Eurographics Conference on Visualization (EuroVis) 2017
J. Heer, T. Ropinski, and J. van Wijk
(Guest Editors)

Volume 36 (2017), Number 3

Integrating Visual Analytics Support for Grounded Theory Practice in Qualitative Text Analysis

Senthil Chandrasegaran[1], Sriram Karthik Badam[2], Lorraine Kisselburgh[3], Karthik Ramani[4,5], Niklas Elmqvist[1,2]

[1]College of Information Studies, [2]Department of Computer Science, University of Maryland, College Park, MD, USA
[3]Center for Entrepreneurship, [4]School of Mechanical Engineering, [5]School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN, USA

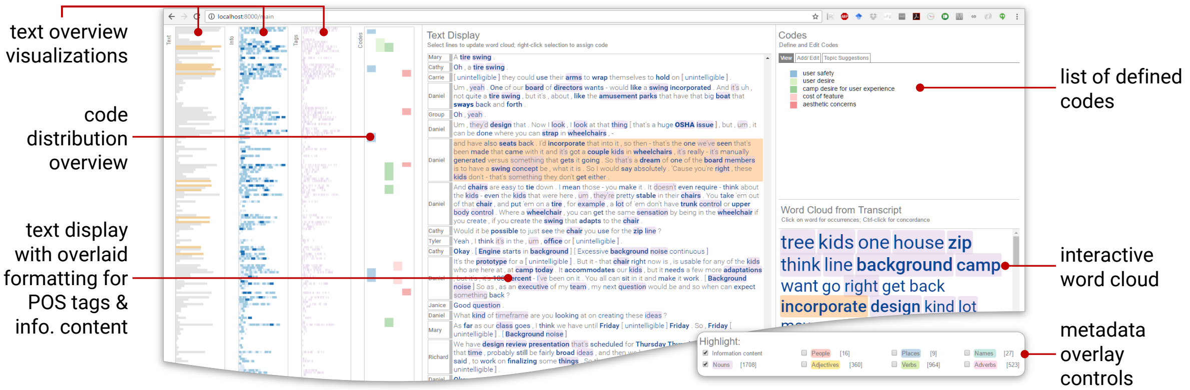

Figure 1: *Interface featuring visual analytics support for open coding in grounded theory. Text data is processed at the server end to obtain linguistic metadata such as parts of speech, named entities and information content. These are displayed as a series of coordinated overview and detail visualizations to help the analyst identify concepts and relationships between them, and iteratively code the data.*

Abstract

We present an argument for using visual analytics to aid Grounded Theory methodologies in qualitative data analysis. Grounded theory methods involve the inductive analysis of data to generate novel insights and theoretical constructs. Making sense of unstructured text data is uniquely suited for visual analytics. Using natural language processing techniques such as parts-of-speech tagging, retrieving information content, and topic modeling, different parts of the data can be structured and semantically associated, and interactively explored, thereby providing conceptual depth to the guided discovery process. We review grounded theory methods and identify processes that can be enhanced through visual analytic techniques. Next, we develop an interface for qualitative text analysis, and evaluate our design with qualitative research practitioners who analyze texts with and without visual analytics support. The results of our study suggest how visual analytics can be incorporated into qualitative data analysis tools, and the analytic and interpretive benefits that can result.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Computer Graphics]: Information Interfaces and Presentation—User Interfaces—*Interaction styles*

1. Introduction

The Grounded Theory methodology is one of the most influential models in qualitative data analysis [LT10]. It is an inductive method used to form theoretical constructs that emerge from the data, rather than those derived from pre-existing theories and concepts. Grounded theory involves systematically collecting and analyzing data with the aim of generating theory that is based on, or *grounded in* the data [SC]. The method requires repeated readings of a corpus of text pertaining to a process, narrative, or topic, in order to code the data and glean new insights into the phenomenon under study. This process is iteratively repeated in order to create more abstract categories of concepts, to identify relationships between concepts, and then contextualize these conceptual categories to finally construct a theory that is grounded in the data.

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

The first stage of grounded theory practice, called *open coding*, is data immersive where the analyst generates descriptive codes through multiple readings of the data. This method bears a strong resemblance to the sensemaking loop proposed by Pirolli and Card [PC05], where analysts make sense of data by foraging for information, collecting evidence, and forming schema that leads to hypotheses. The stages of "shoeboxing" information, organizing evidence, forming schema, and forming hypotheses in the sensemaking loop closely parallel the processes of open coding, axial coding, selective coding, and theory formation of the grounded theory method. The science of visual analytics supports data analysis using computational techniques and interactive visualizations designed to facilitate this sensemaking loop [CT05]. In this paper, we draw parallels between the data-driven, bottom-up approach of grounded theory to the data-driven, bottom-up approach inherent to visual analytics. We review the methods of grounded theory, focusing on the open coding stage, and determine the design requirements for identifying and coding concepts and relationships during text analysis. We then explore the design space of visual analytics and text visualization, and implement a prototype to support open coding, using multiple coordinated visualizations to explore and analyze text data. We use parts-of-speech, named-entity recognition, and topic modeling algorithms to assist in identifying potential concepts, which are presented to the analyst in the form of overview and detail visualizations.

We conduct a user study with novices and experienced practitioners in qualitative analysis perform a set of tasks to create codes, assign codes, and infer relationships between the data and themes generated through topic modeling. Through observations, participant feedback, and coding activity logs, we identify differences in interaction patterns between two interfaces—with and without visual analytic support—and find that participants exhibit a distinct pattern of data exploration when using the interface with visual analytics support. Based on these findings, we suggest guidelines for providing visual analytics support to grounded theory practice.

The contributions of this paper are (a) identifying the requirements for grounded theory support by drawing parallels between the grounded theory method and the sense-making process supported by visual analytics, (b) an exploration of text analytics techniques to address the identified requirements, (c) the prototype interface designed to address the requirements, and (d) a set of guidelines informed by the user study for designing visual analytics tools to aid grounded theory.

2. Background

The social scientist Donald Campbell famously stated, "all research ultimately has a qualitative grounding" [MH94, p. 40]. We can certainly agree that qualitative evaluation is necessary when the goal is to answer open-ended research questions. Qualitative data are typically non-numerical information about a phenomenon and its attributes, and are generated through a variety of methods, including in-depth interviews, direct observation, written documents, as well as non-verbal data such as sketches, images, and videos [Dey05]. Grounded theory is a method of qualitative data analysis where a theoretical understanding is developed inductively by identifying themes that emerge from a given dataset [Cha06,SC]. In this section we briefly discuss the grounded theory method and draw parallels between its approach and that of visual analytics. We then review related research in visual analytics and text visualization, drawing inspiration to support text data exploration in grounded theory.

2.1. Grounded Theory

One of the fundamental principles of grounded theory is that data collection and analysis are interrelated and iterative [CS90]. Grounded theory is characterized by its "fitness" or faithfulness to the realities of what transpires in the data. This forms the basis of its inductive nature according to Strauss and Corbin [SC], who maintain that a theoretical construct is sound only when it is induced from the given dataset. *Concepts* lie at the root of grounded theory. A concept in this context is shorthand for *conceptual label*, a descriptive identifier for an activity or phenomenon in the raw dataset. For instance, if the dataset were a transcript of a design session, then "idea generation" can be a conceptual label that can be used to tag all instances in the transcript where the speakers come up with an idea and describe it. The process of labeling raw qualitative data with conceptual labels is called *coding*.

There are four main stages in grounded theory: *open coding*, *axial coding*, *selective coding*, and *theory formation* [Dil12]. Open coding marks the start of the coding process, and focuses on the iteratively making multiple passes over the dataset to identify and categorize of events, actions, and processes—and their properties—into conceptual labels. The next stage is axial coding, where causal and semantic relationships are determined between concepts, and conceptual labels are in turn created to describe them. The next stage is to identify a "core" concept around which all or most of the other concepts seem to, or need to be unified. This process is called *selective coding*, and is identified by asking the larger question of "how can I concisely conceptualize my findings?" Finally, in theory formation, the researcher attempts to explicate the relationships between the selected concept and the remaining concepts, or to the dataset. Every stage includes *memoing* or note-taking, either as a precursor to labeling data or as general observations about parts of the dataset. Figure 2 shows these stages with brief descriptions.

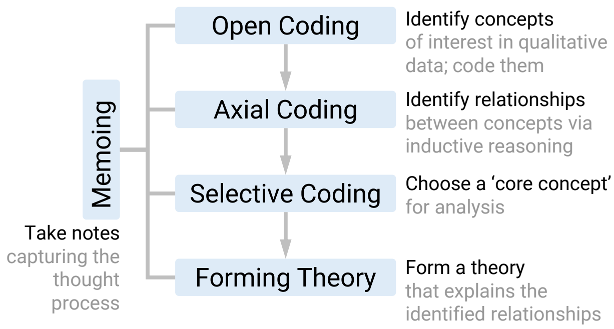

Figure 2: *The stages in grounded theory, from Dillon [Dil12].*

Computer-Aided Qualitative Data Analysis Software (CAQDAS) such as ATLAS.ti and NVivo are typically used for analyzing text data with the grounded theory method. While these tools are very effective for document collection and management, they

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.

could be enhanced by integrating NLP techniques to identify potential concepts and relationships, and by facilitating the exploration of these relationships through multiple coordinated views or abstractions. This allows deeper and more abstract explorations of meanings and relationships among text data, thus enhancing the conceptual density that is a fundamental part of grounded theory approaches. The visual representations that are available in these tools facilitate code comparison and linking, which are processes followed *after* open coding. Case studies with NVivo and Atlas.ti [BJB06, WPAM15] confirm that analysts chiefly use these software for organizing data and comparing coded text.

Corbin and Strauss [CS90] emphasize the importance of making constant comparisons between an observed incident and other incidents in the data. Identifying similarities and differences by exploring the data and making comparisons is a crucial to establish redundancy and reliability in data interpretations. Eventually, this graduates to looking for patterns and anomalies in the dataset, or "looking at the data for regularity and where that regularity is not apparent" [CS90, p. 421].

The grounded theory method bears a strong resemblance to the sensemaking loop proposed by Pirolli and Card [PC05], where analysts make sense of data by foraging for information, collecting evidence, and forming schema that leads to hypotheses. The field of visual analytics is strongly motivated by this sensemaking loop, and has developed techniques that aid the exploration and foraging of raw, unstructured data—precisely the kind of support that is currently lacking for open coding. Our goal is to enrich interpretive possibilities and conceptual depth during coding, and to explore relationships. Through visually scaffolding the systematic process of coding, categorization, constant comparison, and triangulation with other data and theory, this approach serves to facilitate the exploration of concepts and relationships, enhance analytic reliability, and ultimately deepen the interpretive depth of grounded theory practice [Pat99]. Our goal is not to *replace* existing CAQDAS tools, but to emphasize the benefits such tools stand to gain by integrating visual analytic approaches.

2.2. Visual Analytics and Text Analysis

Defined as "the science of analytical reasoning facilitated by interactive visual interfaces" [CT05, p. 4], visual analytics was developed for the intelligence analysis community to aid information processing of unstructured data. The *analyst* is kept at the center of all visual analytics systems and techniques, and is aided toward their goal of developing an effective understanding of large, complex, and unstructured datasets by combining automated analysis techniques with interactive visualizations. As Keim et al. [KAF*08, p. 155] state, "the goal of visual analytics is to make *our way of processing* data and information transparent for an analytic discourse."

When dealing with text data, the automated analysis techniques typically involve natural language processing (NLP) techniques and topic modelling, using various forms of visualizations to display the results. Visual analytics support for text analysis often focuses on analyzing connections between multiple sources of text, from intelligence reports to news articles to microblogs. Perhaps one of the more influential tools in this domain, Jigsaw [SGL08], designed for intelligence analysis, identifies entities and establishes connections between documents using occurrences of these entities. These connections are displayed to the analyst through a combination of coordinated views such as lists, graphs, calendar views, as well as views of the documents, allowing the analyst to examine, filter, and analyze intelligence data to identify threats. Tiara [WLS*10] is a system designed for analysis of text documents with a temporal component, such as emails and patient records. It uses topic modeling to summarize document collections into sets of topics, and displays the prominence of each topic over time, allowing users to select a topic and drill down to examine the underlying documents at that time. HierarchicalTopics [DYW*13] integrates a hierarchical topic modeling algorithm with a temporal view showing evolution of topics over time, allowing users to explore topics hierarchically, and edit them based on their own mental models. VariFocal Reader [KJW*14] combines visual abstractions with focus+context techniques to help navigate and analyze large documents, using topic segmentations and automatic annotation to reveal inherent document structure and entities in a document. With the exception of Varifocal reader, the techniques listed here are designed to draw connections *between* large collections of small documents, while our approach is to support the identification of concepts and relationships *within* a large document.

2.3. Exploring Text Data through Abstractions

Visual abstractions of text are often used to convey an overview of a document, to give the user a general understanding of it without having to delve into the text. Such overviews can reveal document structure, mapping lines of software code to thin lines colored to show editing statistics as done in Seesoft [ESS92], or showing repetitions of substring sequences as done with arc diagrams [Wat02]. Overviews can also be semantic, revealing the hierarchy of concepts in a document as in the case of Docuburst [CCP09]. Other overview representations can show thematic views that range from basic word frequency representations such as word clouds, or representations that reveal context, such as keyword in context (KWIC) views [MS99] More sophisticated aggregate representations include Word Tree [WV08], which aggregates concordant terms to form a tree of phrases spanning from a single word, or metadata representations such as Parallel Tag Clouds [CVW09] and ThemeDelta [GJG*15] that combine word clouds on parallel axes revealing thematic relationships between multiple documents. Serendip [AKV*14] combines thematic topic model views, coordinated with word rankings and text views to allow users to explore a text corpus at multiple levels of abstraction.

In the next section, we identify requirements for the grounded theory method, focusing on the data exploration component that is inherent to open coding. We then use existing text visualization and visual analytics techniques to address these requirements, design a prototype, and evaluate it with open coding tasks.

3. Requirements and Design

In this section, we define five key requirements for exploring and coding text data using grounded theory. We explore visual analytics approaches that meet these requirements, and select techniques to create a prototype that integrates the above approaches.

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.

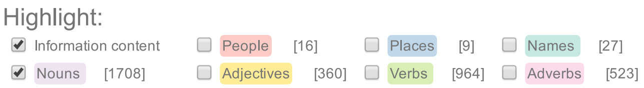

Figure 3: *Options for highlighting concepts and attributes. Multiple checkboxes control the display of named entities and parts of speech in the text view. The checkboxes also function as scented widgets [WHA07], showing the number of occurrences of potential concepts (nouns/verbs) and their attributes (adjectives/adverbs).*

3.1. Requirements

We focus our requirements on open coding—the identification and categorization of concepts—as it stands to benefit the most from a visual analytics approach. These requirements are based on literature on the grounded theory method and information visualization.

- **R1 Provide multiple data abstractions:** When analyzing unstructured text data, it helps to examine and compare multiple perspectives or views of the same data to infer structure and relationships with it. Presenting these data with multiple perspectives, e.g. a combination of word clouds, normal text, and graphical line representations can help the user identify patterns and relationships in the text.
- **R2 Support concept identification:** The identification of concepts is key to grounded theory, and is a labor-intensive process. Computational support can help identify and tag concepts, such as nouns and verbs [Bor03], and their properties, such as adjectives and adverbs. However, the decision to incorporate these suggestions is to be left to the analyst.
- **R3 Help infer relationships:** Relationships between concepts are context-specific and are informed by the knowledge, training, and experience of the researcher. However, inferring relationships based on established semantic metrics such as hypernymy or synonymy can assist interpretive processes. For instance, the WordNet taxonomy can be used to highlight relationships between ."sadness" and "happiness" through their hypernym, "emotion".
- **R4 Facilitate comparisons:** Recall Corbin's tenet of constant comparison [CS90]: once a concept is identified, the analyst should be able to look for related concepts, and make comparisons based on context, e.g. viewing multiple occurrences of a word to understand how meaning is contextualized.
- **R5 Support memoing:** Note-taking or memoing is an essential process in grounded theory research to document interpretive insights. Effective support for memoing should preserve the relationship between the memos and text, as well as support the analysis of the memos to suggest labels or infer relationships.

3.2. Exploring The Design Space

Visual analytics facilitates analytical reasoning through interactive visual interfaces designed to maximize the human capacity to perceive, understand, and reason about data [CT05]. Our exploration of the design space will thus look at text analytic techniques and interactive text visualizations to address the above requirements.

Linguistic Processing: Concepts in grounded theory could be entities, phenomena, actions, or events. In other words, concepts are nouns or verbs, and their attributes are adjectives or adverbs. The notion of looking for linguistic markers to identify themes is established procedure in qualitative analysis [RB03]. Parts-of-speech (POS) tagging [Bri92] and named entity recognition (NER) [FGM05] are existing techniques in natural language processing (NLP) that can be employed to identify concepts. Highlighting all nouns and verbs in a corpus of text would not help sift through the data: we need filtering techniques to identify unusual or unique concepts. In our prototype, we use information content, a measure based on the probability of finding a word in a corpus: rarer words are thus more "significant" [Res95]. Combining these, we have a technique for identifying potential concepts, and filtering them using various metrics, thus meeting requirement **R2**. Topic modeling techniques such as Latent Dirichlet Allocation (LDA) [BNJ03] can be used to statistically cluster words to discover abstract topics related to the clustered words (**R3**).

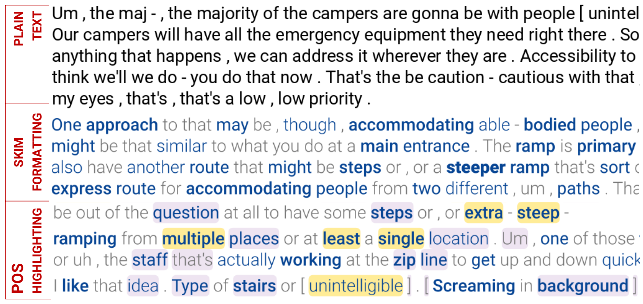

Figure 4: *Illustration of a plain text view, skim formatting [BB15] that maps information content to word weight, and parts-of-speech highlighting to show nouns (purple) and adjectives (yellow). Potentially important words stand out and can be identified either as concepts (nouns/verbs) or their attributes (adjectives/adverbs).*

Metadata Representation: A direct way of displaying the metadata obtained through linguistic processing would be to either present it side-by-side with the dataset, or overlay it on top of the dataset, or both (requirement **R1, R2**). A more nuanced method would be to provide an overview of the metadata next to the controls, forming "scented widgets" [WHA07]. Figure 3 shows this technique in use in the checkboxes to toggle the overlays in our prototype. Choosing the right visual variable for the metadata is informed by the kind of data being displayed: part-of-speech labels are nominal categories and can be represented as color highlights [CAG13, SOK*16] while frequency counts or information content measures fall under interval scales, and each needs to be represented appropriately. One solution would be to show Sparkline representations of word information content next to each word, but this compromises readability. A better solution would be "skim formatting" [BB15] where each word that falls within a range of information content measure is assigned a proportional text weight. This highlights potentially interesting concepts without sacrificing readability. Figure 4 shows the use of color highlights and skim formatting to suggest concepts and their importance. Finally, memos can be visually linked to text via markers (**R5**).

Overview Representations: As we saw in our review of text visualization techniques, overviews representations can be used to

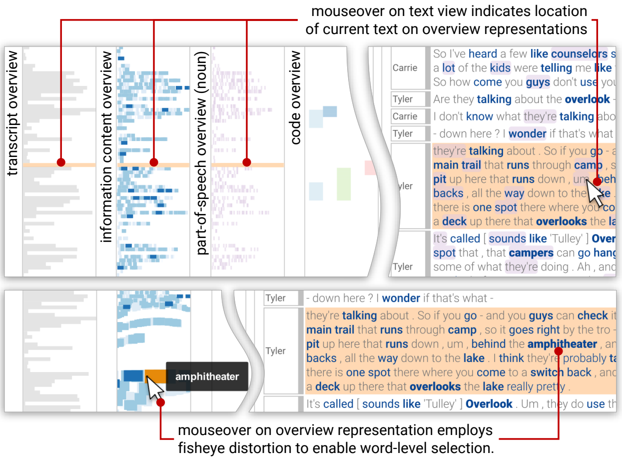

Figure 5: *Coordination between overview and detail text visualizations. An interaction with the text panel, updates the text overviews indicating the overall position of the current line in the text. When interacting with the text overviews, the corresponding line in the text panel is highlighted, and scrolls into view on mouse click. Overview element selection is aided by a fisheye distortion effect.*

represent document structure, which helps the analyst identify patterns and anomalies. They can also be used to reveal thematic content, showing dominant words or themes. Providing a structural overview of flowing text is now commonplace, seen in applications ranging from word processors to e-readers. In our case, they need to provide an on-demand overview of the text and Metadata in a way that allows the analyst to identify items of interest or simply to filter the text for contextual information (**R1, R4**). Color and position can serve as useful visual variables in this scenario: quantitative metadata such as information content can be overlaid on text overviews as a monochromatic colormap, in a manner similar to Seesoft [ESS92]. These representations can become very dense, and a fisheye distortion effect magnifies the regions of interest within densely packed overviews, and helps in pointing to individual entities [Fur86]. Figure 5 shows four structural overviews of the same text, each showing different attributes: line length, information content, POS occurrence, and code occurrence. The figure also shows the use of fisheye distortion in selecting specific lines or words. Categorical labels such as named entities and parts of speech can similarly be superimposed on the overview. Keyword in context (KWIC) views can be used to check for patterns, to answer questions such as "do all occurrences of this word have the same meaning?" Overview representations of code distribution can visually indicate code co-occurrences, and reveal patterns or relationships (**R3, R4**). Finally, thematic overviews can be provided using word clouds or topic clusters. We chose the word cloud for our thematic overview, as it is a familiar representation which is still compact, and can be powerful if coordinated with other views. The skim formatting and parts-of-speech highlighting can also be applied to the word cloud for a richer thematic overview (Fig. 6).

View Coordinations: Overview and detail representations provide multiple perspectives, but to aid the analyst in rapidly switching between these perspectives, we need to connect the two representations. View coordination is a commonly used and effective technique in information visualization where interacting with one view shows relevant information on another. These can be positional information: hovering or selecting a line in the detail text view can update the overview to indicate position in the text, or vice versa: hovering on a highlighted item in the overview can update the text view to provide detail and context. They can also reveal patterns: selecting a word in the word cloud can highlight all occurrences of that word in the text and in the overview, to reveal patterns (**R4**). For instance, such a view can help reveal whether a concept is dominant throughout a text or only at a particular portion of the text. View coordinations can also be used for filtering datasets: selecting a text or a code can update a thematic overview to reveal themes within the selection. Figure 6 shows view coordinations and filtering for the word cloud, helping identify potential concepts in the filtered overview that may be related to each other. Finally, topics generated via LDA or other means can be explored by highlighting occurrences of keywords from the topic in the transcript, to provide context and identify new relationships.

Figure 6: *The word cloud is linked to the text panel and the text overview visualizations, and reflects the frequency of word occurrence. The words in the word cloud are also overlaid with information content-based skim formatting and parts-of-speech highlighting. Selecting a block of text from the transcript, or selecting a code distribution, updates the word cloud to reflect the selection.*

We have illustrated the visualization and visual analytic components incorporated into our prototype design. In the next section, we will discuss the implementation of our prototype.

4. Implementation

We implemented the features discussed in the previous section as a web-based interface using HTML5 and JavaScript at the client end, and a Node.js server on the backend. We used the D3.js [BOH11] library for generating the interactive visualizations, and the AnnotateIt library for annotations and memoing. Figure 1 shows the prototype with a central text view with options for skim formatting and parts-of-speech/named entity tagging support. Overview visualizations on the left show various structural and thematic overviews of the transcript as discussed in the previous section. The word cloud on the right is automatically generated from the uploaded text data. Codes can be defined and viewed using the coding tabs, one of which displays a list of topics with keywords. The topic view is coordinated with the text and the overview visualizations.

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.

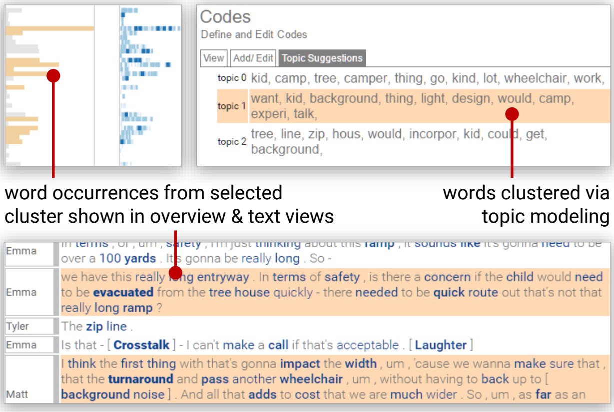

Figure 7: *The topic modeling view shows three topics identified using the Gensim library. Each topic is associated with a group of words. Selecting a topic highlights all lines in the overview and text view with one or more occurrences of the words in the group.*

The text processing at the server end uses the Stanford Named Entity Recognizer [FGM05] and the Stanford Log-linear Parts-of-Speech (POS) tagger [TKMS03] for the NER and POS tasks respectively. Information content is retrieved from the WordNet lexical database [Mil95]. Finally, we use Gensim [ŘS10] library in Python to perform topic modeling. The word groups resulting from the topic model are shown to the user as coordinated views where selecting a topic highlights all the lines of text where at least one of the words in the topic occurs (Fig. 7). Because the aim of this study was to focus on the integration of visual analytics into grounded theory, the topic model integration was minimal: there was no control given to the user to change the number of topics, or identify an optimal number of topics for their research question(s).

5. User Study

The goal of this work is to understand how visual analytics can be integrated with the grounded theory method to aid qualitative text analysis. Specifically, we sought to evaluate whether and how: (a) skim formatting based on information content would help users identify concepts of interest; (b) POS tagging in overview and text displays would help users identify concepts and their attributes; (c) interactive word clouds would help users in coding, and (d) keyword clusters generated by topic modeling would help the users uncover new insights. We chose an open coding task for our evaluation, where analysts take their initial pass in reading the text and make observations and coding assignments. Because this is the first stage of grounded theory, visual analytics could provide tangible support to analysts less familiar with the data.

Open coding is an iterative process, requiring several passes of creating and assigning codes over the dataset, and periods of reflection between coding sessions. An evaluation of a system that supports open coding would perhaps be best conducted through a longitudinal study. However, since we are evaluating a *method* of integrating visual analytics with grounded theory, our goal is to closely observe participant interaction patterns during simulated open coding tasks. We thus asked participants to perform an abbreviated open coding session conducted in a single, hour-long study for every participant. We selected a dataset that was short enough for the participants to familiarize themselves with before the study, yet long enough to require exploring during the session.

5.1. Dataset

We used the transcript of a 40-minute discussion between design students and project stakeholders as our dataset. The discussion concerns the design of a universally accessible treehouse in a summer camp for children with disabilities. The project stakeholders are organizers of the summer camp, and the students are part of a service-learning design program to benefit the community. The transcript is part of the open dataset provided by the Purdue Design Thinking Research Symposium (DTRS) [AS], with the names of the stakeholders and students changed to protect their identities.

5.2. Experimental Conditions

We used two versions of our interface for the evaluation:

- The *prototype* version of the interface described in the design section (Figure 1), and
- A *baseline* version of the interface with *no* interactive visualizations or controls for filtering/tagging, but with only coding and memoing support (Figure 8). The coding overview is the only visual representation provided in the baseline version. The reason for this is practical: participants can keep track of the assigned codes, and not inadvertently repeat their code assignments.

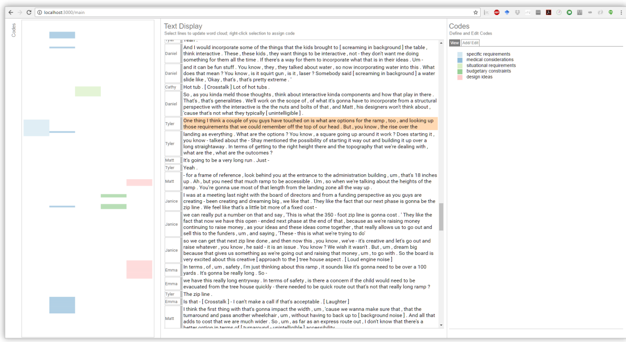

Figure 8: *The baseline version used for the user study. This interface is identical to the prototype interface, but has been stripped of all visual analytics support. The only visual representation is the interactive code overview, to help keep track of code assignment.*

We specifically avoided the use of commercially available tools for qualitative analysis for the control version. As we mention at the end of Section 2.1, our goal is to evaluate how visual analytics can aid analysts, and not to compare existing tools with our prototype. Furthermore, by using a stripped-down version of our prototype interface, we eliminate confounds due to different software.

We asked participants to analyze the same dataset using both variants of the tool (the baseline and prototype). We minimized learning effects in two ways: (1) we asked participants to read

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.

through the data (the transcript described in Section 5.1) and familiarize themselves with it *before* arriving for the study, and (2) we counterbalanced the order in which the interfaces were used: half the participants used the baseline interface first and then the prototype version, while the other half reversed this order.

We provided participants with the transcript one day before their session, and asked them to read through it once before they appeared for the study. To ensure they had read the transcript, we asked participants to write a five-sentence summary describing the discussion. We specifically requested participants to *not* annotate or mark the transcript, or take any other notes before the study. The goal was to familiarize participants with the dataset so that during the study, they could focus on the analytic task.

5.3. Participants

We recruited 6 participants (3 male, 3 female) aged between 18 and 35 years, all graduate students from information management, information studies, and human-computer interaction. All participants had experience with qualitative analysis, and three participants had experience with grounded theory methods. Four of the 6 participants had taken a course in data visualization.

5.4. Apparatus

For both the control and prototype versions, all participants used a Lenovo ThinkPad laptop with a 14” 1920×1080 display, running on Windows 10 and equipped with an external mouse. Both interfaces were presented using the Google Chrome browser.

5.5. Procedure & Tasks

Participants were introduced to the first assigned interface and its features, and then were allowed to familiarize themselves with the interface before commencing the tasks. This typically took less than ten minutes for the prototype version, and less than five for the baseline version. They were then given the following prompt:

> *The transcript presented to you concerns a discussion between designers and stakeholders discussing the design of a universally accessible tree house in a summer camp for people with disabilities. Imagine you are a researcher trying to understand the design process when design is centered around the needs of those with disabilities. Your research question or goal is to glean new insights about design considerations that are important when designing for users with special needs.*

With each version of the interface, participants were asked to perform the following tasks in sequence:

T1 *Identify and create at least three codes that you think are important to the research question/goal* (5 minutes). When using the prototype interface, participants were suggested the use the word cloud and the information content-based visualization to identify these concepts, but were given free rein to use any feature they deemed relevant.

T2 *Identify attributes or other concepts that are related to, or relevant to the concepts that you have just identified* (10 minutes). When using the prototype interface, participants were also asked to use the parts-of-speech and NER displays in addition to the features used in task T1.

T3 *Based on the topics and corresponding keyword groups shown, examine the parts of the text that are linked to each topic, and see if any of the topics in the model help uncover new insights to be explored in addressing your research question* (5 minutes, only using the prototype interface).

The same process was then repeated for the second assigned interface, but with an added instruction to reflect on the previous coding tasks and either continue with the similar codes as earlier, or create new codes that reflected insights they may have gained.

5.6. Data Collection & Analysis

We collected and analyzed four forms of data from the user study: (1) server logs of user-defined codes, (2) brief user interface surveys at the end of each task, (3) observations of participant behavior, and (4) an end-of-session usability discussion where participants drew from their experience in qualitative research to comment on features of the prototype interface. Observations made in (3) were compared across participants to identify interaction patterns. A strategy or sequence of interactions was deemed a pattern if it was observed among at least two participants. These “interaction patterns” were tabulated to identify possible connections between these patterns and the version of the interface used, or the participant expertise in qualitative analysis.

6. Results

We separated the observed interaction patterns based on the tasks of code identification and code assignments. In this section, we report on these tasks and on the usability feedback from the participants.

6.1. Code Creation: Interaction Patterns

Regardless of the interface used, participants started Task 1 by creating a number of codes. For their first pass of code creation, all participants explained that these tentative codes were based on their recollection of the transcript that they had read and summarized before starting the study. The codes created in this first pass are marked with a * in Figure 9. For their second pass, the codes created were also influenced by the previous pass of coding tasks they had just performed. They then browsed through the transcript following three main patterns of interaction.

Direct Read: This interaction involved just reading through the transcript, either in detail or by skimming through, in an attempt to discover parts of the discussion that would help them with the assigned prompt. This pattern was more dominant among the practitioners than the novices, which makes sense: this is typically how open coding is performed using most existing tools. This pattern was observed more when using the baseline interface.

Search & Skim: Observed mainly when participants used the baseline version of the interface, this interaction involved using the browser search function to look for occurrences of keywords that they recalled from the transcript, such as “safety” and skimmed the transcript around occurrences of such words to identify any specific

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.

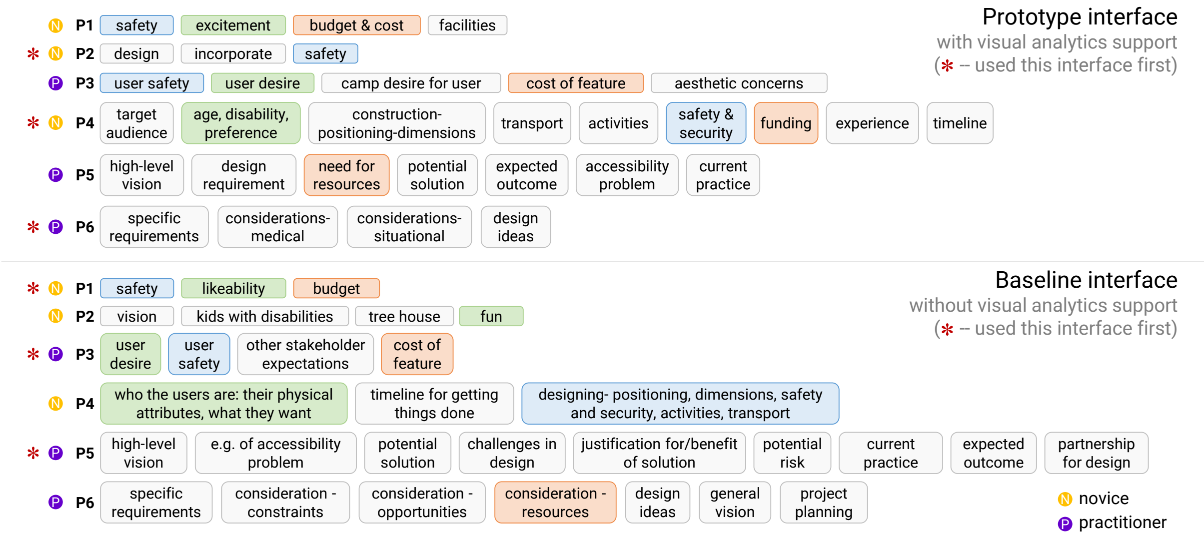

Figure 9: *Codes generated by the participants (P1–P6) using the prototype interface and the baseline interface. Each row represents all codes generated by the participant using the indicated interface. Each participant's experience is indicated on the left with an 'N' (novice) or a 'P' (practitioner). Colors are used here to indicate three dominant themes occurring in the codes across participants and in both interfaces: user experience (green), safety (blue), and resources (orange).*

context of "safety" that may be identified as a code. This technique, indicative of deductive, rather than inductive coding, was used by all but one participant (P6; practitioner) to identify codes.

Search & Overview: Every participant followed this interaction pattern when using the prototype interface, specifically the word cloud, the keyword in context (KWIC) view, and the transcript overview. Starting by first creating codes based on their recollection of the transcript, participants searched for keywords relevant to these codes. They then used the keyword's occurrence in the word cloud to overview all its occurrences in the transcript. They then either used the KWIC view to get an overview of the contexts in which the keyword is used throughout the transcript, or directly scrolled through the transcript and read the portions of the transcript highlighted for the keyword occurrence.

We observed other instances of interaction in individual participants that cannot be deemed patterns, but are still worth noting. Participant P5 (practitioner), when using the prototype interface, looked up the word "need" in the word cloud, he used the KWIC view to confirm that the word "need" was used in the context of "requirements". Further identifying that requirements would generally be verbs, he used the POS highlighting for verbs in the word cloud to identify other keywords that could relate to requirements. Describing his approach to the task, he reported, *"I used 'need' and 'incorporate' to figure out user requirement-related sentences – this helped refine design requirement-related themes."*

Table 1 shows interaction patterns observed during code creation for each interface. The table shows the number of novices or practitioners who used each interaction pattern at least once. For instance, from the table below, we can see that all three practitioners used "direct read" when using the baseline interface, while one of them used "search & skim" in addition.

Table 1: *Interaction patterns for code creation*

Interaction Pattern	Condition	Novices	Practitioners
Direct Read	Baseline	0	3
	Prototype	0	0
Search & Skim	Baseline	3	1
	Prototype	1	0
Search & Overview	Baseline	NA	NA
	Prototype	3	3

Finally, the codes created by each participant are shown in Figure 9, with dominant themes of safety, user experience, and budget highlighted in blue, green, and orange respectively. These themes were identified regardless of the order in which participants used the interfaces, indicative of a mitigation of any learning effects.

6.2. Coding: Interaction Patterns

Interaction patterns for this task primarily echoed the earlier tasks, with more distinct differences appearing between the novice users and practitioners. We observed three main patterns of interaction:

Read & Code: Congruent to the *direct read* pattern in code creation, this followed the traditional technique of open coding, where participants carefully read through each line of the transcript, assigning it to one or more of the defined codes. All three practitioners, trained in coding, exhibited this interaction pattern when using the baseline interface.

Search, Skim & Code: This pattern, similar to the *search & skim* pattern, was marked by participants using a "breadth-first" approach to coding. They searched for keywords they could recall from their reading of the transcript that were related to the defined codes, reading sections of the transcript around these keywords, and coding the sections relevant to the codes. All three novice participants exhibited this pattern when using the baseline interface.

Search, Explore & Code: Following the *search & overview* pattern from code creation, participants searched for keywords related to the defined codes, and used the word cloud and transcript overview and/or KWIC views to determine the context in which these keywords were used. They then coded these sections of the transcript based on the identified contexts. All of the novice participants used this technique almost exclusively, and one of the three experienced practitioners used it sparingly when working with the prototype interface.

Read, Code & Explore: When using the prototype interface, experienced practitioners exhibited a variant of the *read & code* pattern they followed for the baseline version. However, if they noticed words in their selection that caught their attention, they would use the word cloud to look for other occurrences of that word, to see if the same code could be assigned to those sections. While all three practitioners used this pattern of interaction, the mechanics varied between them. Participant 5 used the KWIC view to check the keyword for context, using this view to decide whether to explore further. P6 used the dynamic update feature of the word cloud: when selecting a section of the transcript to assign a code, the word cloud updates to reflect only the selected text. The words that show up in the word cloud are presumably related to the code being assigned. Using this updated view, she found it easier to explore different words related to the current code, and coded those sections of the transcript accordingly. Similarly, selecting a code in the coded timeline updates the word cloud to reflect only the lines of text to which that code is assigned. However, this feature was not used by the participants. This interaction pattern lies at the crux of the visual analytics approach: using the visualizations to make observations, confirm patterns, and delving into the data to identify anomalies. Table 2 shows the occurrences of interaction patterns during code assignment among the two versions of the interface, and among the novices and practitioners.

Table 2: *Interaction patterns for code assignment*

Interaction Pattern	Condition	Novices	Practitioners
Read & Code	Baseline	0	3
	Prototype	0	2
Search, Skim & Code	Baseline	3	0
	Prototype	0	0
Search, Explore & Code	Baseline	NA	NA
	Prototype	3	0
Read, Code & Explore	Baseline	NA	NA
	Prototype	0	2

6.3. Usability Feedback

Participants reported that code creation and code assignment tasks were both easier in the prototype interface. On a 5-point Likert scale, ratings of the prototype interface averaged 4 ($s.d = 0$) compared to the baseline's 3.6 ($s.d = 0.4$). This difference was more pronounced in the coding task, where the prototype interface ratings averaged 3.83 ($s.d = 0.85$) compared to the baseline version's 2.83 ($s.d = 0.9$). Overall, participants found the coordinated views of the word cloud, the transcript overview, and the transcript itself very helpful. While the KWIC view was used by 2 of the 6 participants, both found it useful.

Recall that Task 3 was intended to test the relevance of the keyword clusters generated by the topic model, based on the links highlighted between the keywords in the topics and their occurrences in the transcript. Most of the participants did not find this useful, stating that the occurrences were so high as to render any sense-making a difficult assignment. This was not surprising: our implementation of topic modeling was minimal, and participants had no control over the document resolution or the number of topics identified. Topic interpretability by humans has its limits: the more topics in a complex document, the more difficult it is for humans to successfully interpret it [CBGG*09]. Interactive tools such as LDAvis [SS14] are rapidly gaining popularity to help humans interpret topic modeling results. We plan to integrate similar interactive tools to help refine any topic modeling results, in order to make the underlying data transparent, and reduce bias. This was reflected in the suggestions by two of the practitioners who recommended providing control over pruning keywords and iterating over the topic modeling computation, towards a human-in-the-loop approach to improve the relevance of the feature.

The parts-of-speech highlighting was used by one participant, while the information content overview not used at all: experienced participants suggested that such views would be more useful for directed explorations, when they were more familiar with the dataset. They also suggested adding boolean operations to identify word co-occurrences and reveal semantic relationships.

7. Discussion

Our results indicate that the integration of visual analytics with the grounded theory method holds promise for qualitative research. We saw that novices use a breadth-first approach, i.e. *search-skim* for code creation and *search-skim-code* for code assignment. They search for concepts that they feel are relevant to the research goal, rather than use the "close reading" approach followed by experienced practitioners. We posit that the exploratory interactions used in our visual analytics approach eases the transition to better coding behavior because it allows participants to look for patterns and verify their assumptions. In contrast, experienced practitioners use a more focused, depth-first approach of reading through the material, continuously asking themselves the question "*is this relevant?*" The visual analytics integration helped them answer this question through the *read, code, and explore* pattern, where they could explore the rest of the dataset while keeping this question in mind. They used filtered overviews to help this process: selecting a text for coding updated the word cloud and exposing the relevant concepts to the coder helped them explore the question of "*is there a pattern to be found here?*"

Reinforcing good coding practices, we observed that experienced practitioners were able to deepen their analysis using the prototype interface without compromising coding rigor. While they used search & overview for code creation, they primarily used the close reading approach to assign codes. This was not true for novices, whose practice tended toward the convenient rather than the rigorous. It has to be mentioned that as with most specialized

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.

tools, the functions are not a substitute for rigorous training, and our study shows that trained practitioners indeed find the visual analytics integration useful. This reinforces the importance of training in the method before using tool support.

In terms of visualizations, our observations demonstrate that the integration of overview and contextual visualizations makes sense: these visualizations are driven entirely from the dataset being analyzed, and provides multiple data-driven perspectives to the analyst, which eases their process and also adds depth and robustness to their exploration. Finally, the interaction patterns with the prototype interface indicate that visual analytics is successful in re-centering the analyst in the grounded theory methodology.

When examining the visualizations and data representations that the participants did not find useful, we see a common attribute among them: visual overload. Overview visualizations of parts of speech are dense representations, given the high incidence of, say, nouns and verbs in an English-language dataset. Highlighting all the nouns in a transcript creates a considerable number of elements in the overview that the analyst has no direct way of filtering to identify, say, "concepts of interest". While the information content overview offers a parallel (and coordinated) visualization that may potentially show concepts of interest, participant feedback indicated that this was not very helpful. This is mainly due to the limits of our ability in keeping track of too many visual parameters: research has shown that filtered views are easier to visually parse than combined views [HEH09].

The two overviews that the participants found useful were the transcript overview, and the word cloud. The word cloud representation is directly actionable because each element is a word that holds inherent meaning to the analyst, as opposed to a graphical abstraction common in overview representations. Also, the word cloud and transcript overviews were the only representations that could be *filtered*: the word cloud could be filtered by selecting a block of text or a particular code, while the transcript overview could be filtered by selecting a word from the word cloud. The parts-of-speech/named-entity highlighting and the information content overviews did not have an additional filter that participants could use to prune the number of visual elements.

In fact, both these overview representations use metrics that could be perceived as either too generic or too extraneous to the data to be used by themselves. For instance, the corpus used to calculate information content may be too generic to be useful in analyzing text that is domain-specific. On the other hand, while grammatical constructs such as parts of speech focus more on syntactic structures, grounded theory can be said to focus on semantics and pragmatics. Participant P5, a practitioner, had this to say in explaining the difficulty he had with the information content overview visualizations: "*I think rather than the frequency or scarceness of words, it's more important to navigate some keywords that are related to research questions.*" Once again, this feedback is true for both grounded theory and visual analytics: keep all representations true to the data being analyzed. Based on these observations, we propose the following general guidelines for integrating visual analytics with grounded theory to support qualitative analysis:

Suggest connections based on contextual and semantic relationships: Coordination between multiple views of the dataset is a key element of interactive visualization. In qualitative analysis, however, the breadth-first search of the novice and the detail-oriented scrutiny of the expert can both be tempered by suggesting explorations. This can include a list of words that occur significantly within a code, words common to two selected codes, or even words that exhibit synonymy or hypernymy. The goal is to suggest semantic and conceptual relationships, giving the analyst the control to accept or dismiss them.

Support querying of causal and semantic relationships: Text data is inherently unstructured, and thus difficult to abstract into visualizations that illustrate relationships. However, providing multiple filters to the analyst can help them identify causal and semantic relationships. Boolean operations between filters would be highly beneficial, for instance "all verbs related to a particular requirement", or "all adjectives that describe a particular concept" are filters that use the syntactic structure of the text to expose concepts. Filtering for phrases that imply causal connections, such as "because of", "due to", "and therefore" etc. are part of existing methods used in qualitative coding to identify themes [RB03].

Overlay metadata based on domain-specific rather than extrinsic measures: As seen in the case of our information content views and participant feedback, it is better to represent metadata that is *inherent* to the given text data, rather than calculated from an external corpus. This guideline needs further exploration, however, to verify if a domain-specific metric, while still extraneous, could still be useful in exploring measures of "interestingness" and "connectedness". For instance, when exploring a dataset of a technical design discussion, engineering ontologies could be used to get a sense of specificity of the discussion.

Informed by visual analytics practices, our design introduces a prototype and set of guidelines to aid grounded theory using visual analytics. Our findings with interaction patterns using this prototype serves as a starting point for future research and design.

8. Conclusion

In this work, we reviewed the grounded theory method and identified requirements to aid the exploration of text data to identify and code concepts. Drawing from the field of visual analytics, we addressed these requirements through the use of computational tools and visualization techniques. We implemented these techniques in the form of a prototype and evaluated it with a series of open coding tasks using novice and experienced practitioners of qualitative analysis. Our findings suggest that the integration of visual analytics with grounded theory re-centers the analyst in the process, and thus holds promise for qualitative data analysis. We then suggest guidelines for designing visual analytics tools to support the grounded theory method. In future work, we propose further evaluation with longitudinal studies using expert participants. We also plan to evaluate domain-specific metrics that are intrinsic to the dataset to filter and explore the data, using more sophisticated filtering techniques.

9. Acknowledgments

This work is supported by the U.S. National Science Foundation under grant IIS-1422341. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.

References

[AKV*14] ALEXANDER E., KOHLMANN J., VALENZA R., WITMORE M., GLEICHER M.: Serendip: Topic model-driven visual exploration of text corpora. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology* (2014), pp. 173–182. 3

[AS] ADAMS R. S., SIDDIQUI J.: *Purdue DTRS – Design Review Conversations Database XRoads*. Tech. rep., Purdue University, West Lafayette, IN. 6

[BB15] BRATH R., BANISSI E.: Using text in visualizations for micro/macro readings. In *Proceedings of the ACM Intelligent User Interfaces Workshop on Visual Text Analytics* (2015). 4

[BJB06] BRINGER J. D., JOHNSTON L. H., BRACKENRIDGE C. H.: Using computer-assisted qualitative data analysis software to develop a grounded theory project. *Field methods 18*, 3 (2006), 245–266. 3

[BNJ03] BLEI D. M., NG A. Y., JORDAN M. I.: Latent dirichlet allocation. *Journal of machine Learning research 3*, Jan (2003), 993–1022. 4

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2301–2309. 5

[Bor03] BORGATTI S.: Introduction to grounded theory. www.analytictech.com/mb870/introtoGT.htm, 2003. updated Dec 2003; accessed June 22 2016. 4

[Bri92] BRILL E.: A simple rule-based part of speech tagger. In *Proceedings of the Workshop on Speech and Natural Language* (1992), pp. 112–116. 4

[CAG13] CORRELL M. A., ALEXANDER E. C., GLEICHER M.: Quantity estimation in visualizations of tagged text. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (2013), pp. 2697–2706. 4

[CBGG*09] CHANG J., BOYD-GRABER J. L., GERRISH S., WANG C., BLEI D. M.: Reading tea leaves: How humans interpret topic models. In *Proceedings of the International Conference on Neural Information Processing Systems* (2009), vol. 31, pp. 1–9. 9

[CCP09] COLLINS C., CARPENDALE S., PENN G.: DocuBurst: Visualizing document content using language structure. *Computer Graphics Forum 28*, 3 (2009), 1039–1046. 3

[Cha06] CHARMAZ K.: *Constructing Grounded Theory: A Practical Guide through Qualitative Analysis*. Sage, 2006. 2

[CS90] CORBIN J. M., STRAUSS A.: Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology 13*, 1 (1990), 3–21. 2, 3, 4

[CT05] COOK K. A., THOMAS J. J.: *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Press, 2005. 2, 3, 4

[CVW09] COLLINS C., VIEGAS F. B., WATTENBERG M.: Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 91–98. 3

[Dey05] DEY I.: *Qualitative Data Analysis: A User-Friendly Guide for Social Scientists*, 2 ed. Routledge, 2005. 2

[Dil12] DILLON D. R.: *Grounded Theory and Qualitative Research*. Blackwell Publishing Ltd, 2012. 2

[DYW*13] DOU W., YU L., WANG X., MA Z., RIBARSKY W.: Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2002–2011. 3

[ESS92] EICK S. G., STEFFEN J. L., SUMNER E. E.: Seesoft—a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering 18*, 11 (1992), 957–968. 3, 5

[FGM05] FINKEL J. R., GRENAGER T., MANNING C.: Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics* (2005), pp. 363–370. 4, 6

[Fur86] FURNAS G. W.: *Generalized fisheye views*, vol. 17. ACM, 1986. 5

[GJG*15] GAD S., JAVED W., GHANI S., ELMQVIST N., EWING T., HAMPTON K. N., RAMAKRISHNAN N.: Themedelta: dynamic segmentations over temporal topic models. *IEEE Transactions on Visualization and Computer Graphics 21*, 5 (2015), 672–685. 3

[HEH09] HUANG W., EADES P., HONG S.-H.: Measuring effectiveness of graph visualizations: A cognitive load perspective. *Information Visualization 8*, 3 (2009), 139–152. 10

[KAF*08] KEIM D., ANDRIENKO G., FEKETE J.-D., GÖRG C., KOHLHAMMER J., MELANÇON G.: Visual analytics: Definition, process, and challenges. In *Information visualization*. Springer, 2008, pp. 154–175. 3

[KJW*14] KOCH S., JOHN M., WÖRNER M., MÜLLER A., ERTL T.: VarifocalreaderâĂŤin-depth visual analysis of large text documents. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1723–1732. 3

[LT10] LINDLOF T. R., TAYLOR B. C.: *Qualitative communication research methods*. Sage, 2010. 1

[MH94] MILES M. B., HUBERMAN A. M.: *Qualitative Data Analysis: An Expanded Sourcebook*. Sage Publishers, 1994. 2

[Mil95] MILLER G. A.: Wordnet: A lexical database for english. *Communications of the ACM 38*, 11 (1995), 39–41. 6

[MS99] MANNING C., SCHÜTZE H.: *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. 3

[Pat99] PATTON M. Q.: Enhancing the quality and credibility of qualitative analysis. *Health services research 34*, 5 Pt 2 (1999), 1189. 3

[PC05] PIROLLI P., CARD S.: The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis* (2005), vol. 5, pp. 2–4. 2, 3

[RB03] RYAN G. W., BERNARD H. R.: Techniques to identify themes. *Field methods 15*, 1 (2003), 85–109. 4, 10

[Res95] RESNIK P.: Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the International Joint Conferences on Artificial Intelligence* (1995), pp. 448–453. 4

[ŘS10] ŘEHŮŘEK R., SOJKA P.: Software framework for topic modelling with large corpora. In *Proceedings of the LREC Workshop on New Challenges for NLP Frameworks* (2010), pp. 45–50. 6

[SC] STRAUSS A., CORBIN J.: Grounded theory methodology: an overview. In *Handbook of Qualitative Research*, Denzin N. K., Lincoln Y. S., (Eds.). 1, 2

[SGL08] STASKO J., GÖRG C., LIU Z.: Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization 7*, 2 (2008), 118–132. 3

[SOK*16] STROBELT H., OELKE D., KWON B. C., SCHRECK T., PFISTER H.: Guidelines for effective usage of text highlighting techniques. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 489–498. 4

[SS14] SIEVERT C., SHIRLEY K. E.: Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces* (2014), pp. 63–70. 9

[TKMS03] TOUTANOVA K., KLEIN D., MANNING C. D., SINGER Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* (2003), pp. 173–180. 6

[Wat02] WATTENBERG M.: Arc diagrams: Visualizing structure in strings. In *IEEE Symposium on Information Visualization* (2002), pp. 110–116. 3

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.

[WHA07] WILLETT W., HEER J., AGRAWALA M.: Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1129–1136. 4

[WLS*10] WEI F., LIU S., SONG Y., PAN S., ZHOU M. X., QIAN W., SHI L., TAN L., ZHANG Q.: Tiara: a visual exploratory text analytic system. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining* (2010), pp. 153–162. 3

[WPAM15] WOODS M., PAULUS T., ATKINS D. P., MACKLIN R.: Advancing qualitative research using qualitative data analysis software (qdas)? reviewing potential versus practice in published studies using atlas. ti and nvivo, 1994–2013. *Social Science Computer Review 34*, 5 (2015), 597–619. 3

[WV08] WATTENBERG M., VIÉGAS F. B.: The Word Tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1221–1228. 3

© 2017 The Author(s)
Computer Graphics Forum © 2017 The Eurographics Association and John Wiley & Sons Ltd.