ConVisQA: A Natural Language Interface for Visually Exploring Online Conversations

Nadia Siddiqui
School of Information Technology
York University
Toronto, Canada
nasidiqi@yorku.ca

Enamul Hoque
School of Information Technology
York University
Toronto, Canada
enamulh@yorku.ca

***Abstract*—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. Traditional sites present a conversation in a paginated list view, making it very difficult to find comments of interests about a specific topic and/or opinions which may be scattered around a long thread of discussion. In this paper, we introduce a natural language interface that supports the user to quickly locate and browse through the comments that are relevant to her information needs. Our system takes a question asked by the reader about a conversation as input and then automatically finds the answer using natural language processing techniques. It then presents the results by highlighting in a visual interface, enabling the user to quickly navigate through the comments that match her information needs. Our case studies with three users suggest that the system can help the user to effectively fulfill her information needs by highlighting the relevant comments to their question.**

***Index Terms*—Online Conversations, Natural Language Interface, Information Visualization, Text Analysis, Parser, Context Free Grammar**

I. Introduction

With the proliferation of web-based social media, asynchronous conversations have become very common for supporting online communication and collaboration. An online conversation such as a blog may start with a news article or an editorial opinion, but it can quickly become very long with hundreds of comments [2]. Traditional blogs and social media sites present a post and subsequent replies as a paginated indented list without providing any high-level summary of a conversation. Going through such a large amount of conversational data often lead to information overload, where the reader gets overwhelmed, starts to skip comments, and eventually leaves the conversation without satisfying her information needs [13].

Recent research has attempted to address this problem by combining automatic text analysis and interactive visualizations [6]. For instance, several works automatically extracted the topics of discussions and sentiments from a conversation and then visually represented such information to help the user in getting an overview of the conversation and exploring the topics of interests [4] [6]. When these topics are presented within an interface, the user can select the one(s) she is interested in and then quickly navigate through its related comments. In addition, if information about sentiment (e.g., positive vs negative) is visually encoded along with the topics, then the user can assess what comments were in favor or against a particular opinion.

While visual interfaces for exploring conversations have been found useful by users for various high-level tasks [4] [6], they may not always provide effective ways to fulfill specific information needs. For instance, consider John who is reading a blog conversation from Slashdot, a technology-related social news site. The blog discusses how some hackers have got access to US army servers. John is interested in knowing answers to several specific questions that he has in mind. For instance, which topics have generated the most discussions? Which topic has generated the most controversial discussion? Who are the sources of most negative/positive comments on a topic? However, to get answers to these questions he needs to navigate through many topics and comments which can become tedious and time-consuming.

To address the problem, we introduce a natural language interaction technique as a complementary way along with traditional interactive visualization techniques for making sense of large conversations. The resulting system, named ConVisQA, allows the user to express her specific information needs about an online conversation through natural language (see Figure 1). In response, the system uses natural language processing (NLP) techniques to automatically parse the questions to identify the relevant topics, sentiments, comments and participants of a conversation and highlight them to the user. This enables the user to quickly locate the relevant information to her given question so that she can fulfill her information needs more effectively.

The primary contributions of this paper are:

i) We developed a natural language interaction system for exploring online conversations that allows the user to ask questions about the conversations and get the answer presented in a visual interface. To the best of our knowledge, this is the first attempt of building a natural language interaction system for exploring and making sense of online conversations.

ii) We incorporated our natural language interaction technique by extending ConVis [6], a visual interface that was

Identify applicable funding agency here. If none, delete this.

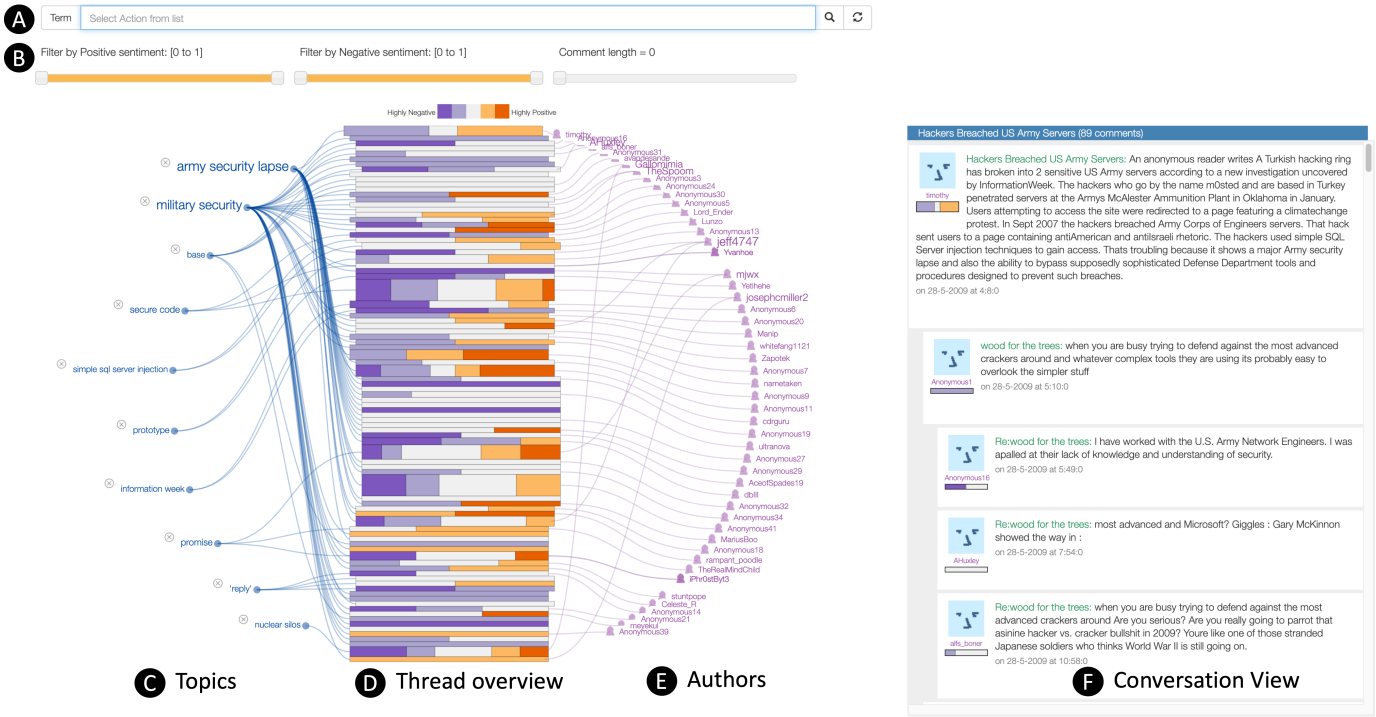


Fig. 1. The ConVisQA user interface allows users to explore a blog conversation and get their questions answered about that conversation. Here, the user can ask a question using a textbox (A) and can filter comments based on sentiment and comment length (B). The Thread Overview visually represents all the comments using stacked bars (D) while topics and authors are arranged circularly around this overview (C E). Finally, the Conversation View at the right presents the comments in a scrollable pane (F).

originally developed for visually exploring conversations.

iii) Our case studies with three participants demonstrated how the ConVisQA interface may be helpful for quickly locating comments of interests to fulfill specific information needs that the user may have.

II. Related Work

A. Visualizations for Exploring Conversations

Early work on visualizing online conversations has mostly focused on visualizing 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 [27], thumbnail metaphor using a sequence of rectangles [3], [30], and radial tree layout [20]. However, such visualizations do not analyze the actual content (i.e., the text) of the conversations.

In contrast, those that analyze the textual content of conversations primarily focused on summarizing and showing the major topic of conversations [23], [3], or visualizing the content evolution over time [31], [28]. For example, ConVis [6] and ConVisIT [7], [8] facilitates multi-faceted exploration of a blog conversation based on topics, authors, and sentiment. It also provides various interaction techniques such as highlighting based on multiple facets to support the user in exploring and navigating the conversation. MultiConVis extended this interface by supporting the exploration of a set of conversations [9]. ConToVi [4] visualizes speakers' dynamics with regard to different topics in conversations like political debates using animations using radial visualization. It also displays the speaker's behavior using categories like sentiment, politeness, and eloquence. TopicPanorama visualizes topics that are discussed in multiple sources such as news, blogs, and micro-blogs. The key functionality in Topicpanorama is to jointly match the topics extracted from each source together in order to interactively and effectively analyze common and distinctive topics in all sources [29].

While the above visualizations techniques have shown Visualizations for Exploring Conversations promises in providing an overview of the conversations and interactive features for navigating through the comments the user may find it still difficult and time consuming to locate the comments of interests related to a specific question (e.g. "Which comments are saying negative about Obama Care"?) using such visual interfaces.

B. Natural Language Interactions with Data

Natural language interfaces for data visualization have received considerable attention recently [11]. Typically, these interfaces respond to user queries by either creating a new visualization (DataTone [5]) and/or by highlighting answers within an existing visualization (Eviza [24]). Some systems enable follow-up data queries from users with limited support for pragmatics (e.g. Evizeon [11], Orko [26]). Some of them provide query auto-completion features either by supporting syntactic query formulation [24] or by supporting information recall and data preview [25].

Generally, these systems recognized the importance of providing feedback on how the system interprets queries and enabling users to correct misunderstandings through interface widgets. However, most of these works largely depend on heuristics for parsing which are incapable of handling questions that are compositional or otherwise incomplete and ambiguous.

There has been a recent surge in research on conversational interfaces [12], one of the avenues of such research is automatic question answering. For instance, some works focus on answering questions with semi-structured table using semantic parsing techniques [21]. More recently, Kim et al. extended such technique to answer questions about a visualization by applying the algorithm on the underlying data table of that visualization [14]. However, the above works have mainly focused on interacting with tabular data whereas the conversational data used in our system is textual.

III. ConVisQA

We now present our user requirements analysis which informed our system design as well an overview of our ConVisQA system. Finally, we demonstrate how the natural language interaction and other features of ConVisQA support users to fulfill information needs while exploring conversations.

A. User Requirements Analysis

In order to guide our system design, we perform an initial formative study with 3 users (2 females and one male, age range 21-35 yrs) who regularly read blogs. During the study, the participants were asked to read some given blog conversations according to their own needs and interest and then write a set of questions that come to their mind. We also requested them to suggest any improvements in the system design. In addition, we rely on previous literature review of why and how people read blogs [6] to get a sense of what kind of information needs they have in mind.

Through the formative study and the literature review to compile a list of questions that people may ask naturally, given a blog conversation they are reading. We used this list to inform the design of our prototype, including the most common types of questions they ask as well as what kind of keywords they use and the grammatical structures their questions usually follow. Table I shows a set of example questions that people typically ask while exploring conversations. We also identify what kind of data variables are involved in each of the questions and the analytical functions that are necessary to answer these questions.

During the initial formative study, P1 suggested various questions about authors who post comments in the blog conversation e.g which authors always post controversial comments? P2 suggested to have sentiment filters for positive and negative comments. She also added that it will be better to cluster comments based on similar content or a key phrase. P3 reported "that there should be a guide or tool-tip on each facet such as topic or author to guide the users what to do. She found it difficult to understand without demonstration of system functionality".

B. System Overview

Fig. 2 presents an overview of our system, which is organized in two parts. In the offline step (Fig. 1a) we preprocess the set of conversations collected from different blog sites (e.g. Slashdots, Macrumors) by cleaning the data 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 [17] to tokenize the text and annotate the tokens with their part-of-speech tags. After that, the conversation analyzer module performs topic modeling and sentiment analysis over the whole set of conversations and store the results in a database (see [6] for more details). We also determine the level of controversy for each topic using a supervised classifier [1]. Finally, the results are stored in a database for efficient retrieval.

When the user types a question (e.g. "Which one is the most controversial topic of discussion?"), the system performs the following three steps on the fly:

- lexical and semantical analyzer pre-process the questions and re-write the original input question so that the parser can recognize the question.
- The ANTLR parser takes the modified question and uses a set of pre-defined grammar rules to automatically recognize the entities and analytical functions involved in the question to build a query for retrieving results from the database.
- The ConVisQA Interface obtains the results from the database and presents the results to the user by highlighting the results in the visualization as well as by using text.

We briefly discuss the above steps below.

C. The Lexical and Semantic analyzer

This module first tokenizes the question, removes the stop words, and applies part of speech tagging using Stanford CoreNLP [17]. It then finds the terms that are mentioned in the input questions but do not exist in our pre-defined lexicon. For this purpose, it applies the word2vec model [18] to check if the user enters a term that is not in our lexicon, but it is most similar to a term in our lexicon. In such cases, we replace the user's term with the term in our lexicon so that the parser can recognize it.

D. The ANTLR Parser

We use an ANTLR parser that employs a top-down parsing strategy named LL(*) [19]. We choose ANTLR parser because it allows for greater flexibility in specifying the grammar rules and has been successfully applied in the natural language interface for data visualizations recently [22]. This parser reads the input from left to right, performing the leftmost derivation of the input query. For this purpose, it employs a context-free grammar with a set of production rules. We designed the hand-crafted grammar rules based on prior analysis on

TABLE I

A SET OF EXAMPLE QUESTIONS ALONG WITH DATA VARIABLE AND ANALYTICAL FUNCTIONS THAT ARE INVOLVED IN THE QUESTIONS.

Sr.	*Question*	*Data Variables*	*Analytical Functions*
1	What is the most/least controversial topic?	Topics, comments	Find extrema
2.	What is the most/ least controversial comment?	Author, Comment	Filter, Find extrema
3.	What is the most/ least controversial comment on topic X?	Topic, Comment	Filter, Find extrema
4.	Who is the most/least controversial author?	Author, Comment	Filter and Find extrema
5.	What is the most/least controversial comment?	Comment Filter	Find extrema
6.	What is the most/ least controversial comment of an author?	Comment Author	Filter, Find extrema
7.	Who had posted the most negative/positive/ neutral comments about topic X?	Author, Topics, Comments	Filter, Find extrema
8.	Who was the most dominant participant of the conversation?	Author, Comments	Find extrema
9.	Can you get rid of topic X?	Topic, comments	Filter
10.	Which topics are generating more discussions?	Topic, comments	Sort

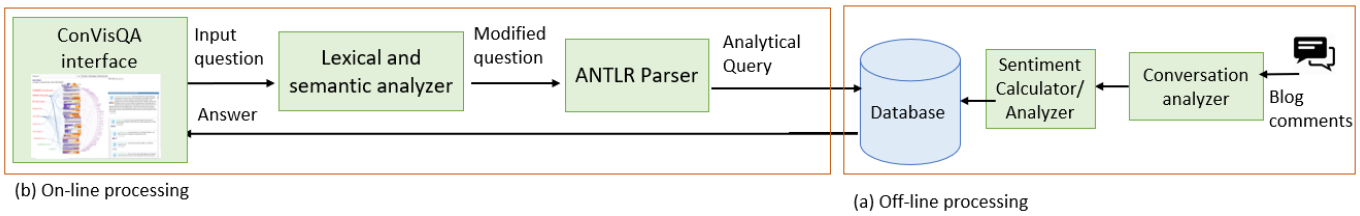


Fig. 2. Overview of our ConVisQA system for supporting question answering with online conversations

how people may ask different analytical questions while they explore conversations [6]. In particular, we first identified and defined a set of analytical functions (e.g., sort, filter, find extremum, compare) as well as a set of data variables that are involved in a question (e.g. topic, author, comment, sentiment, posting time). Table 1 shows some example questions along with the corresponding analytical functions and data variables that are involved with them.

In the next step, we use the data variables and analytical functions to build grammar rules that the ANTLR parser can process. For instance, consider the following production rule:

$$G \wedge 0^{\rightarrow}.* < extreme >< sentiment > \quad (1)$$

$$Comment\ about < topic >?$$

In this rule, $< extreme >$ is a non-terminal symbol that can represent analytical tokens like most, least etc. The $< sentiment >$ symbol can take values like 'negative', 'positive' and 'neutral'. The $< topic >$ symbol can take any topic name in the conversation. Given the question "Which one is the most negative comment about army security lapse?", the parser generates a query that finds all the comments about the topic 'army security lapse' from the database and then selects the comment that has the most negative score.

IV. THE CONVISQA INTERFACE

We now demonstrate how the ConVisQA interface helps the user in exploring conversations by using natural language question answering[1]. We first describe the visual encodings and interactive features of ConVisQA first and then how the user performs exploration of a conversation via question answering.

A. Visual Encodings and Interactive Features

We developed ConVisQA by extending ConVis [6], a visual interface for exploring an online conversation which was originally designed based on requirements of the blog reading tasks. The ConVisQA interface is primarily designed as an overview+details interface as shown in Figure 1. The overview consists of the whole conversation thread as well as the discussion topics and authors who participated in 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 as well as depth of the comment within the thread).

A set of five diverging colors is used to visualize the distribution of sentiment orientation of a comment, ranging from purple (negative polarity) to orange (positive polarity). 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 cyclically around the Thread Overview.

By default, 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. However, such ordering may change based on certain questions (e.g. if the user asks to sort topics based on how controversial there are). To indicate

[1]A video demo of the *ConVisQA* system is available at: https://www.yorku.ca/enamulh/video/ConVisQA.mp4

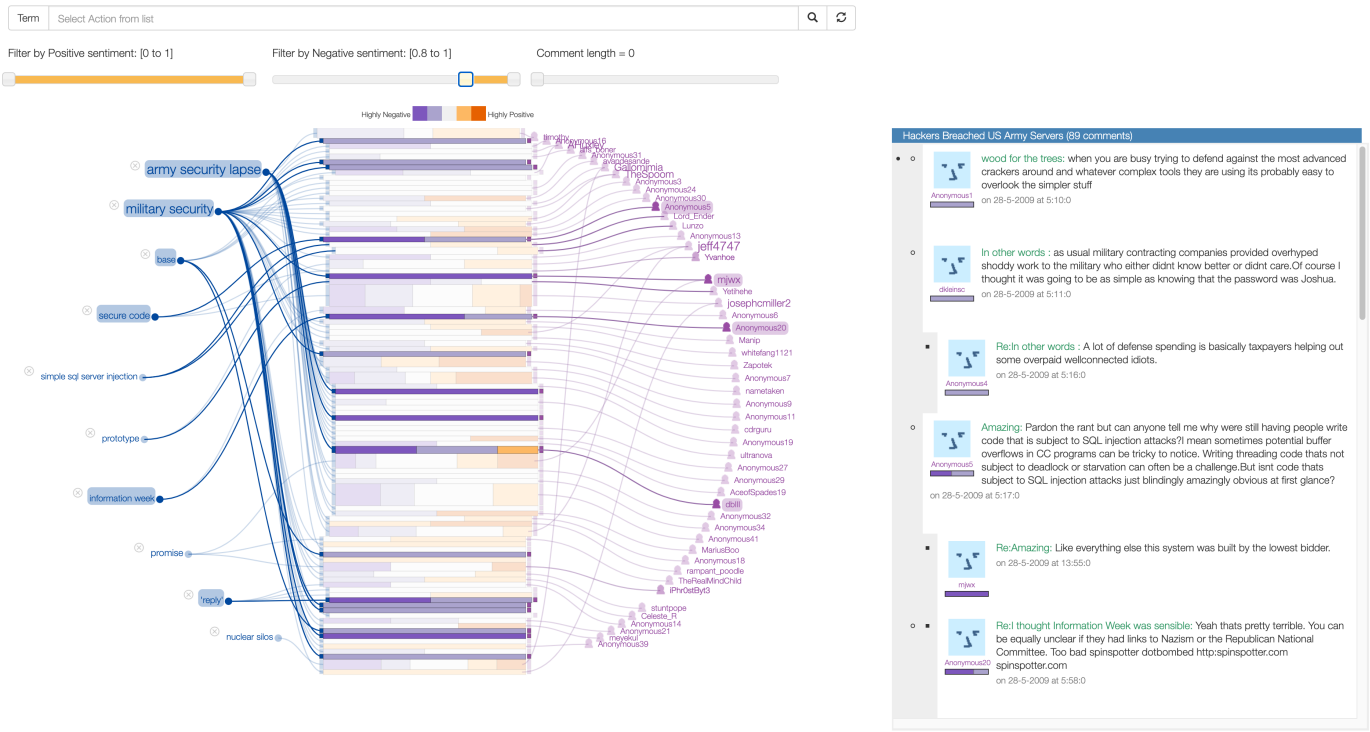


Fig. 3. ConVisQA supports the user to filter comments by sentiment and comment length. Here, the user uses a slider to select highly negative comments which are highlighted in the Thread Overview.

the topic-comment-author relationship, the facet elements are connected to their corresponding comments in the Thread Overview via subtle curved links. Finally, the Conversation View displays the actual text of the comments in the discussion as a scrollable list (Fig 1, right).

We also made some extensions to the ConVis interfaces by including additional interactive features for filtering, sorting and deletion. For example, the user can filter comments based on sentiment score (see Figure 3) and the comment length i.e. number of sentences in a comment. The user can also sort comments by comment length. They can delete topics as well whenever they find some of them as less interesting or irrelevant.

B. Interactive Question Answering

ConVisQA allows the user to ask questions by typing in the query box at the top of the interface. As the user starts typing the system parses the current query input and provide the possible query suggestions that may follow the current input. Internally, the system uses the ANTLR parser to check what are the possible valid parse trees given the current input and the grammar that we have designed. The user can select only question from the dropdown list at any point (Fig 4a).

Once the user provides a query string and clicks on the search button, the system analyzes the query and then automatically presents the answer by highlighting or filtering the relevant facets e.g. topics, comments or authors in the overview as well as in conversational view (Figure 4b–d). For instance, given the question "What topic is generating more discussions?", the system finds that 'army security lapse' topic has the most number of comments therefore it highlights this topic as well as relevant comments and authors (Figure 4b). This response to the user's question is also mirrored in the Conversation View which automatically scrolls to the corresponding comment for immediate access. In this way, the user can locate the comments that match her information needs without having to navigate through all the comments.

In addition to question answering, ConVisQA also supports users to search through various facets such as comments, author or comment+author and system locates and highlights the comments that matched the search criteria. For example, if the user searches for the words 'security', the system highlights all the comments that match this word in the Thread Overview as well as in the Conversation View.

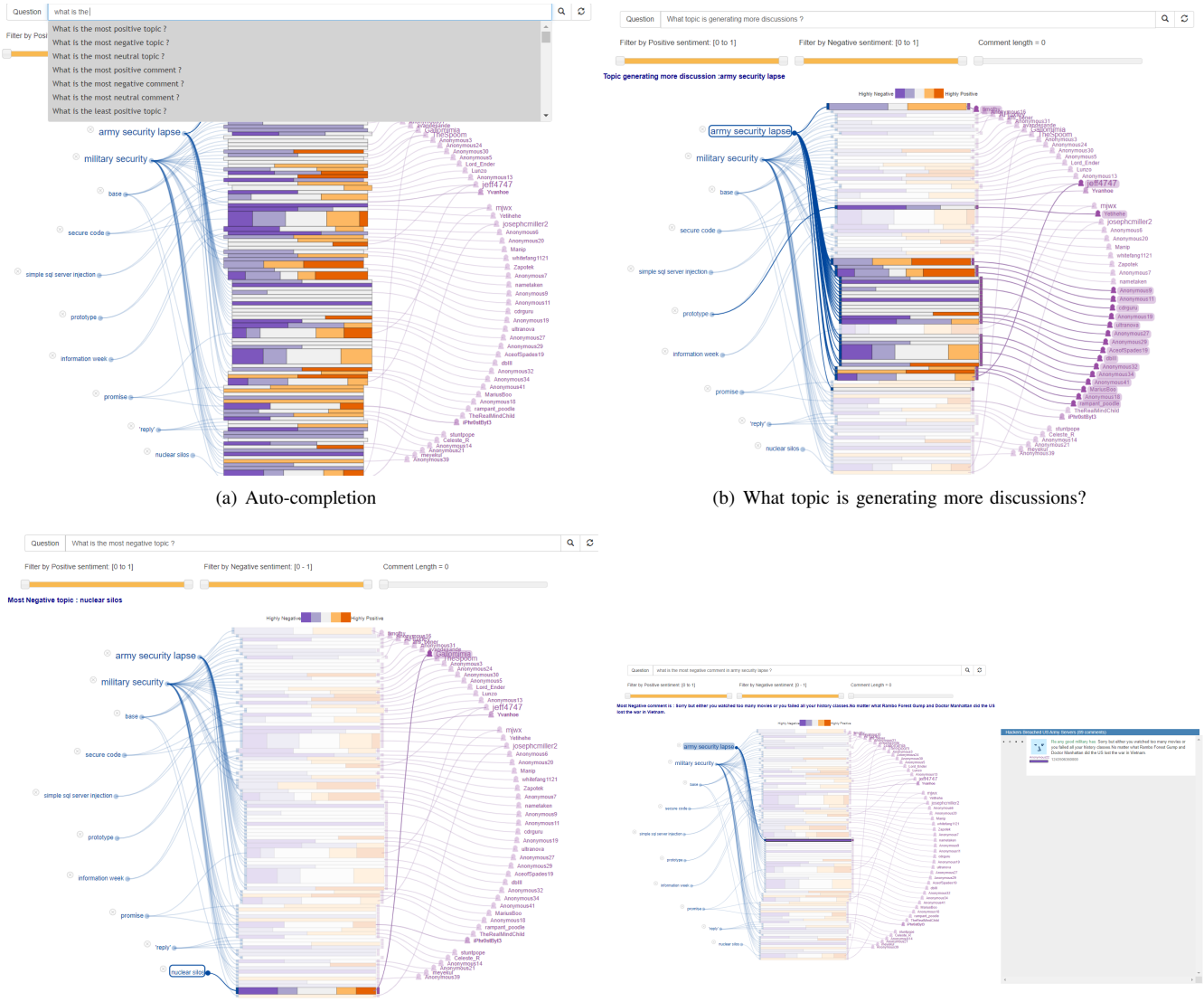
V. Technical Implementation

A server-side component (in PHP) retrieves conversations which are annotated with topics, comments and sentiment scores. The visualization component is implemented in JavaScript (using the D3, JQuery, and bootstrap libraries). An ANTLR library [2] is used in php environment to automatically parse the question.

VI. Case Studies

After developing ConVisQA we wanted to understand how the system may support users in performing question answering with conversational data. In particular we wanted to

[2]https://github.com/antlr/antlr-php-runtime



(a) Auto-completion

(b) What topic is generating more discussions?

(c) What is the most controversial topic

(d) what is the most negative comment in 'army security lapse'?

Fig. 4. A series of interactions with ConVisQA. The system helps the user to complete a query through auto-completion feature (a). The user continues to ask several questions and in response the system provides the answer in text as well as highlights the relevant comments in the Thread Overview (b–d).

understand: 1) Whether ConVisQA helps users in performing information seeking tasks form blog conversations, and 2) their interaction patterns during their tasks.

A. Participants

We conducted the study with three participants (age range 24 to 44, 2 male and 1 female), who are frequent blog readers (two of them reported to read blogs at least several times a day and one reported every day). The three most common reasons for them to read blogs are information seeking, guidance/opinion seeking, and entertainment. They are primarily interested in blogs about technology, politics, and education.

B. Procedure and Datasets

A pre-study questionnaire was administered to capture demographic information and prior experience of participants with blog reading. Then the ConVisQA interface was demonstrated to the participants. After that, they were allowed to choose any conversations of their interest from a set of six blogs from Slashdot, all of them having similar lengths. Participants were asked to explore the conversations according to their own interests and share key insights (if any) gained while exploring each conversation. During the study, we primarily focused on gathering qualitative data such as observations and semi-structured interviews. Each case study took approximately 40 - 50 minutes to complete and these sessions were conducted online using skype or zoom. The participants shared their screens to facilitate the observation.

For the purpose of case studies, we have used the same datasets used in ConVis [6]. These datasets were collected from two different blog sources: Slashdots and DailyKos.

C. Results and Analysis

We analyzed the case study sessions and results by triangulating between multiple data collection methods, including observations, notes taken by participants during the analysis session, and semi-structured interviews. We now present our key findings from the sessions.

1) Subjective measures: At the end of the study, we asked the participant to fill-up a questionnaire regarding the usability aspects of the interface. All the participants suggested that ConVisQA was useful for exploring conversations. Two out of three participants reported that they found the interface very easy to use for exploring a blog conversation. For the effectiveness of question answering ability of the interface, again two out of three participants found it to be very useful in retrieving answers given by the interface in analyzing blog conversations.

2) Interaction patterns: Two participants started by typing questions by selecting the "Question" option. As the suggestions for auto-completion are shown they selected one of the suggestions without typing further. Some of the common questions that they selected include "What is the most negative topic?", "What topics are generating more discussions?", and "Which user is most dominant in the discussion?". They suggested that the way ConVisQA presents the results by highlighting topics, comments, and authors in the Thread Overview is very effective. However, one participant suggested that it would be better if Conversational View can only display the comments posted in response the question asked to avoid the need for scrolling down through the Conversation View, For example, he selected a question "what is the most negative topic?" and the relevant comment(s) was almost at the end of the conversational view and he had to scroll down to read related comments. The third participant started by performing the keyword search, where she types a term and in response the system highlights all the relevant comments. She found it very effective and easy to view the answers to these questions.

3) Reactions to Interface features: In general, all participants agreed that the way ConVisQA is showing the answers of various questions by connecting all participating facets e.g. authors, topics and the makes it easy to get the answers. It is relatively clear which authors are participating in a discussion or topic while answering a particular question. P1 found *"the sentiment filter will be very helpful especially for huge conversations with many comments"*. He said that *"deleting topic is also a convenient feature to get rid of unwanted topic(s) and the user can enjoy only the topics of interest."*. P2 found that *"filtering by sentence length is very helpful to avoid huge conversation trails"*. P3 also reported *"Author search is very useful in locating comments posted by a particular author. The system filters the comments posted by a particular author as mostly blog readers tend to search for someones' (friends or relative) comments in a blog conversation."*.

The participants also made suggestions for improving the system. P2 reported that *"It was difficult to perceive the interface at first glance without any explanation he suggested to add a help-note to explain all components of the visualization"*. P3 suggested that *"It will be helpful to connect ConVisQA with live conversations e.g twitter or reddit"*.

VII. Conclusion and Future work

We have presented ConVisQA, an interactive question answering system for analyzing online conversations. The system supports the user to quickly locate and highlight comments of interests to her given information needs, by a novel combination of natural language processing and information visualization features. Given a specific question from the user, the interface highlights the answer in a visual overview of the conversations and help her to in rapidly navigating through the relevant comments, even if they are scattered around the conversation to fulfill her information needs.

We believe that this work provides an initial step towards building an effective natural language interface for exploring and analyzing a large amount of online conversations. There are several avenues of this research that we plan to explore in the future. First, currently we are handling limited types of questions that involve meta-data like topics, authors, and sentiment. In the future, it would be useful to handle more varieties of questions related to the content (e.g. why people support or oppose the obamacare policy?). More content analysis techniques such as argument mining [16] and question-answer similarity measures [10], [15] could help the system to answer such questions. Second, we would like to handle additional challenges while parsing the question including the ambiguities in natural language. We would like to enhance auto-completion features to help users formulate the questions as user types and resolves ambiguities. Third, while this work provides an initial idea of question answering using hand-crafted grammar, we are building large corpora question-answer pairs so that we can apply more advance deep learning models to automatically learn grammar rules in a supervised fashion. Finally, we would like to validate our system extensively among real users through longitudinal studies and field trials.

References

[1] K. Allen, G. Carenini, and R. Ng. Detecting disagreement in conversations using pseudo-monologic rhetorical structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1169–1180, 2014.

[2] G. Carenini, G. Murray, and R. Ng. Methods for mining and summarizing text conversations. *Synthesis Lectures on Data Management*, 3(3):1–130, 2011.

[3] K. Dave, M. Wattenberg, and M. Muller. Flash forums and forumreader: navigating a new kind of large-scale online discussion. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 232–241, 2004.

[4] M. El-Assady, V. Gold, C. Acevedo, C. Collins, and D. Keim. Contovi: Multi-party conversation exploration using topic-space views. In *Computer Graphics Forum*, volume 35, pages 431–440. Wiley Online Library, 2016.

[5] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 489–500, 2015.

[6] E. Hoque and G. Carenini. Convis: A visual text analytic system for exploring blog conversations. In *Computer Graphics Forum*, volume 33, pages 221–230. Wiley Online Library, 2014.

[7] E. Hoque and G. Carenini. ConVisIT: Interactive topic modeling for exploring asynchronous online conversations. In *Proc. ACM conf. on Intelligent User Interfaces*, 2015.

[8] E. Hoque and G. Carenini. Interactive topic modeling for exploring asynchronous online conversations: Design and evaluation of ConVisIT. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6(1):7:1–7:24, Feb. 2016.

[9] E. Hoque and G. Carenini. MultiConVis: A visual text analytics system for exploring a collection of online conversations. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI)*, pages 96–107, 2016.

[10] E. Hoque, S. Joty, M. Lluís, and G. Carenini. CQAVis: Visual text analytics for community question answering. In *Proceedings of the ACM conference on Intelligent User Interfaces (IUI)*, pages 161–172, 2017.

[11] E. Hoque, V. Setlur, M. Tory, and I. Dykeman. Applying pragmatics principles for interaction with visual analytics. *IEEE transactions on visualization and computer graphics*, 24(1):309–318, 2017.

[12] M. Huang, X. Zhu, and J. Gao. Challenges in building intelligent open-domain dialog systems. *arXiv preprint arXiv:1905.05709*, 2019.

[13] Q. Jones, G. Ravid, and S. Rafaeli. Information overload and the message dynamics of online interaction spaces: A theoretical model and empirical exploration. *Information systems research*, 15(2):194–210, 2004.

[14] D. H. Kim, E. Hoque, and M. Agrawala. Answering questions about charts and generating visual explanations. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, 2020.

[15] M. T. R. Laskar, X. Huang, and E. Hoque. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5505–5514, 2020.

[16] J. Lawrence and C. Reed. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818, 2020.

[17] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[19] T. Parr. *The definitive ANTLR 4 reference*. Pragmatic Bookshelf, 2013.

[20] V. Pascual-Cid and A. Kaltenbrunner. Exploring asynchronous online discussions through hierarchical visualisation. In *2009 13th International Conference Information Visualisation*, pages 191–196. IEEE, 2009.

[21] P. Pasupat and P. Liang. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*, 2015.

[22] M. K. Rohil, R. K. Rohil, D. Rohil, and A. Runthala. Natural language interfaces to domain specific knowledge bases: an illustration for querying elements of the periodic table. In *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 517–523. IEEE, 2018.

[23] W. Sack. Conversation map: An interface for very large-scale conversations. *Journal of Management Information Systems*, 17(3):73–92, 2000.

[24] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST 2016, pages 365–377, New York, NY, USA, 2016. ACM.

[25] V. Setlur, E. Hoque, D. H. Kim, and C. A. X. Sneak pique: Exploring autocompletion as a data discovery scaffold for supporting visual analysis. In *Proceedings of ACM User Interface Software and Technology Symposium (UIST) (to appear)*, pages 1–13. ACM, 2020.

[26] A. Srinivasan and J. Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE transactions on visualization and computer graphics*, 24(1):511–521, 2017.

[27] G. D. Venolia and C. Neustaedter. Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 361–368, 2003.

[28] F. B. Viégas, S. Golder, and J. Donath. Visualizing email content: portraying relationships from conversational histories. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 979–988, 2006.

[29] X. Wang, S. Liu, J. Liu, J. Chen, J. Zhu, and B. Guo. Topicpanorama: A full picture of relevant topics. *IEEE transactions on visualization and computer graphics*, 22(12):2508–2521, 2016.

[30] M. Wattenberg and D. Millen. Conversation thumbnails for large-scale discussions. In *CHI'03 extended abstracts on Human factors in computing systems*, pages 742–743, 2003.

[31] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162, 2010.