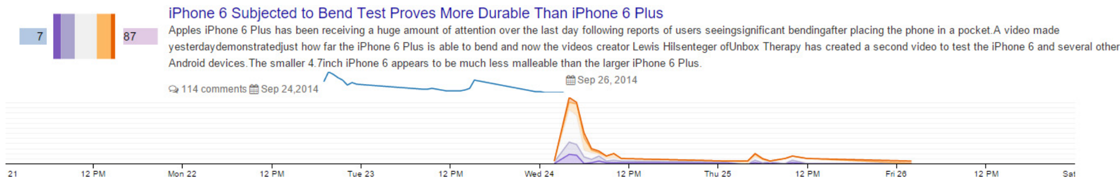


Figure 7. A conversation from the 'iPhone bending' dataset, showing stacked area chart to represent how sentiment distribution evolves over time.

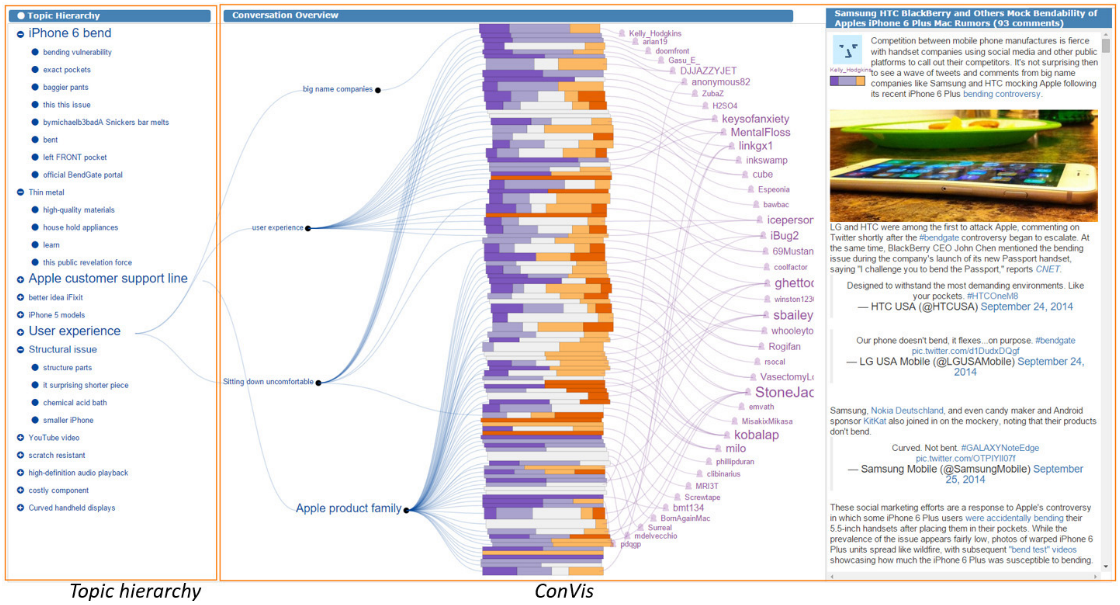


Figure 8. As the user selects a particular conversation, the Conversation List is replaced by the ConVis interface, where the Thread Overview visually represents the whole conversation encoding the thread structure and how the sentiment is expressed for each comment(middle); The Facet Overview presents topics and authors circularly around the Thread Overview; and the Detail View presents the actual conversation in a scrollable list (right). Here, topics a are connected to their related comments as well as to their parents in the Topic Hierarchy via curved links.

To promote exploration based on the topic facet, we provide coordinated highlighting and selection of conversations by topic. For example, hovering on a topic highlights all the conversations where this topic was discussed, and conversely hovering on a conversation temporarily highlights topics in the Topic Hierarchy. Moreover, when the user selects a topic by clicking on it, a vertical outline is drawn alongside the related conversations, allowing the user to see the conversations in which this topic was discussed, even when she is exploring different conversations/ topics. Throughout the filtering and selection processes, the representation of various attributes from both topics and conversations serve as information scent, thus enhancing the ability of the user to navigate and filter data more effectively [42].

Often, as the user finds a subset of conversations that are relevant to her information needs, she may become interested to know more detailed information about them, for instance to see the temporal evolution of sentiment over time for each conversation. We provide such feature based on user interactions, i.e., as the user clicks on the 'Show timeline' button, the sentiment distribution of comments over time is represented as a stacked area chart, within each conversation item in the list (See Figure 7). This helps the user to understand temporal patterns of sentiment in different conversations, supporting her to fulfill information needs related to the time facet.

Drill down to one conversation: As the user continues her exploration, she may become particularly interested in a specific conversation. In this case, she can drill down into that conversation with the ConVis interface, which was designed to explore a single conversation [20]. Here, an important design question arises: once the exploration has reached a single conversation, should we show ConVis along with both the Conversation List and the Topic Hierarchy, so that the user can simultaneously glance at all of them? Notice that showing all the levels would be extremely challenging because of horizontal space limitations. However, we found this not even to be necessary. Our initial formative evaluations and case studies indicate that users do not need to jump back and forth to the Conversation List while exploring a single conversation. On the contrary, users tend to spend most of the time reading specific comments of the conversation they have decided to focus on before going back to the Conversation List. In light of this, when the user drills down into one conversation the Conversation List is replaced with the ConVis interface, as shown in Figure 8.

Now, we briefly describe the visualization components of the ConVis interface, and how they interact with other views of MultiConVis (a more detailed description of ConVis is provided in [20]). ConVis consists of an overview (Thread Overview) of the conversation along with two primary facets, topics and authors, which are presented circularly around this overview. Once ConVis is displayed within MultiConvis, the

Topic Hierarchy over the whole collection is still shown to provide helpful context to the user in understanding the relationship between the topics of the selected conversation and the topics of the other conversations. As shown in Figure 8, the topics of the selected conversation displayed with ConVis are explicitly linked to the ones in the Topic Hierarchy.

The ConVis Thread Overview visually represents the sentiment distribution of each *comment* of the conversation as a stacked bar, similarly to how the sentiment distribution of a *conversation* was represented in the Conversation List. In addition, three different metadata are encoded within the stacked bar: the comment length (height), ordinal position of the comment in the conversation (vertical position), and depth of the comment (horizontal position). 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 [38]. Finally, the Detail View (see Figure 8) 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. If the user clicks on a topic, 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.

As the user drills down to the conversation, she might become interested to know whether similar topics are discussed in other conversations. At any point, the user can look at the Topic Hierarchy to see what are the other similar topics to her current topic of interest, but not discussed in this conversation. For instance, when the user is exploring the topic 'Thin metal' in the current conversation, she may select a related topic labeled 'Structural issue' in the Topic Hierarchy, which results in abandoning the ConVis interface and switching back to the Conversation List, where the conversations related to 'Structural issue' would be highlighted. Finally, at any time the user can return to the Conversation List by clicking on the 'Back' button.

IMPLEMENTATION

The data acquisition, preprocessing, and analysis components were developed using python and a server-side component (in php) which feeds the data to the visualization pipeline. The visual interface was implemented using a combination of HTML, Javascript (using D3, JQuery, crossfiter, and dc.js libraries), so that the tool can be make available as a web application accessible to a large audience.

EVALUATION

We evaluated the MultiConVis interface in two different ways:1) case studies with different domain experts, 2) a formal user study with regular blog readers. While the case studies provided qualitative evidence for the utility of the MultiConVis system, the user study allowed us to compare the system with a traditional interface. Note that ConVis, the interface for single conversations embedded in MultiConVis, had already been evaluated in a previous user study [21], which showed that ConVis outperformed traditional interfaces along several subjective metrics (e.g., usefulness, enjoyable).

Case Studies

We conducted case studies with three users, whose professions are quite diverse, but who come from populations that could all arguably benefit from MultiConVis:

U1: a regular blog reader who visits the Macrumors blog site several times a week. Therefore, he had a genuine interest in exploring the conversations returned by our 'iPhone bending' query. His primary goal was to verify whether the problem of 'iPhone bending' reported by some customers was really serious or not.

U2: a graduate student in the school of Journalism, who contributes to local newspapers about recent political issues. He had strong interest in our dataset about the recent 'ObamaCare health reform'. His primary goal was to understand and summarize the key opinions expressed by the participants about the ObamaCare health reform.

U3: a business analyst in a social media company, where she often needs to analyze a large amount of conversations to understand how customers react to newly released products. So, her goal in the study was to explore conversations about the 'iWatch release' to identify comments that express negative opinions about the product, which is a task that matches what she performs on a regular basis for her company.

For the purpose of case studies, we have collected three different datasets from two different blog sources: Macrumors [3] (a technology-news related blog site dedicated to the discussion of recent news and opinion relating to the Apple Inc) and Daily Kos [1] (a political analysis blog site) between September to December 2014. To create each dataset, we provide a query to the blog site to retrieve the set of conversations that appear on the first page of the search results.

For each case study, we analyzed the results by triangulating between multiple data collection methods, including observations, notes taken by participants during the analysis session, and semi-structured interviews. In addition, we logged interface actions to better understand the usage patterns.

For lack of space, we can only report the primary results of the case studies. The key findings were that: (a) all three users relied on the topic hierarchy to accomplish their task, (b) each user used the hierarchy differently, (c) all users found the topic hierarchy extremely useful. For instance, while the blog reader started his exploration by quickly scanning through the topics in the hierarchy and then going back and forth between topics and conversations, the journalist explored the topics in the hierarchy more systematically, exploring all the comments about one topic before moving to a new one. Still differently, the business analyst started by skimming through the titles of the conversations. But, as she was skimming through the conversations, she also kept an eye on the topics that were highlighted for each conversation in the topic hierarchy. In this way, she identified controversial topics that were intensely debated in recent conversations.

Overall, the semi-structured interviews revealed that users were very satisfied with the interface. In particular, U1 said *"The comments about that chemical acid bath was buried down in the middle of one conversation, which I don't think I would have noticed with a regular interface. Using MultiConVis, I was able to pick this topic from the hierarchy and then jumped into the related comments without having to read the entire conversations....".* U2 found the topic hierarchy to be very helpful in supporting a systematic exploration of the conversations by organizing the key opinions into meaningful topical groups. More interestingly, he realized the potential utility of MultiConVis system for other exploratory tasks that he would like to perform, *"This tool could be not only useful when I want to write a story, but also to prepare for interviewing a policy maker, or a politician by quickly understanding what topics are triggering the most interesting or controversial discussions in the public spheres."* Finally U3 anticipated that this tool could be very useful to understand what features of their products worked (or didn't work) and then revise the products accordingly, *" The MultiConVis interface would definitely help me to understand the requirements and needs of my customers more effectively. Our current way is just to skim through the comments, often missing the important feedback from customers ...but this interface can help me identify what are the biggest concerns from the customers and get clues about the ways to satisfy their needs."*

User study

We run a formal user study to evaluate the efficacy and usability of the MultiConVis interface compared to an interface that represents the traditional interfaces for blog reading. The aim of the user study is to answer the following two questions: (1) When we compare MultiConVis with the traditional interface for exploring a set of conversations, is there any difference in user performance and subjective reactions? (2) What specific features of the MultiConVis interface are perceived as more/less beneficial by the potential users (e.g., Topic Hierarchy, Timeline etc.)?

Methodology

Since the first research question requires comparisons among two different user interfaces, we conducted a summative evaluation through controlled experiments [28]. The study was designed with two interfaces as conditions: a) the traditional interface for blog reading, and b) *MultiConVis*. Here, the traditional interface shows a set of blog conversations as a linear list, where each item represents a set of metadata of the conversations (e.g., title, number of comments, and posting date). The user can click on any conversation in the list, which results in showing all the comments of that conversation using an indented list representation. In addition, we provided a set of interactions that are common in most blog reading interfaces, i.e., searching for terms and sorting conversations by attributes (e.g., number of comments). A within-subject design was used with interface as the within-subject factor, allowing us to directly compare the measures of each participant with respect to both interfaces. Finally, all study aspects, including instructions and setup, went through several iterations of evaluation and pilot testing.

Task and procedure

At first, a pre-study questionnaire was administered to capture demographic information and prior experience with exploring blog conversations. Then, the participant went through the following steps for each of the two interfaces: 1) In a scripted warm-up session, the interface was introduced to the participant using a sample dataset. 2) The participant was then asked to perform a task based a set of conversations. For each interface, a different set of conversations was provided.

Task: Considering the open-ended nature of blog reading, no specific set of questions was given. Instead, the participant was asked to explore a set of conversations about the given query and then write a single summary of what she thought were the major discussion points and most insightful comments within the conversations. The study lasted approximately 60 minutes and each participant was paid \$15 to participate.

We selected two different datasets crawled from the Macrumors site for testing ('iphone bend' and 'ipad release'). The number of conversations in the datasets are kept the same (16 conversations in each dataset) to avoid potential variations due to the amount of conversational data. Also, to counterbalance any potential learning effects due to the order of exposure to specific interfaces and dataset, the order was varied using a 2 x 2 Latin square. During the study, we collected both quantitative data such as task completion time and qualitative data such as observations and questionnaires. Finally, a post-study questionnaire followed by an semi-structured interviews were administered regarding the user's experience with two interfaces.

Participants

We conducted the study with 16 users (aged 18-37, 6 females) who have considerable experience of reading blogs. The participants held a variety of occupations ranging from journalists, engineers, system analysts and students from both graduate and undergraduate levels. They were recruited through emails and social networks (Facebook and reddit posts).

Results analysis

After completing the task with each interface, participants rated six different measures in the form of in-study questionnaires. Since these measures were rated using a standard 5 point Likert scale, standard parametric analysis was not suitable due to the lack of normality [25]. Instead we performed nonparametric analysis i.e., Mann-Whitney's U tests on the responses for each of these measures.

The results of these questionnaires are presented in Figure 9. The pairwise comparisons using Mann-Whitney's U tests indicate that MultiConVis was superior on five different measures out of six: usefulness ($Z = -1.823; p < .05$); enjoyable to use ($Z = -3.697; p < .01$); find insightful comments ($Z = -3.95; p < .01$); find major points ($Z = -2.909; p < .01$); and

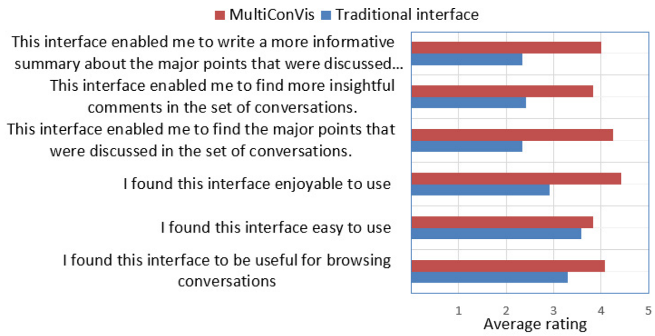


Figure 9. Average rating of interfaces by the participants on six different measures. Longer bars indicate higher rating.

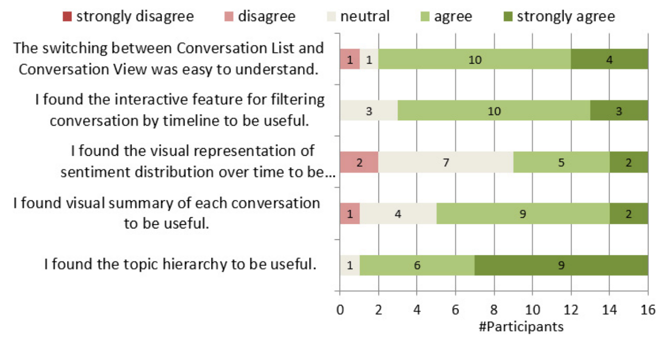


Figure 10. Responses to statements regarding specific features of the MultiConVis interface.

enable to write more informative summary ($Z = -3.915; p < .01$). For the other measure i.e., ease of use, MultiConVis was still rated higher over the traditional interface, however the results was not significant. This is interesting, because MultiConVis appears to be as easy to use as the other interface in spite of its complex interface features.

Interface features: Each participant was also asked a set of questions regarding the usefulness of specific features of the MultiConVis interfaces. From Figure 10, we can readily see that the majority of the responses were dominated by positive ratings. Among the interface features, the Topic Hierarchy received the most positive ratings (strongly agree:9, agree:6), followed by the visual summary of each conversation, and interactive filtering by timeline.

Time: The average time required to complete the tasks was not significantly affected by the interfaces, with MultiConVis and the traditional interface requiring 1065 ± 249 and 1029 ± 204 secs respectively.

Overall Preference: In the post-study questionnaire, participants were asked which system they prefer for exploring a collection of conversations. 75 % of the participants indicated a preference for MultiConVis, whereas 25 % preferred the traditional interface. Many of the participants who chose MultiConVis indicated that the utility of Topic Hierarchy was the primary reason for their preference: *"By having a topic hierarchy of the relevant topics, as well as highlighting which conversation refers to which topic, it was very easy to filter out the blogs that were not relevant.'* (P8). They also found the visual summary provided for each conversation was very useful, *"The summary offered by this visualization is quite impressive and throws a lot of instant information."* (P2). Additionally, for the sentiment distribution over time *"...made it very easy to see how opinions changed over time. While investigating bend gate it was clear how the community opinion changed after the event had played out in the media"* (P4).

Those who preferred the traditional interface indicated that they like its familiarity *"I preferred the older style of interface mainly because it's what I'm more familiar with..."* (P1). They also pointed out that sometimes the topic hierarchy was inaccurate (e.g., topic labels did not always make sense to them) *"..maybe with better tagging I'd find it (MultiConVis) more useful..."* (P1), and *"the keywords weren't necessarily the most useful ones or the relevant ones"* (P5).

DISCUSSION

After conducting the evaluation with real users, we reflect upon our visualization design in order to understand to what extent our system can support the kind of tasks the users wanted to perform. Our case studies demonstrate that the system can be useful in a variety of contexts of use, while the formal user study provides evidence that the MultiConVis interface supports the user's tasks more effectively compared to traditional interfaces. In particular, all our participants, both in the case studies and in the user study, appear to greatly benefit from the topic hierarchy and the high-level overview of the conversations. The user study also shows that the MultiConVis interface is significantly more useful than the traditional interface, enabling the user to find insightful comments from thousands of comments, even when they were scattered around multiple conversations, often buried down near the end of the threads. Remarkably, MultiConVis was preferred by the majority of the participants over the traditional interface, suggesting the potential value of our approach for combining NLP and InfoVis.

CONCLUSIONS AND FUTURE WORK

MultiConVis is an interactive visual text analytics system for exploring a collection of blog conversations. Unlike traditional systems, MultiConVis takes the unique characteristics of online conversations into account to tightly integrate NLP and InfoVis techniques. The resulting visual interface aggregates data across different levels, supporting a faceted exploration starting from a whole set of conversations, to a subset of conversations, to one conversation.

We would like to extend our work along the following avenues. First, we would like to explore the possibility of applying higher level argumentation analysis of the conversations [5] by extracting the discourse structure of the posts [23], to better support the user in understanding the opinions expressed about each topic. Second, even though the topic hierarchy was found to be very useful, still in a few cases the extracted topics were either noisy or did not match the user's current information needs. To deal with this problem, we aim to extend the recently proposed interactive topic modeling approach for single conversations [21], to handle a large collection of conversations. Finally, to increase the ecologically validity of our evaluations [7], we would like to perform longitudinal studies to observe how the system is used by real users to satisfy their information needs over an extended period of time.

REFERENCES

1. Daily Kos. (????). `http://dailykos.com/`.
2. Illinois Part of Speech Tagger. (????). `http://bit.ly/1xtjFHe`.
3. Macrumors. (????). `http://macrumors.com/`.
4. Slashdot. (????). `http://slashdot.com/`.
5. Kelsey Allen, Giuseppe Carenini, and Raymond T Ng. 2014. Detecting Disagreement in Conversations using Pseudo-Monologic Rhetorical Structure. In *Proceedings of the EMNLP*.
6. Eric Baumer, Mark Sueyoshi, and Bill Tomlinson. 2008. Exploring the role of the reader in the activity of blogging. In *Proc. Conf. CHI*. 1111–1120.
7. Sheelagh Carpendale. 2008. Evaluating information visualizations. In *Information Visualization*. 19–45.
8. Weiwei Cui, Shixia Liu, Zhuofeng Wu, and Hao Wei. 2014. How Hierarchical Topics Evolve in Large Text Corpora. *IEEE Trans. Visualization and Computer Graphics* 20, 12 (2014), 2281–2290.
9. Kushal Dave, Martin Wattenberg, and Michael Muller. 2004. Flash forums and forumReader: navigating a new kind of large-scale online discussion. In *Proc. ACM Conf. on CSCW*. 232–241.
10. Munmun De Choudhury and Hari Sundaram. 2011. Why do we converse on social media?: an analysis of intrinsic and extrinsic network factors. In *Proceedings of the 3rd ACM SIGMM international workshop on Social media*. ACM, 53–58.
11. Nicholas Diakopoulos, Mor Naaman, and Funda Kivran-Swaine. 2010. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*. IEEE, 115–122.
12. Judith Donath, Karrie Karahalios, and Fernanda Viï£¡gas. 1999. Visualizing Conversation. *Journal of Computer-Mediated Communication* 4, 4 (1999), 0–0.
13. Marian Dork, Sheelagh Carpendale, Christopher Collins, and Carey Williamson. 2008. Visgets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Trans. Visualization & Comp. Graphics* 14, 6 (2008), 1205–1212.
14. Marian Dork, Daniel Gruen, Carey Williamson, and Sheelagh Carpendale. 2010. A visual backchannel for large-scale events. *IEEE Trans. Visualization and Computer Graphics* 16, 6 (2010), 1129–1138.
15. Wenwen Dou, Li Yu, Xiaoyu Wang, Zhiqiang Ma, and William Ribarsky. 2013. HierarchicalTopics: Visually Exploring Large Text Collections Using Topic Hierarchies. *IEEE Trans. Visualization & Comp. Graphics* 19, 12 (2013), 2002–2011.
16. Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *Proc. Annual Meeting on Association for Computational Linguistics*. 562–569.
17. P. Goffin, W. Willett, J.-D. Fekete, and P. Isenberg. 2014. Exploring the Placement and Design of Word-Scale Visualizations. *IEEE Trans. Visualization & Computer Graphics* 20, 12 (Dec 2014), 2291–2300.
18. S. Havre, E. Hetzler, P. Whitney, and L. Nowell. 2002. ThemeRiver: visualizing thematic changes in large document collections. *IEEE Trans. Visualization & Comp. Graphics* 8, 1 (2002), 9–20.
19. Marti A Hearst, Matthew Hurst, and Susan T Dumais. 2008. What should blog search look like?. In *Proc. ACM Workshop on Search in Social Media*. 95–98.
20. E. Hoque and G. Carenini. 2014. ConVis: A Visual Text Analytic System for Exploring Blog Conversations. *Computer Graphics Forum (Proc. EuroVis)* 33, 3 (2014), 221–230.
21. E. Hoque and G. Carenini. 2015. ConVisIT: Interactive Topic Modeling for Exploring Asynchronous Online Conversations. In *Proc. ACM conf. on Intelligent User Interfaces*.
22. Indratmo, Julita Vassileva, and Carl Gutwin. 2008. Exploring blog archives with interactive visualization. In *Proc. working Conf. on Advanced visual interfaces*. 39–46.
23. Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Yashar Mehdad. 2013b. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In *Proc. Annual Meeting on Association for Computational Linguistics*.
24. Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2013a. Topic Segmentation and Labeling in Asynchronous Conversations. *Journal of Artificial Intelligence Research* 47 (2013), 521–573.
25. Maurits Clemens Kaptein, Clifford Nass, and Panos Markopoulos. 2010. Powerful and Consistent Analysis of Likert-type Ratingscales. In *Proc. Conf. CHI*. 2391–2394.
26. B. K. Kaye. 2005. Web side story: An exploratory study of why weblog users say they use weblogs *(AEJMC Annual Conf.)*.
27. Daekyung Kim and Thomas J. Johnson. 2012. Political blog readers: Predictors of motivations for accessing political blogs. *Telematics and Informatics* 29, 1 (Feb. 2012), 99–109.
28. H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. 2012. Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Trans. Visualization & Comp. Graphics* 18, 9 (2012), 1520–1536.
29. Shixia Liu, Xiting Wang, Jianfei Chen, Jun Zhu, and Baining Guo. Topicpanorama: a full picture of relevant topics. (????).

30. Craig Macdonald, Rodrygo L.T. Santos, Iadh Ounis, and Ian Soboroff. 2010. Blog track research at TREC. *SIGIR Forum* 44, 1 (2010), 58–75.
31. Adam Marcus, Michael S Bernstein, Osama Badar, David R Karger, Samuel Madden, and Robert C Miller. 2011. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proc. CHI*. ACM, 227–236.
32. Gilad Mishne. 2006. Information access challenges in the blogspace. In *Workshop on Intelligent Information Access (IIIA)*.
33. Sean A Munson and Paul Resnick. 2010. Presenting diverse political opinions: how and how much. In *Proc. Conf. CHI*. 1457–1466.
34. Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E* 69, 2 (2004), 026113.
35. Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations. In *Proc. ACL*. 172–180.
36. Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.
37. Param Vir Singh, Nachiketa Sahoo, and Tridas Mukhopadhyay. 2010. Seeking variety: A dynamic model of employee blog reading behavior. *Available at SSRN 1617405* (2010).
38. Markus Steinberger, Manuela Waldner, Marc Streit, Alexander Lex, and Dieter Schmalstieg. 2011. Context-preserving visual links. *IEEE Trans. Visualization & Comp. Graphics* 17, 12 (2011), 2249–2258.
39. Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics* 37, 2 (2011), 267–307.
40. Fernanda B Viégas, Scott Golder, and Judith Donath. 2006. Visualizing email content: portraying relationships from conversational histories. In *Proc. Conf. CHI*. 979–988.
41. Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. 2010. Tiara: a visual exploratory text analytic system. In *Proc. ACM Conf. on Knowledge Discovery and Data Mining*. 153–162.
42. Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. 2007. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Trans. Visualization & Computer Graphics* 13, 6 (2007), 1129–1136.
43. Y. Wu, S. Liu, K. Yan, M. Liu, and F. Wu. 2014. OpinionFlow: Visual Analysis of Opinion Diffusion on Social Media. *IEEE Trans. Visualization and Computer Graphics* 20, 12 (Dec 2014), 1763–1772. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346920
44. J. Zhao, N. Cao, Z. Wen, Y. Song, Y. Lin, and C. Collins. 2014. #FluxFlow: Visual Analysis of Anomalous Information Spreading on Social Media. *IEEE Trans. Visualization & Computer Graphics* 20, 12 (Dec 2014), 1773–1782.
45. Ding Zhou, Sergey A Orshanskiy, Hongyuan Zha, and C Lee Giles. 2007. Co-ranking authors and documents in a heterogeneous network. In *Seventh IEEE International Conf. on Data Mining*. 739–744.
46. Aaron Robert Zinman. 2011. *Me, myself, and my hyperego: understanding people through the aggregation of their digital footprints*. Ph.D. Dissertation. MIT.