ABSTRACT

Title of dissertation: VISUAL ANALYTICS FOR OPEN-ENDED TASKS IN TEXT MINING

Deokgun Park, Doctor of Philosophy, 2018

Dissertation directed by: Professor Niklas Elmqvist
College of Information Studies

Overview of documents using topic modeling and multidimensional scaling is helpful in understanding topic distribution. While we can spot clusters visually, it is challenging to characterize them. My research investigates an interactive method to identify clusters by assigning attributes and examining the resulting distributions.

ParallelSpaces examines the understanding of topic modeling applied to Yelp business reviews, where businesses and their reviews each constitute a separate visual space. Exploring these spaces enables the characterization of each space using the other. However, the scatterplot-based approach in ParallelSpaces does not generalize to categorical variables due to overplotting. My research proposes an improved layout algorithm for those cases in our follow-up work, Gatherplots, which eliminate overplotting in scatterplots while maintaining individual objects. Another limitation in clustering methods is the fixed number of clusters as a hyperparameter. TopicLens is a Magic Lens-type interaction technique, where the documents under the lens are clustered according to topics in real time. While ParallelSpaces help characterize the clusters, the attributes are sometimes limited. To extend the analysis by creating a custom mixture of attributes, CommentIQ

is a comment moderation tool where moderators can adjust model parameters according to the context or goals. To help users analyze documents semantically, we develop a technique for user-driven text mining by building a dictionary for topics or concepts in a follow-up study, ConceptVector, which uses word embedding to generate dictionaries interactively and uses those dictionaries to analyze the documents.

My dissertation contributes interactive methods to overview documents to integrate the user in text mining loops that currently are non-interactive. The case studies we present in this dissertation provide concrete and operational techniques for directly improving several state-of-the-art text mining algorithms. We summarize those generalizable lessons and discuss the limitations of the visual analytics approach.

VISUAL ANALYTICS FOR OPEN-ENDED TASKS IN TEXT MINING

by

Deokgun Park

Dissertation proposal submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Niklas Elmqvist, Chair/Advisor
Professor Hal Daume III
Professor Hector Corrada Bravo
Dr. Bongshin Lee
Professor Jaegul Choo

Foreword

A man's character is his fate.

Heraclitus

It all began with the naïve and lazy man's dream. I wanted interesting information to come to me even though I didn't know if such information existed, and I didn't search for it. I am lazy but addicted to information. I am also a Maximizer. According to the book *The Paradox of Choice: Why More Is Less* by Barry Schwartz, a Maximizer is the kind of person who scans all the available cereals in the supermarket and tries to select the best one. Frankly speaking, I envy the Satisficers because they will choose whatever option meets the requirements and forget about the rest.

That's how I initially became interested in recommender systems, which are a class of algorithms that recommend something to my taste, as Amazon and other web-based services that are trying hard to expand their outreach on social media platforms. Also, from my previous experiences with wearable technology, I knew that the ability to extract valuable information from data will be the key component in the so-called big data value chain. Without the ability to turn data into insights, big data is just investment and cost. Analytics will be what generates revenue and profit.

However, the journey never goes as expected and you never know where you will end up when you are setting out. Such was my Ph.D. I started with recommender systems, but the recommendations they make are not satisfactory due to limitations in the quality of these algorithms. It may be rather contentious to suggest that recommender systems algorithms are limited. Indeed, the world is changing and you never know if the next

big scientific breakthrough will improve them to be more effective. But as anecdotal evidence to support my claim, Netflix never ended up using the state-of-art algorithms from the famous 10 million dollar competition. The algorithm shows top-performance, but the performance gain is not meaningful to justify the algorithm implementation cost. That's when I started to look for alternatives. Did I already tell you that I am a Maximizer?

Therefore I concluded that we need to amplify the cognitive ability of the user to tackle the challenges of big data value creation. That is the focus of my Ph.D., presented here, with five design studies. The ideal ending will be that I am satisfied with my methodology and live happily ever after. But after six years of study, I see some fundamental limitations in the visual analytics (VA) approach as well. Those limitations are 1) VA systems are application/domain specific, 2) dependence on back-end algorithms that usually rely on the bag-of-words model, and 3) the requirement of user labor (this can be both desirable and undesirable. But I am rather lazy.).

So here I am packing light again to find more fundamental ways to achieve my dream. Recent advances in neural networking look promising, and, being inspired by those advances, I want to build upon them to start a new journey into this untapped area. I am excited, afraid, humble, and foolish on this new journey.

Dedication

To my brave companions, Hyun Suk, Hayun and Taejoon

Acknowledgments

The journey is the reward, and you don't know where you will go when you begin. Mine was a long one that taught me the pleasure of the academic world, and I would like to thank all of the people who helped me throughout.

First, I would like to thank my advisor, Professor Niklas Elmqvist. I still remember the excitement from when I participated his class and learned the exciting new area of information visualization. I was lucky to learn from him. His positive attitude and kindness supported me when I lost my way along the journey, and he introduced me to new projects to nurture my own research topic, not his own.

I would like to thank my friend and mentor, Professor Jaegul Choo for his support and guidance in the difficult topic of text analysis. His clear explanation of the text mining methods and the diligent focus for the research were as essential as a compass.

I would like to thank Professor Nicholas Diakopoulos for introducing me to the fascinating domain of computational journalism and for guiding me through the analysis. His passion for academic rigor motivated me, and he became a good role model for a researcher wishing to pioneer new domains in computational journalism.

I would like to thank my lab alumni and mentor, Cody Dunne, who helped me to secure a position of study at the IBM Watson Lab in Cambridge. I enjoyed the research and my family had a good time at Boston.

I have also been lucky to work with Steven Drucker on ATOM. I had been having a hard time publishing Gatherplots, and this work with him finally led to the fruitful submission . I learned from him how to be kind and positive researcher, and I thank him

for his mentoring at Microsoft Research. In addition to the academic opportunity, my family had a good time at Seattle and Mountain Rainier.

Bongshin Lee is my teacher and mentor and instilled in me passion for academic rigor. She taught me to be diligent in details while maintaining a long-term point of view. I consistently learn from her what high-quality research is like.

I am honored that I could learn from the colleagues of the HCIL at the University of Maryland, College Park. Ben Shneiderman and Catherline Plaisant are like giant pillars who co-created the HCI and visualization fields, and I am thankful that I could get their honest feedback for my humble work.

I thank my lab mates, Karthik and Zhenpeng, for their friendship. I was able to share my challenges and my happiness with them. Brian, Sigfried and Andrea gave me kind supports including the detailed feedback for the job talk and the proofreading of this.

Jeffrey Chen and Daniel Kim helped me fixing my bad english writing.

Last, but not least, I express my deepest love to my family who shared this adventure. It was not an easy decision to begin the Ph.D. journey with a family. But Hyun Suk Ku, my dear wife, supported my decision and supported me through the ups and downs. Hayun and Taejoon – I understand that you have experienced many challenges but I hope that you two have had some fun along the way from Seoul, to West Lafayette, Indiana and to Crofton, Maryland.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

An open-ended task can be defined as a task where it is inappropriate to have one single true answer, but instead it is appropriate to have several answers depending on the context and situation. Outputs of these tasks are not bounded to fixed sets of possible answers. Closed-ended tasks are the opposite, in which there is a set of possible answers and it is trivial to check whether it is correct or not, with a given ground-truth. One main challenge with open-ended tasks is that it is hard to judge the correctness of output automatically. For example, explaining why a comment or photo is funny may be an intractable computer science problem, at least with current technology. Humans, on the other hand, have no problem in doing these types of open-ended tasks because of their ability to perceive common sense, sympathy, semantics, and context. However, it is also challenging and costly to scale human judgment.

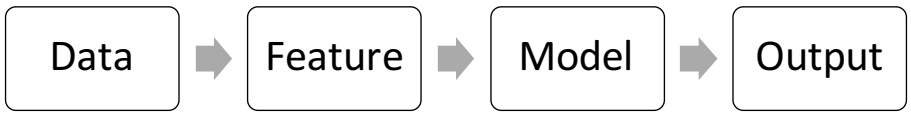


Figure 1.1: Four stages of statistical machine learning process

To tackle these problems, a visual analytics approach tries to augment and amplify a person's ability by combining information visualization and statistical machine learning.

As illustrated in Figure 1.1, the large dataset is first preprocessed to extract meaningful features. These low-level features are then modeled to give a high-level abstract about the data. Finally, these high-level abstracts about the data are visualized to allow interactive exploration, which provides valuable insight about the model's performance. Based on this insight, the user can then choose to adjust the previous steps. This forms a feedback loop where the user can change the previous stages based on the output of the system. Visual analytics is a discipline that combines the computational analysis from machines with analytical reasoning of humans using the information visualizations as a medium to bridge two as shown in Figure 1.2.

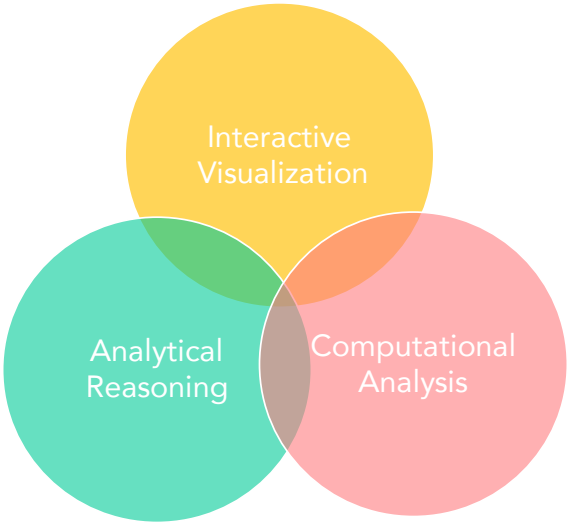


Figure 1.2: Visual Analytics approaches combine the power of the computational analysis of the machines and the analysis reasoning capabilities of humans via interactive visualization as a medium.

In the visual analytics approach, there are two mechanisms that are challenging within the domain of text processing, the first of which is sensing mechanisms. Here, the key challenge is how the person can understand the output of a statistical machine learning process. This is challenging because the text has to be transformed into numbers

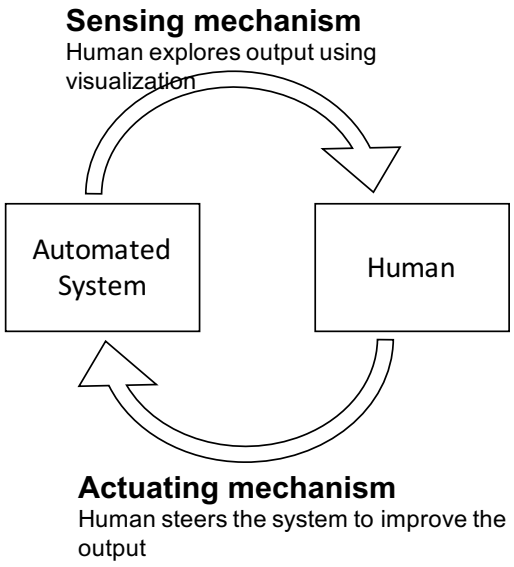


Figure 1.3: Visual Analytics approach for open-ended tasks

that are difficult to understand because of their size and dimensionality. Even when users understand the output, it is nontrivial to guide the machine learning process or update models based on this understanding — this is the second challenge, the steering mechanism. These mechanisms are illustrated in Figure 1.3. The distinction is motivated by the term *Gulfs of Evaluation and Execution*, which was introduced by Norman [1]. The mechanism can also thought as one example of the end-user interactive machine learning idea of Amershi et al [2, 3]. In the end-user interactive machine learning process, a user guides a machine by inspecting the data generated by model. In her work, she suggested design dimensions for the end-user interactive machine learning as three high-level categories. They are system feedback, end-user control and temporal category. A slight distinction is that we emphasize visualization as a means to deliver output from the machine, whereas there can be many other types of feedback in the aforementioned work. Our notion of sensing and steering mechanisms is not novel, but rather a specific instance

of these more general frameworks.

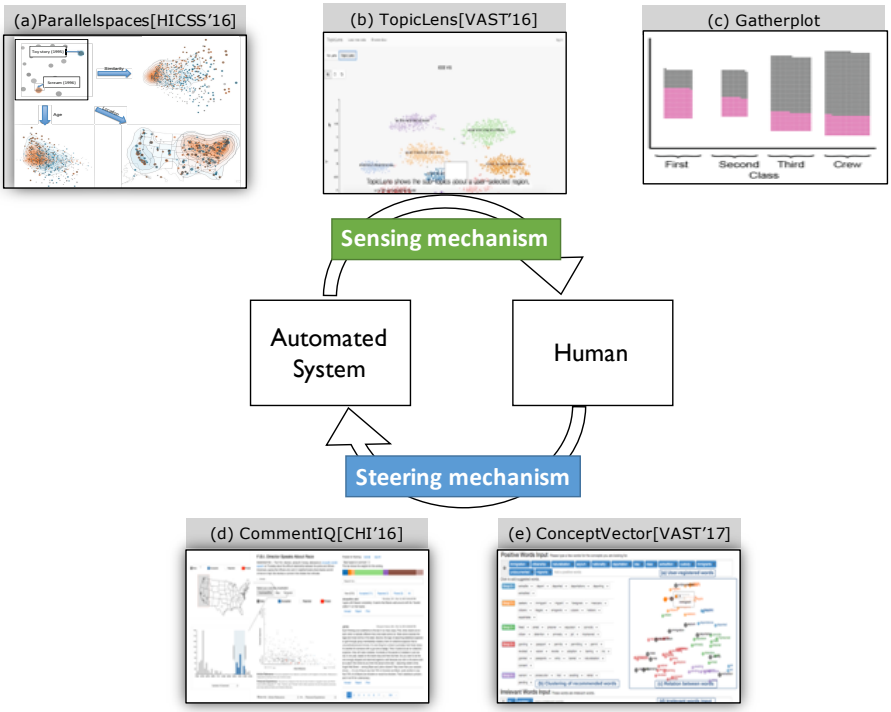


Figure 1.4: Design studies for the sensing and steering mechanism

In this dissertation, we further specialize in the domain of the text mining. We will follow the famous visual information-seeking mantra by Ben Shneiderman: *Overview first, zoom and filter, then details-on-demand* [4]. As we can see from Anscombe's quartet examples [5], overviewing data with visualization is the first step towards the analysis. However, *overview* is a broad term, and how we implement it depends on both the characteristics of the data and the goals of the tasks. Text is one type of data where overviewing is challenging, for the following reasons:

- Text is unstructured data. When it is transformed into multi-dimensional data using the bag-of-words model, it becomes very high dimensional, depending on the size of the utilized vocabulary.

- Text data requires knowledge about the world to understand documents.
- Depending on the reader or context, some text can be interpreted differently.

There are many methods to build an overview from text. Methods such as ThemeRiver show the temporal evolution of the topics [6]. Another popular method is a scatter plot view with coordinates from dimension-reduction methods, which is also often called multidimensional scaling (MDS) [7]. This view is built on a principle that similar documents are positioned closely to each other. As documents get closer in the scatter plot, it is easy to spot clusters composed of documents with similar topics. However, characterizing the clusters is difficult because the locations on the vertical or horizontal axis do not have any meaning. Still, the location on the vertical or horizontal axis is often one of the most efficient ways to encode information visually [8–10]. We propose an interactive method to characterize the clusters using the location as a visual clue for the distribution of the parameters by assigning meaning to the location. I conducted five design studies to verify the effectiveness of this technique and mitigate various issues arising from it. Figure 1.4 shows images of the projects.

ParallelSpaces [11] demonstrates how we can explore reviews of businesses to answer open-ended questions (Figure 1.5). Yelp Business Review data is analyzed by non-negative matrix factorization (NMF) [12]. The output of the NMF has an interesting duality that reveals that a document can be explained with a topic and that each topic can be explained with a word. ParallelSpaces utilizes this property by juxtaposing two canvases, each representing either a business space or a word space. In the business space similar businesses are close to each other. In the word space, similar words are clustered.

By linking those two spaces with brushing, the meaning of each space can be explored using other spaces or associated multi-dimensional properties.

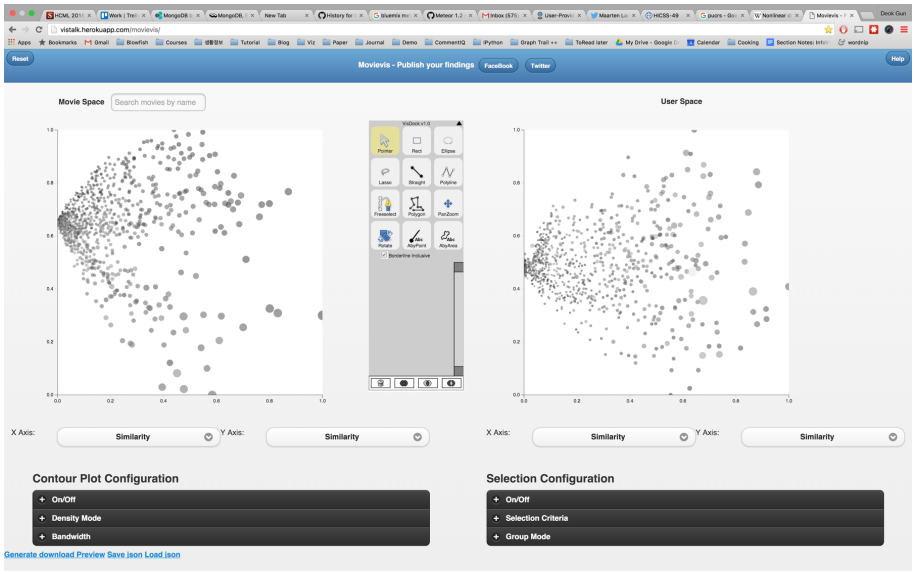


Figure 1.5: A screenshot of the ParallelSpaces system.

Assigning an attribute to the clustering view results in overplotting when the attribute is categorical. Unit visualizations, where all individual rows in the data table are visible, can be a solution. Gatherplots solve this problem by proposing a layout algorithm for categorical variables in scatterplots (Figure 1.6(c)).

Assigning an existing attribute to an axis is helpful for identifying the relationship between the attribute and the cluster. However, existing attributes can be limited. To mitigate this, we explore how users can create a custom axis from a mixture of existing attributes. The CommentIQ project [13] showcases this idea by allowing users to manipulate model parameters to create a custom axis for ranking the best comments (Figure 1.7). Online comments submitted by readers of news articles can provide valuable feedback and critique, personal views and perspectives, and opportunities for discussion.

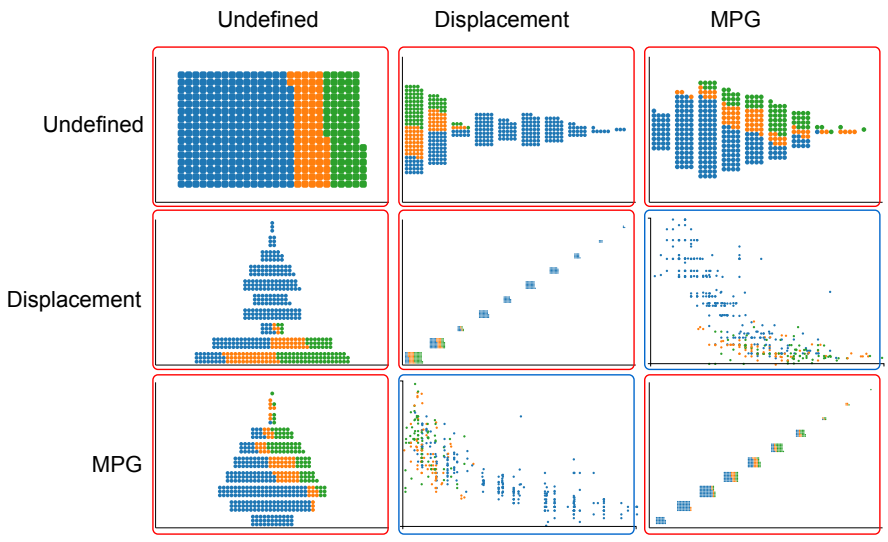


Figure 1.6: These are examples of how gatherplots display data about the cars.

The varying quality of these comments necessitates, however, that publishers remove the low quality ones. There is also a growing awareness that identifying and highlighting high quality contributions can promote the general quality of the community. In this project, we take a user-centered design approach towards developing a system that supports comment moderators, allowing them to interactively identify high quality comments using a combination of comment analytic scores as well as visualizations and flexible UI components.

In addition to creating custom attributes using weighted sums of existing features, users can improve the analytical process by creating or refining semantic features. Central to many text analysis methods is the notion of a textual *concept*: a set of semantically related keywords characterizing a specific object, phenomenon, or theme. Textual concepts have potential for characterizing document collections and can also be shared and reused once constructed. ConceptVector [14] guides the user in building, refining, and sharing

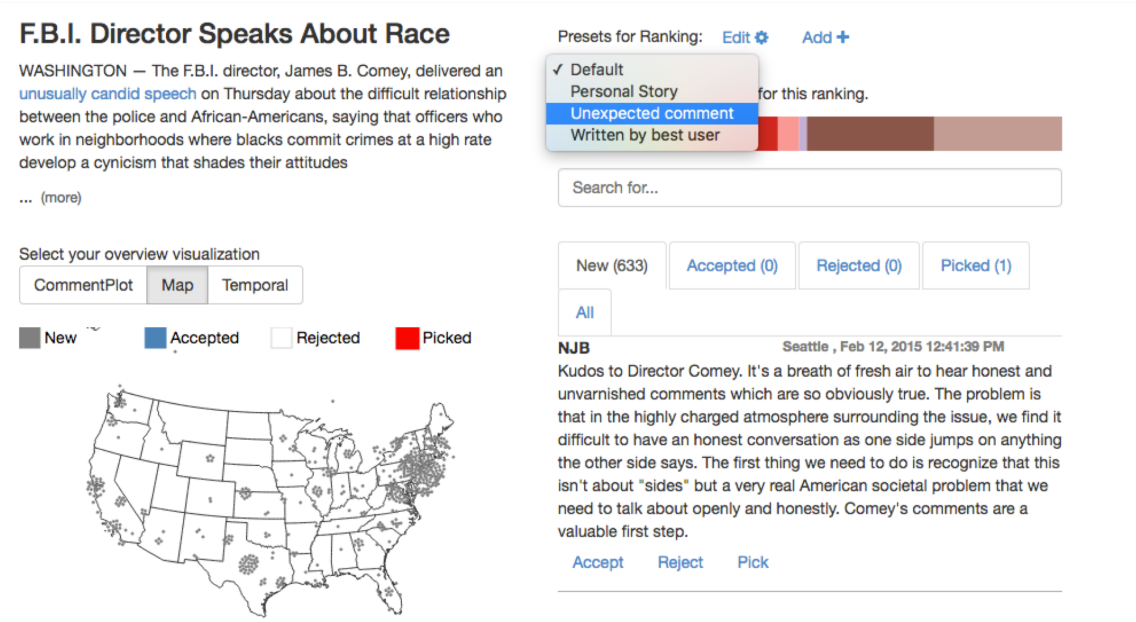


Figure 1.7: A of the CommentIQ system. The CommentIQ helps the community moderator find high-quality comments based on the custom ranked view and overview visualization.

such concepts and then in using them to classify documents (Figure 1.8). Such concepts can be used as user-driven features for text mining tasks.

TopicLens [15] is a MagicLens-type interaction technique for exploring the results of the topic modeling (Figure 1.9). The problem with the previous document galaxy view of topic modeling is that the number of topics or level of detail for the topic analysis is predetermined and cannot be changed during exploration. TopicLens solves this problem by applying a lens-type interaction in which the documents under the lens are sub-clustered by another level of topics, revealing underlying semantic structures.

Figure 1.10 summarizes the five projects in my dissertation. I contribute several interaction methods for the overview of documents in support of humans performing open-ended text mining tasks. The lessons learned from each design study provide guidelines

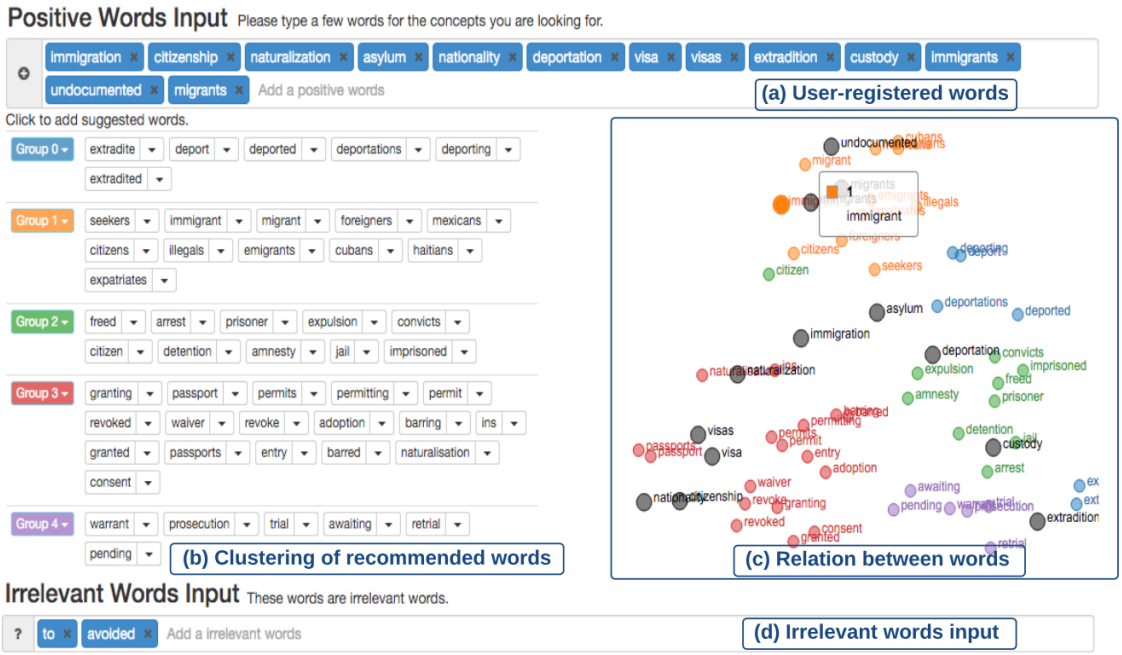


Figure 1.8: Using seed words as input, the ConceptVector system recommends related words with its meaning cluster to help users quickly build a dictionary interactively.

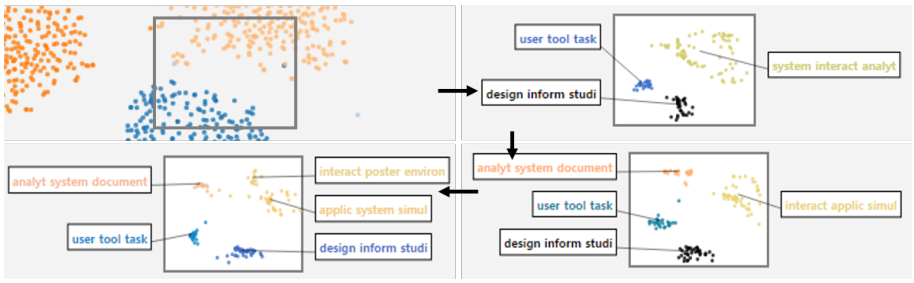


Figure 1.9: This shows an interaction sequence of TopicLens. When we apply the TopicLens, a MagicLens-type interaction for examining the underlying topic distribution, the regions under the lens are subdivided into more detailed topics.

for the application of visual analytics to text mining, empowering human analysts for high-level, open-ended tasks and opening an interesting research avenue.

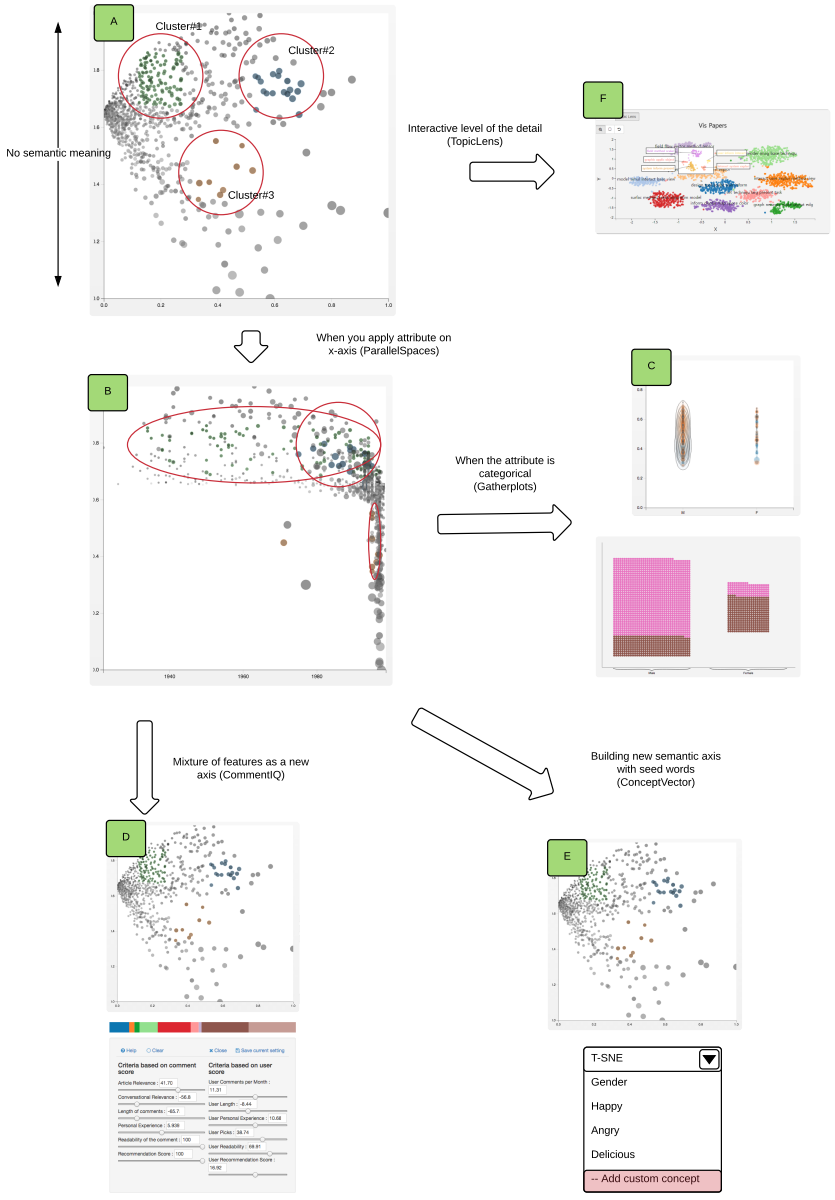


Figure 1.10: This diagram summarizes the five design studies in the dissertation. (A) represents traditional dimension reduction methods. After selecting three clusters, (B) ParallelSpaces applies the date attribute in the horizontal axis. It shows a different distribution according to the relation of the date attribute to the clusters. However, assigning a categorical axis creates overplotting, which can be mitigated by laying them out recursively in the space as in (C) Gatherplots. (D) CommentIQ experiments with building custom attributes using weighted sums of multiple features. (E) ConceptVector supports semantic analysis by building a dictionary for a specific concept and using it to create a semantic axis. (F) TopicLens is a MagicLens-type interaction technique that can zoom in to the region of interest and get detailed clustering results.

Chapter 2: Open-Ended Tasks in Text Mining

In this chapter, we present previous work related to open-ended tasks in text mining. The chapter is divided into two sections: in Section 2.1, we will explore the characteristics of open-ended tasks. In Section 2.2, two examples of open-ended tasks in text mining will be presented, which are document cluster characterizations and analyzing comments, especially to select high-quality comments to highlight.

2.1 Open-ended Tasks

If we compare humans and machines as shown in Figure 2.1, there are things that machines can do much better than humans, such as arithmetic operations. The list of such tasks is growing rapidly as tasks that were originally thought of as impossible for machines are solved using novel algorithms, such as the development of the neural network based model that beat a human champion in Go [16, 17]. Some tasks, however, remain as difficult for machines as they are for humans. Examples are NP hard problems, such as the "traveling salesman" problem [18]. The theoretical limits that keep us from (easily) solving these problems have been, to some extent, circumvented by methods that can approximate the solution with a reasonable amount of resources [19, 20].

However, there are still things that even a five-year-old (human) can do easily that

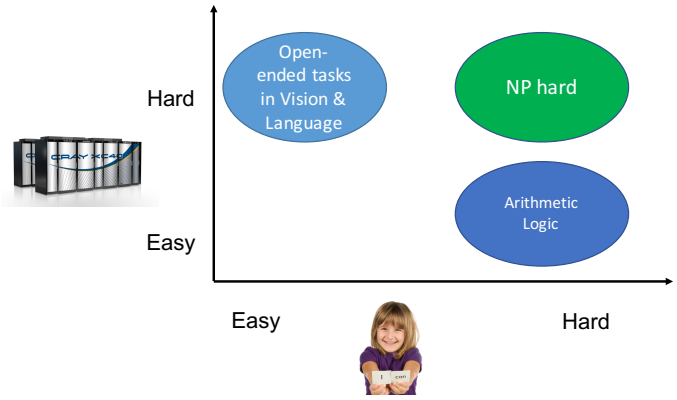


Figure 2.1: Open-ended tasks in language and vision are easy for humans while challenging for machines.

a supercomputer cannot yet perform satisfactorily. Such tasks represent the set of open-ended tasks. These are tasks that can have many different, but justifiable, answers. A concrete example can illustrate properties of open-ended tasks and why they are so challenging for machines. Nowadays, it is common to see a few thousand reviews for a product on an e-commerce site such as Amazon. A typical task for the casual shopper would be finding the answers for the following questions as shown in Figure 2.2:

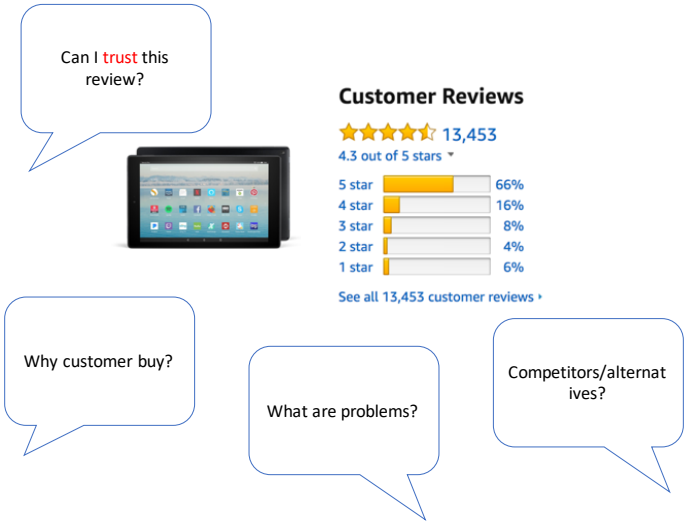


Figure 2.2: Typical questions while shopping on E-commerce site can be Open-ended tasks in language.

- Why do other people buy this product?
- Are there problems?
- What are alternatives and what could be a better choice?
- Can I trust this review?

Answering these questions is an example of an open-ended task that is challenging to automate because of the following properties:

- The tasks deal with unstructured data such as language and images. In the shopping example, you have to deal with both text and images, and the tokens are confounded by slang or misspelled words.
- There can be many different but justifiable answers depending on the context. Again, the criteria for the "best" alternatives will be different according to the situation of the buyer, including technical expertise, preference, and budgets.
- External knowledge about the world is required—not all the information to solve the tasks exists in the given problem. For example, when a comment refers an alternative product, the information about that product is not given in the review page.

Open-ended tasks are similar to AI-complete tasks in the sense that they require a human's level of intelligence. However, some AI-complete tasks are not open-ended tasks. For example, Visual Question and Answering (VQA) tasks [21], as shown in Figure 2.3, and Facebook bAbi tasks [22], as shown in Figure 2.4, require text understanding

Figure 2.3: Examples of the Visual Question Answer (VQA) dataset. While the task deals with unstructured data and knowledge about the world is required, it is not open-ended because the answers for these questions are expressible from the given set of output vocabularies, and there is only a single correct answer for each question.

Task 19: Path Finding
The kitchen is north of the hallway.
The bathroom is west of the bedroom.
The den is east of the hallway.
The office is south of the bedroom.
How do you go from den to kitchen? A: west, north
How do you go from office to bathroom? A: north, west

Task 20: Agent's Motivations
John is hungry.
John goes to the kitchen.
John grabbed the apple there.
Daniel is hungry.
Where does Daniel go? A:kitchen
Why did John go to the kitchen? A:hungry

Figure 2.4: Examples of the bAbi tasks. It is not open-ended because there is a single correct answer for each question.

and reasoning. But they are not open-ended tasks because the output space is given as a bounded set of possible answers and there are fixed sets of the correct answers for each question.

The properties of open-ended tasks challenge the development of automated methods. Previously, machine-based algorithms dealt with logical symbols, and the discrepancy between logical symbols and unstructured data in the real world has been problematic in traditional AI systems. Recently, one of the key lessons in the success of the deep learning based approach was representing textual concepts and sensory images in vector embeddings, where similarity can be expressed as a distance in the high-dimensional vector space. This has resulted in state-of-art performance for object detection, machine translation, and image caption generation, which all deal with unstructured data.

The second challenging property is that there can be many correct answers, and sometimes an unbounded number. This also breaks many previous supervised learning assumptions. For example, in supervised learning the output must be evaluated and the errors back-propagated to minimize loss. For open-ended tasks, however, it is challenging to generate enough labeled sample data to evaluate the outputs algorithmically. Recently, there are models that still generate meaningful output in challenging domains, such as image caption generation or text to image generation. But the lack of evaluation measures makes it difficult to compare different models. Currently, some methods are suggested such as BLEU, METEOR, and Rouge to evaluate the machine translation models and SPICE to evaluate image caption generation models using a scene graph.

The third property of open-ended tasks is that they require external knowledge about the world, which also makes them challenging. The traditional notion of the training dataset is not enough, because the training dataset does not contain all the information needed to solve the problem.

2.2 Examples in Text Mining

While there are many open-ended tasks, in this paper, we will focus on two tasks that are a representative of a effective visual analytics approach. The first is characterizing clusters generated by dimensionality reduction methods, and the second is the analysis of online comments in the journalism domain.

2.2.1 Characterizing Document Clusters

Document clustering is one example of unsupervised learning, in which there are no ground-truth answers, in this case the document clusters. Many methods such as k-means clustering [23] assign cluster membership according to a similarity measure. However, it is difficult to understand or evaluate the quality of the membership assignment in a table or matrix form. Visualization can help the evaluation. In this case, dimensionality reduction methods such as multidimensional scaling (MDS) [24] or principle component analysis (PCA) [25] are frequently used to create a map of the items in which similar items are gathered together.

Word occurrence in the documents is summarized as the Document Term Matrix (DTM). Many statistical methods are available to generate a clustering given the DTM. The main idea is that you can summarize the matrix with a family of matrix factorization methods, where latent variables such as topic or genre can be calculated. In many cases, the latent variables are reduced to two- or three- dimensional vectors for the purpose of visualization. In this view, we can see clusters of items, however, the main challenge is interpreting the cluster characteristics to label the group.

2.2.2 Selecting High-Quality Comments

In the social media domain, there are many examples where a crowd contributes to a forum, such as commenting on news articles. These comments can play a role in sharing opinions and discussing issues. Sometimes, comments can even be a source of information in breaking news cases. Despite their importance, current comments sections

are experiencing two problems. One challenge is how to deal with the comments that are uncivil or contains inflammatory expressions that leads to unproductive disputes. [26]. The challenge is the fact that, while there can be many valuable insights, the volume of the comments makes it difficult to find them. Selecting high-quality comments and promoting them can be a solution for both problems. It is known that promoting high-quality comments improves the quality of the discussion in general by raising expectations [27]. Selecting high-quality comments is an open-ended task, because there can be many different way of defining high-quality comments.

In the following chapters, we will explore how we can improve open-ended tasks in these two domains.

Chapter 3: ParallelSpaces: Characterizing Clusters

In this chapter, we will introduce the ParallelSpaces project, which uses interactive methods to characterize clusters by assigning an attribute to an axis. ParallelSpaces is a novel method to explore bipartite datasets in both feature and data dimensions. This dyadic data is displayed as weighted bipartite graphs using scatter plots in two separated visual spaces, where each entity is positioned according to multi-dimensional properties of the entity or similarity in preferences. Selection or navigation in one space is reflected in the other so that organic visual patterns can be formed to facilitate the characterization of underlying groupings. To aid visual pattern recognition we also overlay a contour plot based on kernel density estimation. We have implemented two instantiations of ParallelSpaces, for (a) movie preferences and (b) business reviews, as web-based visualizations. To validate the method, we performed a qualitative user study involving eleven participants using these web-based tools to explore data and collect deep insights.

3.1 Introduction

Bipartite graphs are a common way to represent content-actor relationships [28], such as movie ratings by users or e-commerce relationships between customers and products. For example, a movie fan may use a movie rating dataset to discover interesting

patterns to discuss with like-minded individuals in their social networks. Similarly, a market researcher may use the technique to find target segments of the market, such as "product A is favored by male engineers from the West Coast of ages 20 to 30." While there are many statistical analyses to aid this process, establishing initial hypotheses remains challenging. In particular, the bipartite nature of these datasets, combined with the immense amount of data, often becomes a barrier. Landesberger et al. [29] pose this as a future research challenge, where interactive feedback enables a hypothesis-insight-driven analytical process.

Even though there exist many statistical and computational tools to support this process, deriving such hypotheses in the first place is a creative, domain-specific, and culture-dependent process that requires human analysts. After the hypothesis has been formulated and tested, large-scale machine learning and statistical tools can streamline the validation process. While large data volumes, such as years of transaction records of a national retailer, can be managed by rapidly evolving technical advances in big data analytics, this is not true for the abilities of human analysts exploring the data to generate the initial hypotheses.

In this project, we argue that the main barrier against effective adoption of big data machine learning methods is in interpreting their results. These methods often yield large coefficient vectors, which are difficult to map to high-level tasks such as selecting the target group for the next advertisement campaign or finding major advantages and disadvantages of a company's products compared to its competitors. To fill in this gap, we propose a novel visualization technique for business transaction data called *ParallelSpaces*. ParallelSpaces visualizes the result of the statistical analysis in a user-friendly

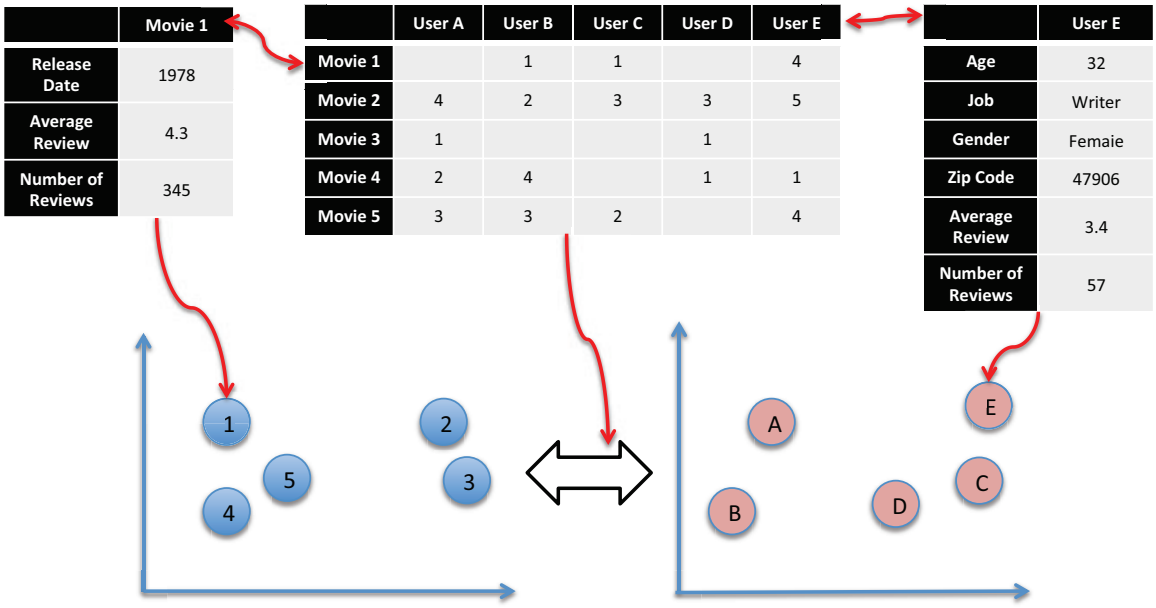


Figure 3.1: Movie entities and user entities are represented as blue circles on the left and red circles on the right, respectively. The system uses the mapping matrix, middle, to brush and link the two spaces according to the user-defined criteria. Selecting users causes selection of movies they prefer and, conversely, selecting a movie or movies leads to selection of users who give similar common ratings. Using axial rotation, the linked users and movie data can be further explored according to demographic criteria, shown in the right table, and according to movie criteria, shown in the left table.

format. The visual design of ParallelSpaces is motivated by the fact that much analytic CRM data can be classified within two categories—*customers* of a business and its *products and services*—each with qualitative and quantitative relations between and among them.

ParallelSpaces thus creates dual, side-by-side scatterplots and assigns separate 2D spaces to each such class of an entity. Each space uses a multivariate visualization of the entities in that class. Nodes are initially shown according to similarity in relation to other spaces. Selections in one space are highlighted in the other space using *brushing* [30] based on the relationship between the items, thereby forming visual patterns in the views. The user can scan these patterns to gain an overview of the transaction data. Furthermore,

scatterplot axes can be changed to enable exploration of multivariate properties of each node, such as customer demographic data or product properties. Figure 3.1 illustrates this basic concept.

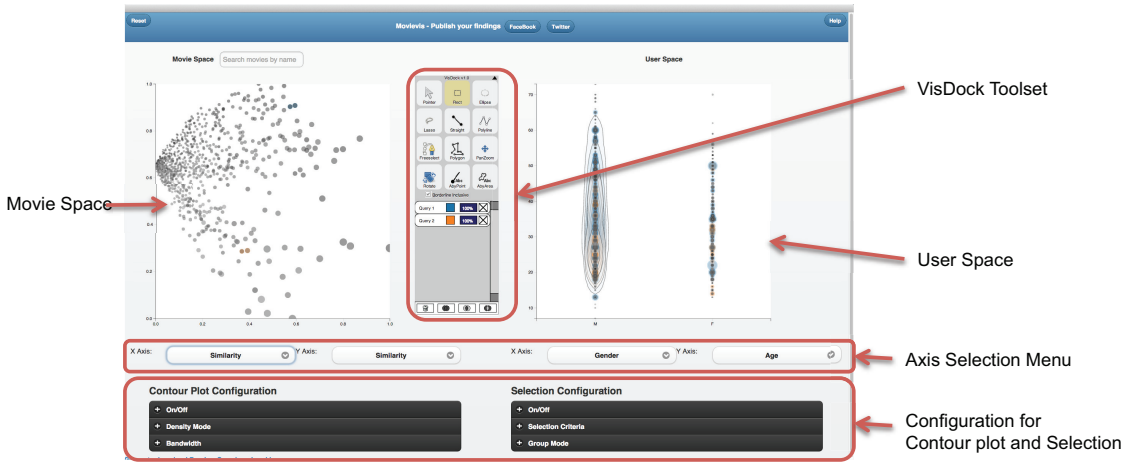


Figure 3.2: The MovieVis tool. Two groups in the movie space have been selected to compare corresponding user distributions. Two movies selected in the upper-center region—*One flew Over the Cuckoo's Nest* (1975) and *Amadeus* (1984)–and are shown in blue color. Another two movies selected in a lower-center region—*Phenomenon* (1996) and *Twister* (1996)—are shown in orange. The highlighted users are those who liked both pairs of movies (because the group mode is set to "common"). Based on the user space axes—gender for the horizontal and age for the vertical—we can see that while the movie *One Flew Over the Cuckoo's Nest* and *Amadeus* were favored by male reviewers of all ages, the *Phenomenon* and *Twister* were liked by relatively younger male audiences.

To demonstrate the effectiveness of the ParallelSpaces visualization technique, we have built web-based prototype implementations for two separate datasets: (1) a movie ratings dataset called MovieLens, and (2) Yelp business reviews. Figure 3.2 shows a screen image of the system. We used these prototypes in a qualitative user study where eleven participants were asked to explore the movie and business data in order to collect interesting findings. Our results highlight the utility of the ParallelSpaces method as well as our interaction techniques for hypothesis generation.

Our contributions are (1) the use of connected plots to show the results of the co-

clustering, (2) the design of visual elements and interactions to enable exploration, and (3) an example system with a user study on the utility of ParallelSpaces to aid in hypothesis generation.

3.2 Related Work

Our work intersects with several research areas within the general areas of visualization and visual analytics:

- **Bipartite graphs:** our data is graph-based and bimodal.
- **Multidimensional visualization:** our focus is on displaying multivariate data associated with graph vertices.
- **Machine learning:** we use mathematical and statistical modeling to extract data from multivariate datasets.

3.2.1 Bipartite Graphs

A *bipartite graph* (*bigraph*) is a graph $G = (V, E)$ whose vertices V can be partitioned into two independent sets (i.e., none of the vertices in the set are adjacent) T and U. The two vertex classes can be seen as two different types, or *modes*, of the graph, and can for example be colored using only two colors. Further, a *weighted graph* is a graph whose edges E have a weight w_i. This means that a *weighted bipartite graph* is a bipartite graph where the edges connecting the two sets have an associated weight.

Graphs in general are an active area of research, and are a core type of data for in-

formation visualization [31]. Multiple general graph visualizations exist [32], with some tools and techniques targeted specifically at bigraphs. Perhaps the closest to our work is NetLens [28], which visualizes so-called "content-actor" networks using two side-by-side and coordinated views. This content-actor network model is essentially equivalent to bipartite graphs, except their model allows for intra-relationships (within-mode) to the same set. Furthermore, the interaction propagation from one mode to the other is similar to those in our ParallelSpaces work. However, NetLens was originally designed for publication data where the contents represent papers and actors represent authors. As a result, whereas NetLens has a complex interface with many different views and visual representations, ParallelSpaces uses two side-by-side scatterplots, simplifying the visual representation and interactions between them. Because the properties of entities are visible in scatterplots, making a query is equivalent to selecting a region, which is easier for users.

Another highly relevant example of prior work is Semantic Substrates [33], where graph nodes of different modes are partitioned into separate 2D regions of the visual space, often using an attribute-based layout such as time. The visualization suppresses edges between modes except for when a node is selected. ParallelSpaces similarly employs parallel 2D spaces to partition the two different sets of vertices in the bipartite graph, and also suppresses edges. However, the main difference is that ParallelSpaces puts more emphasis on visualizing the multivariate attributes of the nodes, and is integrated with contour density plots to show how selections relate across spaces.

3.2.2 Multidimensional Visualization

Named as one of the archetypal information visualization data types [31], multidimensional data consist of multiple (more than three) dimensions and are often represented using data tables. Many systems for multidimensional visualization exist, including Tukey's PRIM-9 [34] system, Becker's and Cleveland's trellis displays [35], Ward's XmdvTool [36], and the GGobi system [37]. ScatterDice [38] demonstrated multidimensional visual exploration using scatterplots, where users can interactively assign properties to axes. ParallelSpaces follows a similar approach, but extends the idea to multimodal datasets by juxtaposing two displays.

The practice of creating multiple data views is rooted in linked graphs, which have been around for more than 25 years of statistics [35, 39], and has often been combined with *brushing*. It is also a common strategy for dealing with multidimensional datasets in interactive visualization; examples of this practice include Mondrian [40], Improvise [41], and Tableau/Polaris [42]. The most common approach to organize multiple views is called *coordinated multiple views* (CMV) [43, 44], which simply juxtaposes views in the same visual space with *brushing* [39] —dynamic highlighting of items selected in one view in all other views—as the main coordination mechanism.

3.2.3 Machine Learning

Machine learning, data mining, and information retrieval are all research areas that, similar to visualization, are tackling sensemaking for big data. Many of the methods proposed from these domains are already extensively utilized in visualization and visual

analytics. Arguably the most popular of such methods is cluster analysis [45], which uses the multivariate properties of data to find similar items so that they can be grouped together. This fits well with the concept of visual variables for visualization, where the position or location of a mark is its most salient visual feature [46]. In other words, visualizations of cluster analysis promote the understanding of latent classes in the data.

There exist many ways to extract visual coordinates from a multivariate dataset. Thus, techniques such as *dimensionality reduction* have long been an active area of research [47]. The challenge is that the process is an inherently lossy one. Self-organizing maps have been widely used as a tool for this purpose [48]. Another algorithm based on singular vector decomposition (SVD) tries to reduce the dimensionality to an underlying set of latent taste dimensions [49]. The reduced dimensionality represents "hidden themes" or "latent concepts" in the document, yielding the name Latent Semantic Indexing (LSI). A generalization of probabilistic LSI called Latent Dirichlet Allocation (LDA) [50] provides improved accuracy.

One of the applications of machine learning where bipartite data is used, is *collaborative filtering* [51] for recommender systems. A *recommender system* [52] is an information filtering system designed to predict contents for a particular user based on their own past ratings and that of other like-minded individuals (collaborative filtering), as well as based on the characteristics of the content itself (content-based filtering). The data used for the former approach—collaborative filtering—is a dyadic dataset containing implicit ratings of the form "User A bought Content 1," or explicit ratings of the form "User A gave Content 1 a rating 4 out of 5." As it turns out, this type of dyadic data can be modeled as a weighted bipartite graph, where the two sets represent users and con-

tent, and the undirected edges between the sets are ratings that individual users applied to specific content. Iwata et al. [53] used latent semantic analysis methods to create scatterplot representations of extracted data. His scatterplot arranged the movies according to their similarity in ratings patterns of users. However, it is hard to see what each cluster means. To overcome this limitation, the ParallelSpaces tries to show the distribution of users who liked each cluster, in terms of their properties like age, gender or job. It will enable hypothetical labeling of each cluster.

3.3 Data Analysis: Business Transactions

There are two kinds of datasets that characterize the majority of business intelligence data: quantitative and qualitative. We chose the two example datasets used in this paper for the purpose of representing both of these general types.

A quantitative dataset is mostly numeric, and an example is customer transaction records for a product. Such a dataset can be expressed as "customer A bought item B five times," or (A, B, 5). In this paper, these kinds of dataset are represented with the movie preference dataset called MovieLens.[1]

Even if not strictly a traditional business dataset, the movie dataset is adequate for the purposes of our paper for two reasons. First, there are no privacy issues, whereas transaction data from a real merchant can reveal the identity of customers and sensitive data related to medical or adult products. Our movie dataset has no such issues to begin with. Second, the movie preferences in our dataset are easily understandable without pre-

[1] http://www.grouplens.org/node/12

requisite knowledge and also generalizes to domain-specific business data. For instance, in the case of real transaction data, we cannot directly compare the preference based on the number of purchases if the product A and product B belongs different category.

Qualitative data is often more subjective in nature, such as customer reviews written for a product. Professional marketers try to understand the market responses by using reviews for their own product or for a competitors product to identify strengths, weaknesses, opportunities, and threats. However, sometimes the sheer number of reviews can be overwhelming. Methods such as topic modeling eases this burden by clustering documents based on their similarity. We argue that our Yelp datasets, which captures business reviews written by customers, represents such qualitative data. For example, the dataset allows for comparing good and bad Mexican or Asian restaurants based on these reviews. Again, the straightforward nature of this dataset demonstrates the ParallSpaces approach and generalizes easily to more specific qualitative business data.

3.4 Task Analysis: Dyadic Data Exploration

Pirolli and Card [54] suggested a model for sensemaking, which can be used as reference for the hypothesis generation process. However given the specific forms of dataset in the context, the task of business intelligence analyst can be further specified as follows:

- **Search and filter:** Retrieve entities according to specific multivariate properties, such as age or rating range.
- **Data distribution:** Find the characteristics of selected entities in a multivariate

dimension.

- **Finding similar entities:** Find entities that shows similar transaction patterns. Two definitions of similarity are possible. First, the properties of nodes can be similar. For example, users can be similar if they belong to the same age, gender, and geographical location group. Second, the nodes can be similar in their relations with the opposite parties. For example, two users can be similar if their buying patterns are similar.

- **Finding similar linked entities:** Find the related entities where relationship can be defined in the context of customer-product matrix. For example, in the context of the movie ratings dataset, given users, find the movies they gave more than 4 ratings. Also the relationship should be interactively adjustable. For example, the system should be able to find people who liked or disliked a certain items.

- **Estimate correlation:** Estimate the strength of relationship. For example, judging whether there is a strong correlation between the age of customers and the kind of movies they like.

As a hypothetical example to illustrate the use of these tasks, let us assume that a BI analyst is trying to find movies to recommend to a set of viewers. First he needs to select these viewers using *search and filter*. Then he may examine the property distribution of the selected moviegoers using *overview of property distribution*. Also the analyst may want to find users who show similar rating patterns with the selected target group using *identification of similar entities*. After identifying the similar users, the analyst

may identify the movies these people like in common using *identification of related entities*. Finally, having the *ability to estimate the strength of correlation* helps the analyst to iteratively explore various options using information foraging models.

3.5 ParallelSpaces: Visual Design

ParallelSpaces is an interactive visualization technique for visualizing multimodal and multivariate data in dual juxtaposed spaces that each use mutually brushed visual representations (often scatterplots). In the section below we describe the visual design of the technique, including layout, position, size, color, brightness, and density plots.

3.5.1 Space Layout

A key observation from our bipartite graphs is that at its core, the graph can be split into two independent sets. For example, in the case of the movie preference data, the users and the movies form these two independent domains. However, because the sets do not overlap, we design a basic visual representation that consists of two parallel 2D spaces, one for each set. This design is similar to the separate content and actor spaces used in NetLens [28].

The bivariate graph closely connects nodes in one space to the other. The natural way to represent this is to support brushing and highlighting between the spaces (even if we, strictly speaking, are not brushing the same entity but connected entities).

Practically speaking, this means that selecting an entity in one space corresponds to selecting the connected entities in the other parallel space. For example, if we select a

movie in the movie space, the users who liked the movie are selected (and highlighted) in the user space. Analogously, if a user is selected in the user space, the movies that received high ratings from that user can be selected in the movie space. Since we have relaxed the traditional constraint that brushing applies to the same item in different views, the underlying relationship is customizable. For example, a researcher may want to see which groups did not like a specific movie. In this case, the researcher can filter the relationship between the two spaces, making this pattern clearly visible.

The position of a visual mark is often the most salient feature in a visualization. In ParallelSpaces, any multidimensional property can be an axis. However, the relationship table is only visible when a user selects some entities. To make the relation between entities more clear, we also allowed the user to organize the 2D layout of points by their similarity. The more similar the entities are, the closer they will be placed.

Given that each user has ratings over m possible movies, each user is represented as an m-vector. Similarly, each movie is an n-vector representing users and their ratings. Finding a position for each entity in a 2D space thus becomes a projection (or dimensionality reduction) problem where m- or n-dimensional vectors are projected onto a two-dimensional space. Naturally, there are many approaches to achieving this goal: principal component analysis (PCA), multidimensional scaling (MDS), singular vector decomposition (SVD), and probabilistic Latent Semantic Analysis (pLSA) are some of the choices. In our implementation, we choose PCA as our solution, but other alternatives—even hybrid ones—are possible and within context of the overall ParallelSpaces method.

This similarity positioning feature provides starting point for the analysis, because the meaning of each cluster can mean a market segment, which shares similar preference

Entity	Feature	Visual Variable
Movie	Number of ratings	Size
User	Number of ratings	Size
Movie	Average ratings	Opacity + brightness
User	Average ratings	Opacity + brightness

Table 3.1: Selection of salience features and the matching visual variables for ParallelSpaces in the MovieVis prototype implementation.

patterns. For example, selecting a region in the movie space at similarity axis, the people who liked the movies can be selected. Further, by interactively changing axes of the user space, we can explore if there are particular patterns in age, gender, job or location dimensions.

Because the number of nodes can be high, we need to differentiate the visibility of nodes according to their importance. In our MovieVis implementation of ParallelSpaces for movie preference, we selected the setup listed in Table 3.1 to represent salience.

In ParallelSpaces, we use colors to represent set memberships when highlighting items. Also transparency is applied, so that when an item is part of two or more selections, the colors are mixed to represent its memberships. The benefit of this approach is that it does not change the size of the mark. However, the drawback is that color transparency and blending are more difficult to perceive, particularly for many selections.

3.5.2 Showing Distribution: Contour Plot

Perceiving the distribution and density of a large number of visual points in a substrate is negatively affected by both scale and overplotting (which leads to occlusion). Meanwhile, being able to assess the distribution of a group of entites corresponding to a brushed selection in another space is an important analytical task. There exist several different approaches to address this problem, such as:

- Smaller marks, yielding less overplotting;
- Transparency, to mitigate occlusion;
- Contour plots, to represent the density pattern; or
- 3D mesh gradient, to characterize the distribution.

Smaller marks may affect user interaction because the users will have difficulties in selecting them. This can be particularly problematic for touch-based tablets or mobile phones with screens. Furthermore, transparency is already assigned to data salience.

Our design choice is therefore to dynamically construct contour plots for each visualization space to show data density. More specifically, we use *kernel density estimation* (KDE) to smooth and quantify the underlying group of points. Basically, the idea is to construct visual representations of KDE clusters around the selected group of points to communicate their distribution. While a 3D mesh representation may also have been useful, we prefer to choose visual representations that fit within our 2D visual design.

KDE algorithms generally have two tunable factors: the *bandwidth* and the *kernel*. The bandwith determines the size of each kernel, which indirectly yields the degree of

smoothness of the resulting image. If it is low, it may cause noisy patterns, which are hard to identify. When it is too high, on the other hand, it can create a distribution pattern that is too smooth and carries little meaningful information. Previous work shows that the optimal bandwidth can be determined by signal characteristics [55]. In our work, the user can interactively change the bandwidth. For bivariate KDE, the kernel parameter can also have an effect on the accuracy.

In our MovieVis prototype for ParallelSpaces, we support two types of contour plots (Figure 3.3):

- **Density mode:** Show the density of selected entities in the particular space (the common approach). All kernels will have the same height, regardless of their weight (i.e., movie rating).

- **Amplitude mode:** Modulate the kernel height by the corresponding entity ratings, causing higher values in areas with high ratings and less influence from areas with lower ratings. While conveying more than just point density, this approach has the drawback of confounding density with weights.

If the analyst merely wants to see which movies a user or group of users rated, the density contour plot can provide that information. However, the amplitude contour plot will also show information on the individual ratings that the selected users gave to these movies. Comparing the two plots may yield interesting new insights.

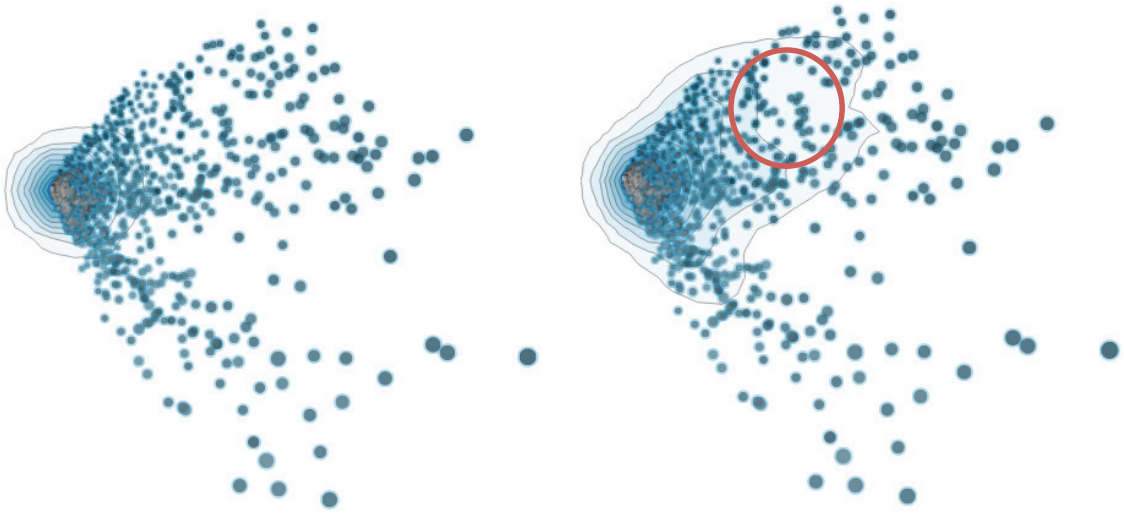


Figure 3.3: Density (left) and amplitude (right) contour plots for all the movies rated by a male educator (age 47). The selection criteria was every movie he liked. By comparing density mode KDE and amplitude mode KDE, we can spot the area where the users especially liked over the area the users have transaction records.In this example, the red circled area will contain the movies he rated more highly.

3.6 ParallelSpaces: Interaction Design

ParallelSpaces relies heavily on interaction to support visual exploration. Below we review our interaction design.

3.6.1 Selection

One of the most frequent tasks of visual analytics is comparing patterns between multiple entities. To support this process, an ordinal color is given to each selection to show which items belong to the selection. Selections can consist of one or multiple

entities defined by an enclosing border. A lasso tool allows selecting multiple entities. Hovering over an item shows a tooltip with the movie title and a link to the IMDb page, where more information is available.

To support finding particular movies and users, we provide a search toolbar with autocomplete support. When the user is looking for a specific movie, he or she can type a few words to find it. Selecting a movie from the search bar is equivalent to clicking it.

Because we regard each space independently, there are two modes of selection. When items are selected in the movie space, movies are selected and the selection propagates to the user space based on their relation. This is movie mode selection. Similarly, when users are selected in the user space, the corresponding movies will highlighted in the movie space. This is accordingly called user mode selection. Selection modes are simply switched by clicking in the opposite space. In the case of movie mode, selecting another movie will add the selected movie to the selection queue to enable the comparison of the visual pattern with previously selected movies.

3.6.2 Relationship between Spaces

As the relationship between the parallel spaces is customizable, we provided a simple range slider to adjust the relationship to investigate. For example, when the range slider is in the 4 to 5 range, selecting a movie entity will highlight all the users who gave that particular movie 4 to 5 ratings. However, when the range slider is in the 0 to 1 range, selecting a user entity will highlight all movies, which that particular user gave 0 to 1 ratings.

Our implementation also supports standard navigation techniques such as zooming and panning using mouse wheel and dragging. To reduce the effect of overplotting, we applied semantic zooming, where the points become smaller when zoomed in. We also use animated transitions to maintain object constancy in the display and allow the user to easily perceive state changes. This is particularly important for the axis rotation, where points change position.

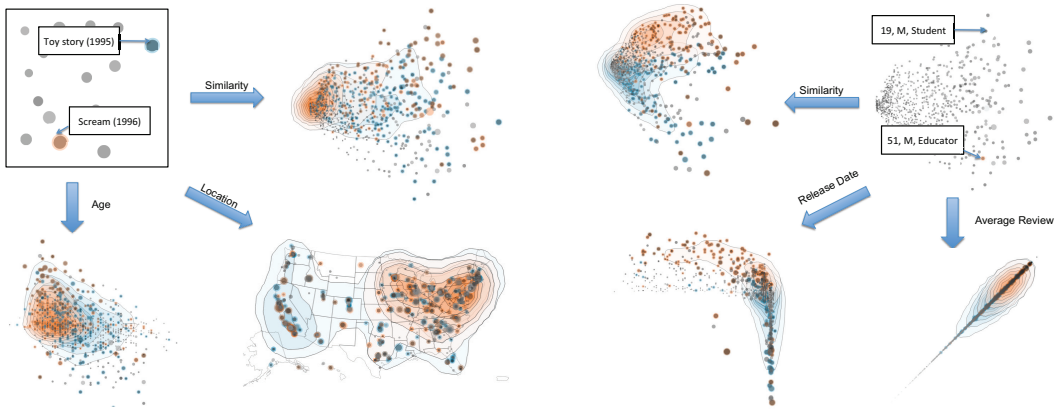


Figure 3.4: On the left, we compare two movies, *Toy Story* (1995), in blue, and *Scream* (1996), in orange, according to the age, location and similarity criteria for users. Some notable observations are while the former is liked all around the U.S. by any age groups the latter is mostly popular in the eastern part and within a younger generation. On the right, we compare two users, a 19-year-old male student, in blue, and a 51-year-old male educator, in orange according to the average, release date, and similarity criteria for movies. We observe that the older user tends to rate older films highly. In addition, his average review tends to conform to the average ratings patterns of all users while the younger user seems to deviate from it.

3.7 Implementation Notes

We have implemented two prototype instantiations of the ParallelSpaces techniques: MovieVis, for movie ratings using the MovieLens 100k dataset, and YelpVis, for business reviews from Yelp.com. In the case of YelpVis, the relationship between words and business was the number of occurrence of the words for particular business. The word space contains words like 'fantastic', 'good', or 'bad' for restaurants and the frequencies at which certain words' appear vary for different restaurants. The rationale is that users can easily discover the patterns in the reviews of restaurants using a set of such words. Both prototypes were built as web-based JavaScript and SVG applications using the D3 visualization toolkit [56]. We use the VisDock[2] library (also JavaScript) for advanced cross-cutting interaction support for selection, query management, and annotation. An interactive demonstration of the MovieVis prototype can be seen at http://vistalk.herokuapp.com/movievis/, and the YelpVis prototype is available at http://vistalk.herokuapp.com/yelpvis/.

3.8 Usage Example

We give a usage scenario to explain how the ParallelSpaces tool can help someone with forming an initial hypothesis about the dataset. Let's say a market researcher uses MovieVis to study the preference data of two movies *Scream* (1996) and *Toy Story* (1995). She selects these two movies in the movie space using the search option provided by MovieVis. This visualizes the preference data on the user space with both axes set to

[2] https://github.com/VisDockHub/NewVisDock

similarity by default. MovieVis provides a drop-down menu to set the axes in the user space to one of the seven quantities: Similarity, Age, Job, Location, Gender, Average Review, and Number of Reviews. She selects "Location" as the X-axis in the user space to display the users on a geographical map of United States. Figure 3.4 shows the visualization after applying the settings above to the user space. She observes from the contour plots in the user space that, while *Scream* is highly rated by users on the East Coast, *Toy Story* is highly rated by users all around the United States. Thus, she changes the Y-axis to "Age" while leaving the X-axis to the default setting, yielding the visualization in the bottom left of Figure 3.4. She then observes that while *Scream* is highly rated by users of age groups 15 to 30, *Toy Story* is highly rated by users of all age groups.

3.9 Qualitative User Study

The primary purpose of ParallelSpaces is to aid in generating initial hypotheses for weighted bivariate graphs. We conducted a qualitative user study to evaluate whether the system achieves this purpose.

3.9.1 Method

We recruited 11 (8 male, 3 female) paid participants to use the MovieVis and YelpVis systems for 20 minutes each. All participants were university students, and the average age was 26, ranging from 20 to 34. Prior to using the systems, participants were given 10 minutes of training in using the tools. During the exploration (two sessions of 20 minutes), they were encouraged to write comments about their findings using an anno-

tation feature embedded in the tools. After completing the exploration sessions, the users were asked to evaluate their experience in terms of usefulness, enjoyability, and ease of use. We also collected subjective free-form feedback (comments and notes) as well as basic demographic and technical information about the participants. A full user study session lasted approximately one hour (10 minutes of training, two 20-minute sessions for exploration, and 10 minutes for the post-test survey).

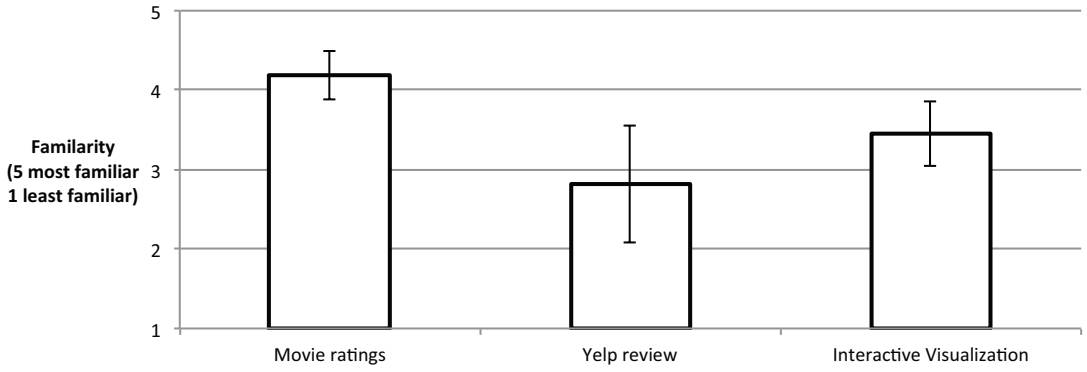


Figure 3.5: The demographic survey shows that our participants were quite familiar with movie ratings, while their knowledge of Yelp business reviews was on average lower and with higher variation. Participant expertise for interactive visualization was also diverse.

3.9.2 Results

Figure 3.5 shows the demographic survey data for our participants. In general, all 11 of our participants were able to understand the MovieVis and YelpVis tools and to independently perform data exploration using them. In total, participants wrote 71 comments for MovieVis and 52 comments for YelpVis using the embedded annotation mechanism in the tools, yielding an average of 6.5 (s.d. 4.8) and 4.7 (s.d. 3.8) comments

per participant, respectively. The overall feedback for the tools was generally positive, but participants provided many specific points of improvement and criticism.

Figure 3.6 shows the post-study survey ratings on efficiency, ease of use, and enjoyability. The ratings for YelpVis were lower than for MovieVis. One explanation might be that the participants' prior interest and knowledge of the datasets was lower for Yelp business reviews than for movies (Figure 3.5). This is supported by the fact that of the 11 participants, the five with low familiarity with Yelp reviews also gave significantly lower subjective ratings than the remaining six who were familiar with Yelp reviews. Interestingly, that same group of five gave MovieVis higher scores.

In the treatment below, we analyze our qualitative results from the study based on three basic aspects: efficiency (the perceived usefulness of the tools), enjoyability and motivation (how well the tool guided and motivated the participants), and ease of use (usability or conceptual barriers hindering the exploration). We also discuss several points of improvement that were raised by participants.

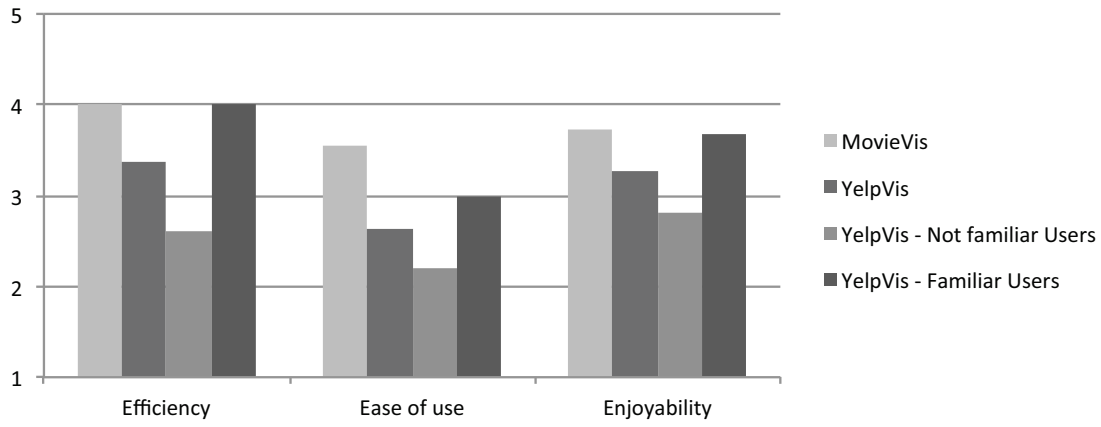


Figure 3.6: Subjective ratings for the MovieVis and YelpVis tools for the qualitative user study.

Efficiency Most participants expressed very positive feedback on the efficiency of ParallelSpaces in terms of general usefulness and utility. MovieVis, in particular, was preferred as highly useful, presumably due to the familiarity and interest bias of the datasets as discussed above. Several of the comments were expressly derived from advanced features of the system. For example, one participant stated *"So Matilda and Contact are both good movies and both liked by a lot of people from all ages, but they have a 'far' similarity because Contact has way more reviews than Matilda and [is] closer to movies [...] like Star Wars..."* The same participant used selections and graphical axes to speculate how the number of reviews affect the similarity metric in the visualization, and also suggested a fragmentation in the audience of these two movies that corresponds to their different genres.

Enjoyability and Motivation Motivational factors play an important role in collective intelligence systems, which rely on the voluntary efforts of individual users. In the feedback from participants, several people provided positive feedback, such as one participant noting that he did not notice how 20 minutes had passed already, and another requesting the URL of the tool to continue exploring after the study. However, a few participants did not seem to enjoy the experience even if this was not clear from their verbal or written feedback. We speculate that this is due to the relatively high analytic and conceptual thresholds in using ParallelSpaces effectively; one participant underscored this by stating that *"as a geek, I would like to play with this, but it is not for non-geeks."*

Ease of Use The score for the ease of use was of the lowest of the three. Several participants were concerned about the usability of the system, in particular for understanding the word business relationships in YelpVis. The stopwords for the general query was not adequate for YelpVis and resulted in many frequent words with little meaning, such as *go* and *place*.

Furthermore, the concept of similarity was not well-understood for some participants. They frequently relied only on the other concrete axes such as age, occupation, and average rating. In addition, participants rarely used the contour plot in the user study, and even those that did expressed confusion on its meaning, suggesting that this functionality could be better integrated into the tool.

3.10 TopicLens

One limitation in ParallelSpaces is that the number of clusters is a fixed hyperparameter. The number of clusters affects the level of detail: if too low, the user will not be able to see the underlying subtopics. Likewise, if the number of clusters is too high, the user will be forced to interpret clusters that do not pertain to the user's interest.

To mitigate this problem, we introduce TopicLens, a Magic-Lens-type interaction technique for the clustering. The main idea is that you can zoom into the regions you are interested, and the area under the lens will be clustered to reveal underlying subtopics.

Topic modeling is a method to investigate the underlying topics from documents. Previous methods such as LDA take significant amounts of time which makes visual analytics challenging.

Recent methods such as non-negative matrix factorization(NMF) take significantly less time. In most cases, the users have to provide the number of topics. However, the number of topics determines the granularity of the analysis. If the number of clusters is too small, many subtopics will be merged into single clusters.

In contrast, if the number is too high, it takes a significant amount of time to analyze the generated topics. Furthermore, the users may be interested only a small set of topics.

To mitigate this challenge, we developed a novel interaction technique named TopicLens, which allows an interactive analysis by conducting a clustering given the user's selection as shown in Figure 3.7.

A novel, efficient topic modeling algorithm was developed to support this interaction in real time [15] based on nonnegative matrix factorization [57]. In addition, to make

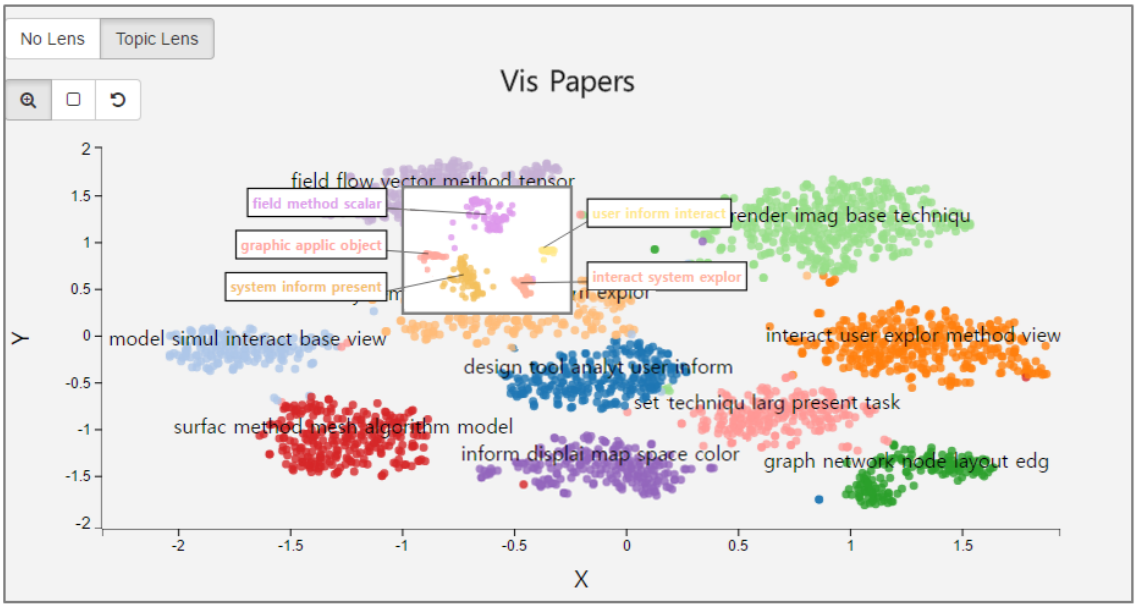


Figure 3.7: Screenshot of TopicLens. After the initial topic modeling is conducted, the result is visualized with a scatterplot. In this view, coordinates of a document are determined by similarity to other documents and topic membership is encoded with the color of the circles. Each topic cluster is labeled with the top-*k* keywords. The small rectangle is the area of the lens, and the region under the lens is recalculated to reveal the underlying topics.

the new, real-time algorithms consistent with the original positioning, a semi-supervised 2D embedding algorithm was developed [58].

Several scenarios demonstrating the capability of TopicLens using real-world datasets were suggested using the developed prototype.

Chapter 4: Gatherplots: Overcome the Overplotting

ParallelSpaces explored the meaning of the cluster by changing axis in the similarity dimension and attribute dimension. Some of the attributes are categorical. When the variable for the axis is a categorical variable, an overplotting will happen. To mitigate this problem, we present the follow-up work called Gatherplots. In gatherplots, we divide the space recursively to layout all the objects without overplotting.

Scatterplots are a common tool for exploring multidimensional datasets, especially in the form of scatterplots matrices (SPLOMs). However, scatterplots suffer from overplotting when categorical variables are mapped to one or two axes, or the same continuous variables are used for both axes. Previous methods such as histograms or violin plots for these cases aggregate marks, which makes brushing and linking difficult. To improve this, we propose gatherplots, an extension of scatterplots to manage overplotting for categorical data, while keeping individual object identities. In gatherplots, every data point that maps to the same position coalesces to form a stacked entity, thereby making it easier to see the overview of data groupings. The size and aspect ratio of data points can also be changed dynamically to make it easier to compare the composition of different groups. In the case of a categorical variable vs. a categorical variable, we propose a heuristic to decide bin sizes for optimal space usage. This means that make better use of visual space to

show the overall distribution. To validate our work, we conducted a crowd-sourced user study that shows that gatherplots enable users to judge the relative portion of subgroups more quickly and more correctly than when using jittered scatterplots.

4.1 Introduction

Scatterplots—-one of the most widely used types of statistical graphics [38, 59, 60]—are commonly used to visualize two continuous variables using visual marks mapped to a two-dimensional Cartesian space, where the color, size, and shape of the marks can represent additional dimensions. It can also be used for exploring multidimensional datasets in the form of scatterplot matrices (SPLOM), where all the possible combinations of axes are presented in table form. However, scatterplots are so-called *overlapping* visualizations [61] in that the visual marks representing individual data points may begin to overlap each other in screen space in situations when the marks are large, when there is insufficient screen space to fit all the data at the desired resolution, or simply when several data points share the same value. In fact, realistic multidimensional datasets often contain categorical variables, such as nominal variables or discrete data dimensions with a small domain, which lead to many data points being mapped to the exact same screen position. This kind of overlap is known as *overplotting* (or *overdrawing*) in visualization, and is problematic because it may lead to data points being entirely hidden by other points, which in turn may lead to the viewer making incorrect assessments of the data. As can be seen in Figure 4.1, there are three situations for mapping variables to axes in scatterplots when overplotting is inevitable:

- **Categorical vs. continuous:** line patterns ((b) and (c));
- **Categorical vs. categorical:** dot patterns (d);
- **Same continuous:** diagonal line patterns (a).

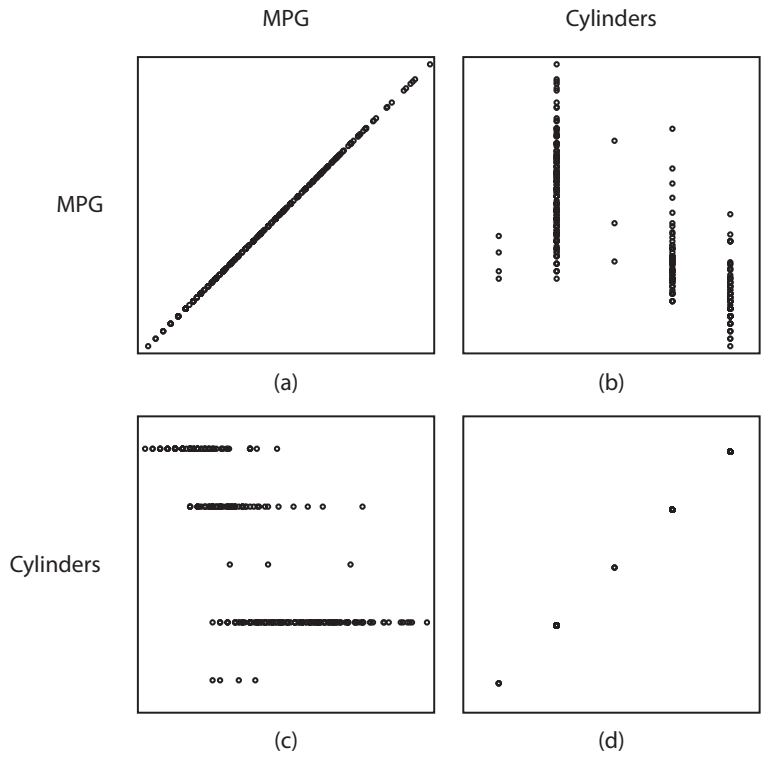


Figure 4.1: A scatterplot matrix for a car dataset with one continuous variable *MPG* and one categorical variable *Cylinders* showing limitations of scatterplots when managing categorical variables. In (a), a scatterplot with the same variable for both axes results in a diagonal line. In (b) and (c), a scatterplot with a continuous vs. a categorical variable results in horizontal or vertical line patterns. In (d), a scatterplot with two categorical variables results in a dot pattern.

Several approaches have been proposed to address this problem [62], the most prominent being transparency, jittering, and clustering techniques. The first, changing transparency, does not so much address the problem as sidestep it by making the visual marks semi-transparent so that an accumulation of overlapping points are still visible. However, this does not scale for large datasets, and also causes blending issues if color is used to encode additional variables. Jittering perturbs visual marks using a random displacement [63] so that no mark falls on the exact same screen location as any other mark, but this approach is still prone to overplotting for large data. It also introduces uncertainty in the data that is not aptly communicated by the scatterplot since marks will

no longer be placed at their true location on the Cartesian space. While the above methods maintains the identities of all data points, there are other approaches that attempt to organize overlapping marks into visual groups that summarize their distribution, such as histograms, violin plots, and kernel density estimation (KDE) plots [64–66]. However, this comes at the cost of losing the identity of individual points, which can be problematic in filter or search tasks. For example, brushing and linking of data points is difficult in histograms [66].

In this paper, we propose *gatherplots* to overcome overplotting for scatterplots. Gatherplots generalizes the linear mapping used by scatterplots by partitioning the graphical axis into segments based on the data dimension, and gathering points into *stacked groups* for each segment, thereby avoiding overplotting. This means that gatherplots relax the continuous spatial mapping traditionally used for a graphical axis; instead, each discrete segment occupies a certain amount of screen space that all maps to the exact same data value. This is also visually communicated using graphical brackets on the axis that show the value for each segment (Figure 4.2(b)).

The contributions of our paper are the following: (1) the gatherplot visualization technique, which extends scatterplots to categorical variables while maintaining individual objects; (2) a practical web-based implementation of gatherplots that supports multidimensional visual exploration; and (3) results from a crowdsourced user study on the effectiveness of different modes of gatherplots. In the remainder of this paper, we first review the literature on scatterplots and overplotting. We then present gatherplots and discuss its design rationale. This is followed by our crowdsourced evaluation. We close with conclusions and our future plans.

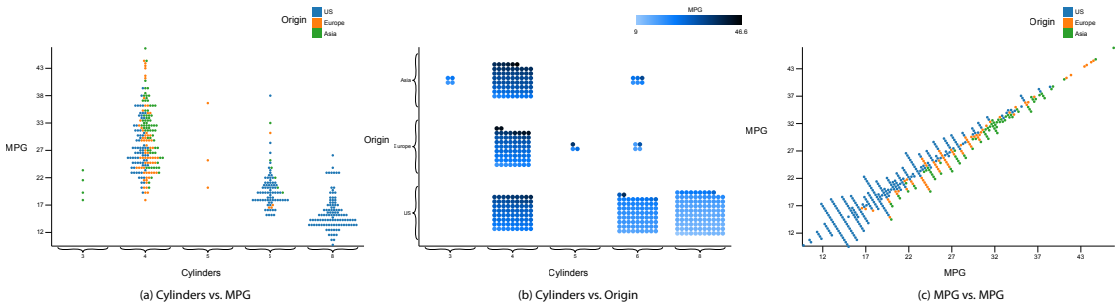


Figure 4.2: Gatherplots showing a dataset related to cars, yielding overplotting in normal scatterplots. The gatherplot in (a) shows Cylinders (categorical) vs. MPG (continuous). The gatherplots show the overall distribution of MPG values of cars with different cylinders. The brackets on the X-axis are used to indicate that the interval within the brackets represent the same value in the data. The gatherplot in (b) shows Cylinders (categorical) vs. Origin (categorical). The gatherplots partition the graphical axes into intervals and stacks points into groups for each interval. In (c), both X-axis and Y-axis show the same continuous variable (MPG). In scatterplots, all these cases create overplotting, which results in points being mapped to lines or dots.

4.2 Background

Our goal with gatherplots is to generalize scatterplots to a representation that maintains its simplicity and familiarity while eliminating overplotting. With this in mind, below we review prior art that generalizes scatterplots for mitigating overplotting. We also discuss related visualization techniques specifically designed for categorical variables.

4.2.1 Characterizing Overplotting

While there are many ways to categorize visualization, Fekete and Plaisant [61] introduced a classification particularly useful for our purposes that splits techniques into two types:

- **Overlapping visualizations:** No layout restrictions on visual marks is enforced, leading to overplotting. (Scatterplots, node-link diagrams, parallel coordinate plots.)

- **Space-filling visualizations:** Layouts that fill the available space to avoid overlap. (Treemaps, matrices, maps.)

Fekete and Plaisant [61] investigated the overplotting phenomenon for a 2D scatterplot, and found that it has a significant impact as datasets grow. The problem stems from the fact that even with two continuous variables that do not share any coordinate pairs, the size ratio between the visual marks and the display remains more or less constant. Furthermore, most datasets are not uniformly distributed. This all means that overplotting is bound to happen for realistic datasets.

Ellis and Dix [62] survey the literature and derive a general approach to reduce clutter. According to their treatment, there are three ways to reduce clutter in a visualization: by changing the visual appearance, by distorting the visual space, or by presenting the data over time. Some trivial but impractical mechanisms they list include decreasing mark size, increasing display space, or animating the data. Below we review practical approaches based on appearance and distortion.

4.2.2 Appearance-based Methods

Practical appearance-based approaches to mitigate overplotting include transparency, sampling, kernel density estimation (KDE), and aggregation. Transparency changes the opacity of the visual marks, and has been shown to convey overlap for up to five occurrences [67]. However, there is still an upper limit for how much overlap is perceptible to the user, and the blending caused by overlapping marks of different colors makes identifying specific colors difficult. Sampling uses stochastic methods to statistically reduce

the data size for visualization [68]. This may reduce the amount of overplotting, but since the sampling is random, it can never reliably eliminate it. Furthermore, one of the fundamental strengths in scatterplots is its ability to show outliers effectively, whereas sampling will most likely eliminate all outliers.

KDE [69] and other binned aggregation methods [64–66, 70] replace a cluster of marks with a single entity that has a distinct visual representation. Splatterplots [65] overcome this by combining individual marks with aggregated entities, using marks to show outliers and aggregated entities to show the general trends. While this technique is effective for overplottin continuous variables, it was not designed to handle cases with categorical variables. The pioneering generalized plot matrices (GPLOMs) [66] were proposed to solve this particular problem by adopting non-homogeneous plots into a matrix. The technique uses a histogram for categorical vs. continuous variables, and a treemap for categorical vs. categorical variables. While effective in providing overview, it loses logical compatibility with scatterplots since it no longer maintains object identity, meaning that each visual mark no longer represents a single data point.

4.2.3 Distortion-based Methods

Distortion-based techniques avoids overplotting by changing the spatial mapping of the space and has the advantage that it keeps the identity of individual data points. The canonical distortion technique is jittering, where a random displacement is used to subtly modify the exact screen space position of a data point. This has the effect of spreading data points apart so that they are easier to distinguish. However, most naïve

jittering mechanisms apply the displacement indiscriminately to all data points, regardless of whether they are overlapping or not. This has the drawback of distorting all points away from their true location on the visual canvas, and still does not completely eliminate overplotting.

Bezerianos et al. [71] use a more structured approach to displacement, where overlapping marks are organized onto the perimeter of a circle. The circle is grown to a radius where all marks fit, which means that its size is also an indication of the number of participating points. However, this mechanism still introduces uncertainty in the spatial mapping, and it is also not clear how well it scales for very dense data. Nevertheless, it is a good example of how deterministic displacement can be used to great effect for eliminating overplotting.

Trutschl et al. [63] propose a deterministic displacement ("smart jittering") that adds meaning to the location of jittering based on clustering results. Similarly, Shneiderman et al. [72] propose a related structured displacement approach called *hieraxes*, which combines hierarchical browsing with two-dimensional scatterplots. In hieraxes, a two-dimensional visual space is subdivided into rectangular segments for different categories in the data, and points are then coalesced into stacked groups inside the different segments. This work inspired our gatherplots technique, which refines the layout and design of hieraxes further.

4.2.4 Visualizing Categorical Variables

While we have already ascertained that scatterplots are not optimal for categorical variables, there exists a multitude of visualization techniques that are [73–75]. Simplest among them are histograms, which allows for visualizing the item count for each categorical value [76]. Boxplots and violin plots show the distribution of continuous variables over categorical variables [77]. While hieraxes, histograms, and treemaps are effective in dealing with categorical variables, it is difficult to extend these to continuous vs. categorical variables. One way is to apply binning to continuous variables to create groups of values. However, the optimal number of bin depends on statistical characteristics of the data and the required task. Dot plots by Wilkinson [78] renders continuous univariate variables without overplotting by stacking nodes within dot size. Dang et al. [79] extended this to scatterplots by stacking nodes whose values are similar in 3D visual space. These pioneering works provide the theoretical background for the determination of optimal bin size for gatherplots.

Another method for visualizing categorical data that is of practical interest is for making inferences based on statistical and probabilistic data. Cosmides and Toody [80] used frequency grids as discrete countable objects, and Micallef et al. [81] extend this with six different area-proportional representations of categorical data organized into different classes. Huron et al. [82] suggested sedimentation as metaphor where individual objects coming from a data stream gradually transforms into aggregated areas, or strata.

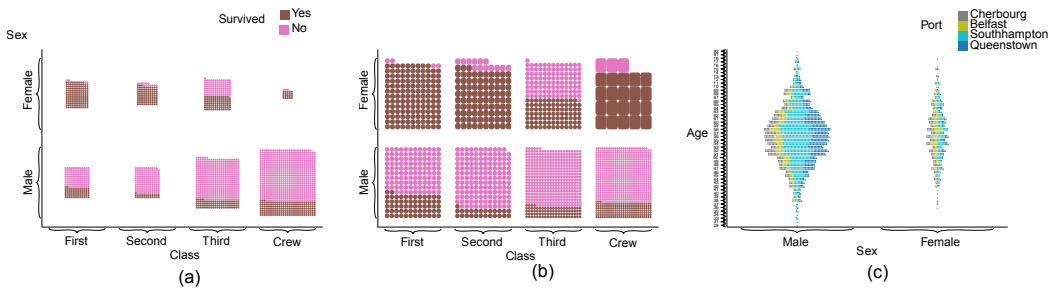


Figure 4.3: Main layout modes for gatherplots: (a) absolute mode with constant aspect ratio, which arranges items following the aspect ratio of given area; (b) normalized mode of (a); and (c) streamgraph mode, where each cluster maintains the number of element in the shorter edge, making it easier to see the distribution of the subgroups along the Y axis.

4.3 Gatherplots

Gatherplots are a distortion-based extensions of scatterplots that alleviate overplotting by gathering data points into *stacked groups*, thereby eliminating overplotting without losing the identity of individual data points. Compared to jittering, which relies on random permutation, gathering organizes visual marks according to visual features, so that the resulting group of objects forms a meta-object. According to Haroz et al. [83], grouping marks by feature helps in performing perceptual (and visual statistical) tasks such as finding outliers, counting items, seeing trends, and so on. The technique is particularly designed for visualizing categorical variables. Below we discuss the open design parameters for the technique, including layout, aspect ratio, and item shapes.

4.3.1 Layout

Gatherplots eliminate overplotting by gathering marks with similar visual properties into *stacked groups*. This is inspired by previous works such as hieraxes [72] or

frequency grids [80, 81]. However, there are many design possibilities for organizing the visual representation depending on the context, especially on the size distribution of each groups, the aspect ratio of assigned space, and the task at hand. As a result, we derive the following three layout modes (see Figure 4.3):

- **Absolute mode:** Here stacked groups are sized to follow the aspect-ratio of the assigned region. The size of the items are determined by the maximum length dots which can fill the assigned region without overlapping. This means with the same assigned space, the groups with the maximum number of members determines the overall size of the nodes (Figure 4.3(a)).

- **Normalized mode:** In this mode, the mark size and aspect ratio is adapted so that every stacked group has equal dimensions. This is a special mode to make it easier to investigate ratios when the user is interested in the relative distributions of subgroups rather than the absolute number of members. Items also change their shape from a circle (absolute mode) to a rounded rectangle (Figure 4.3(b)).

  This normalized mode is useful for two specific tasks:

  - Finding the ratio of the subgroups in a group (Figure 4.3). Because groups of different size are normalized to the same size, any comparison in area results in a relative comparison, which can aid statistical Bayesian reasoning [81].
  - Finding the distribution of outliers. When there are many items on the screen for absolute mode, all marks must be reduced in size. This can make outliers hard to locate. When normalized mode is used, the outliers are expanded to fill the assigned space, making them easier to see.

- **Streamgraph mode:** Here stacked groups are reorganized so that they maintain the same number of elements in their shorter edge. This mode is used for regions where the ratio of width and height are drastically different (in our prototype implementation, we use a heuristic threshold aspect ratio value of 3 for activating this mode). This means there are usually many times more groups in the axis in parallel with shorter edges. A good example is for visualizing the population distribution with regards to gender and age; the resulting gatherplot approaches ThemeRiver [84] as the number of entities increases (Figure 4.3(c)).

The choice between absolute and streamgraph mode happens automatically based on the aspect ratio of assigned space and without user intervention. Therefore, only a simple interaction is required to toggle between absolute and normalized mode.

Maintaining the aspect ratio of all stacked groups means that the size of the group is represented by its area. The length of the group is only used in special cases when the aspect ratio is very high or low. According to Cleveland and McGill [8], length is far more effective than the area for graphical perception. However, Figure 4.4 shows the three problems associated with layout to enable length-based size comparison. In this view, the items are stacked along the vertical axis to make the size comparison along the horizontal axis easier. The width of rectangle is all set to be equal to so that the length can represent the size of subgroups. However, they show drastically different shape of line vs. rectangle, which may cause users to lose concept of equality. Furthermore, to make length-based comparison easier, the stacking should be aligned to one side of the available space: left, right, top, or bottom.

In this case, the bottom is selected to make it easier to compare along the X axis. However, this creates two additional problems. The first problem is that the center of mass of each stacked group is different, so that the concept of belonging to the same value can be misleading. The second problem is that choosing alignment direction is arbitrary and depends on the task. For example, in this view it is more difficult to compare along the Y axis. In this sense, this layout is biased to the X axis, while sacrificing the performance along the Y axis. For this reason, the most general choice is to use center alignment with aspect ratio resembling the assigned range to avoid bias.

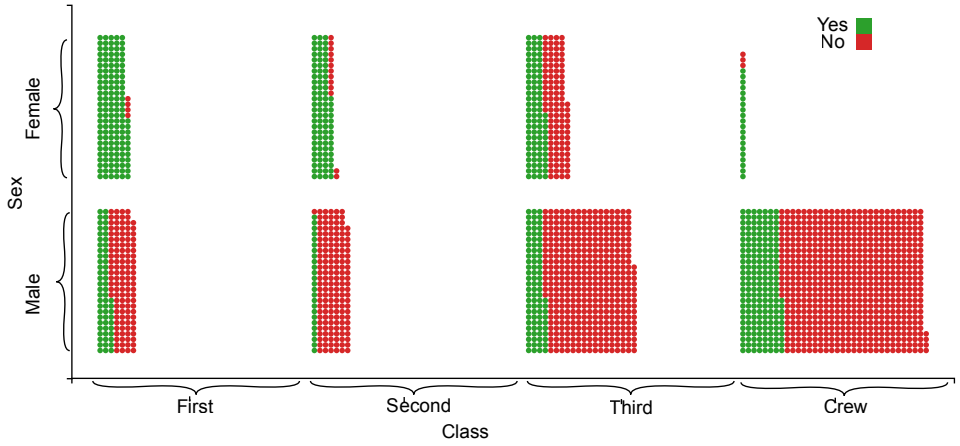


Figure 4.4: Stacked group layouts for gathering. This layout supports comparing group sizes; comparing the length yields the size because the height of stacked groups is all fixed.

4.3.2 Managing Continuous Variables

To use gatherplots for continuous variables, we apply binning to partition the variable into discrete intervals. The resulting visualization resembles dots plots by Wilkinson [78], where bin size is equal to dot size. The size of individual bins is important when applying binning because it determines the spatial accuracy and legibility of the

visualization.

Wilkinson proposed $.25n^{-1/2}$ as the optimal dot size for dot plots. This creates reasonable dot plots for fixed aspect ratio of 5 to 1, which is common in statistical charts assuming normal distribution of nodes. However, gatherplot requires two different assumptions: First, the aspect ratio varies according to the space given to the categorical variables. Second, the dot size or bin size is determined by the global maximum in the dataset, which may not be in the same cluster. Furthermore, because bin size is the same as dot size, selecting bin size can be thought of as a trade-off between accuracy and legibility. Using very small bin size and dot size increases the spatial accuracy, but results in poor legibility, and vice versa. Balancing accuracy vs. legibility is common in visualization for large datasets; for example, splatterplots limit the information shown to users based on the available visual space [65]. Similarly, gatherplots chooses bin size based on spatial accuracy and legibility. When the visual space is small, we use a comparably large bin size to increase dot size, thus resulting in poor spatial accuracy and high legibility; for larger space allocations, the bins can be made smaller to increase accuracy without loss of legibility. This is shown in Figure 4.5.

Figure 4.6 (a) shows how gatherplots handle the situation when continuous variables are assigned to *both* axes, causing both to be binned. The plot is using normalized mode with two random variables. The normalized mode makes it easier to identify the outliers and the distribution of outliers. Furthermore, the case of scatterplots with the same continuous variables on both axes can be treated as a special case of continuous vs. categorical variables. Here, the gatherplot is rotated to maintain integrity with scatterplots (Figure 4.2 (c)).

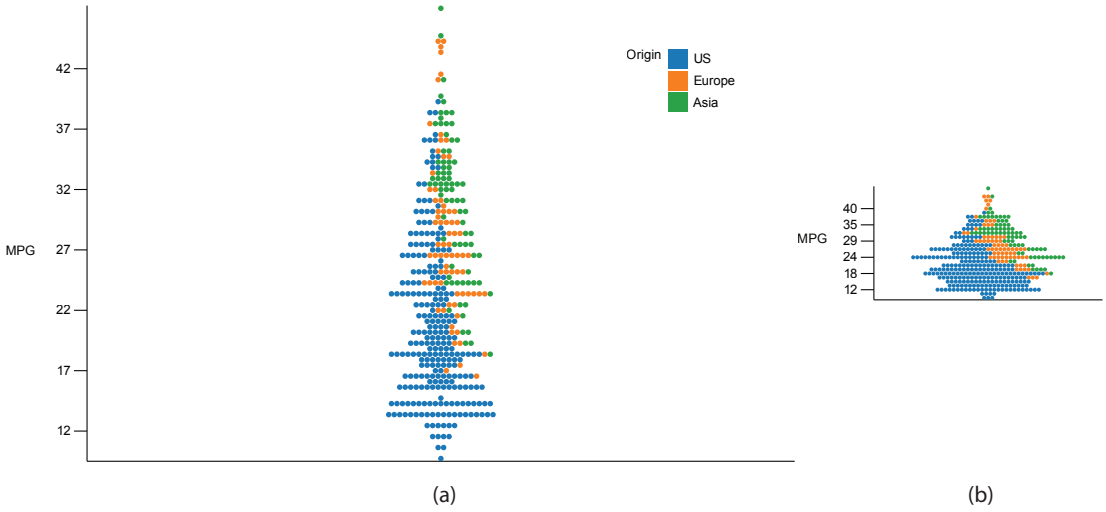


Figure 4.5: Choosing optimal bin size based on available display space. In (a), there is enough space so that the dot size can be maximized, improving spatial accuracy. In comparison, in (b) the assigned space is small, so the dot size is determined so that the most crowded bin interval will fit within the width of assigned space. This results in the two different overview, even though the two plots have identical aspect ratio.

One limitation of gatherplots is that it requires binning to manage a continuous variable, yet binning creates arbitrary boundaries that can be misleading. However, combining gatherplots with scatterplots makes this problem less severe.

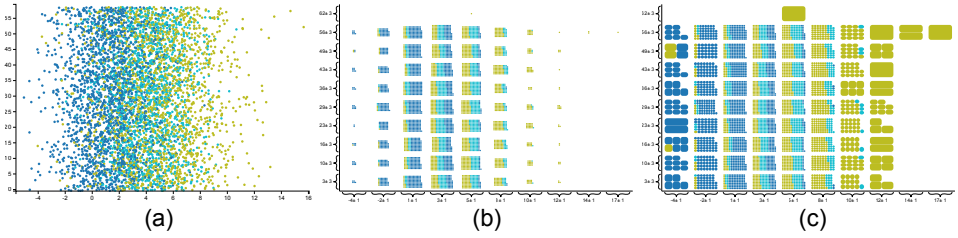


Figure 4.6: Using gatherplots to manage overplotting. (a) shows a scatterplot with 5,000 random numbers with severe overplotting in the center area. In (b), gathering is applied to create a more organized view. However, the gathering resizes the items so small that it becomes difficult to detect outliers. (c) shows normalized mode, where the outliers are enlarged. This makes identifying the distribution of sparse regions easier.

4.3.3 Undefined Axis

Traditionally, scatterplots have been used to see the correlation of two variables. However, for a multidimensional exploration task, one subtle difficulty is when the user wants to see only the effect of a single variable, without definition in other axes. In gatherplots, an undefined axis results in the aggregation of all nodes in one group, which is a logical extension. Figure 4.7 shows an example of this using a dataset on survivors of the Titanic.

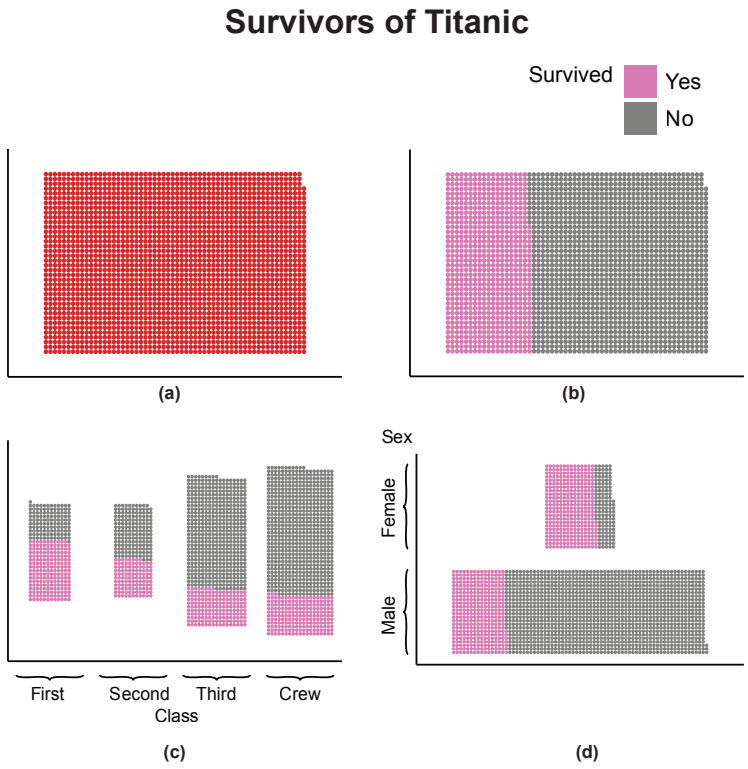


Figure 4.7: Gatherplot showing survivors of the Titanic. (a) All people on board. Note that the X and Y axis are not defined. (b) Survivors with color coding. (c) Distribution of survivors over class variables. Here the Y axis is undefined. (d) Distribution of survivors over the gender. Here the X axis is undefined.

4.3.4 Visual Design

Gatherplots build on the same visual language as scatterplots. However, some aspects are different; below we discuss our design choices for visual mark shape as well as tick marks.

4.3.4.1 Visual Marks

Scatterplots typically use a small circle or dot as a visual representation for items, but many variations exist that use glyph shapes to convey multidimensional variables [59, 85–88]. However, in normalized mode, sometimes the aspect ratio of visual marks changes according to the aspect ratio of the space assigned to that value. Also, as gathering changes the size of marks to fit in one cluster, sometimes the marks size becomes too small or too large compared to other marks. This results in several unique design considerations for item shapes. After trying various design alternatives, we recommend using a rectangle with constant rounded edge without using stroke lines. Using constant rounded edge allows the nodes to be circular when the mark is small, as in Figure 4.3(b), and a rectangle to show the degree of stretching, as shown in Figure 4.3(b).

4.3.4.2 Interval Tick Marks

Because we are representing ranges rather than single points, the single line type tick marks for scatterplots are not appropriate for gatherplots; instead, ticks should communicate the partitioned segments on the axes. Without this visual representation, when the user is confronted with a number, it can be confusing to determine whether adjacent

nodes with different offset has same value or not. After considering a few visual design alternatives, we recommend a bracket type marker for this purpose. Figure 4.8 shows various types of markers for range representation. The bracket is optimal in that it uses minimal ink and creates less density with adjacent ticks.

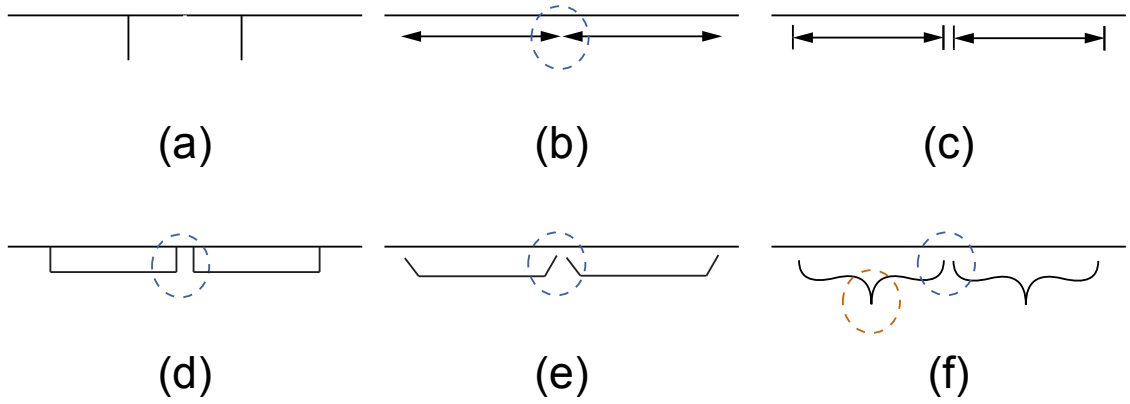


Figure 4.8: Various tick mark types. The blue dotted region represents the area between adjacent tick marks. (a) is a typical line type tick mark for scatterplots. (b) lacks guidelines, which will make anchoring easier. (c) creates a packed region between adjacent marks. (d) uses less crowded region in this region, but (e) is the least crowded. (f) is the final recommendation, with the data label in the orange region.

4.3.5 Interaction

Gatherplots support the same types of interactions as scatterplots. However, some additional interaction techniques are required to specifically control the gathering transformation.

For example, when exploring multidimensional datasets, it is crucial to have a mechanism to filter unwanted data. To support this process in gatherplots, we provide an optional mechanism to go back to the original continuous linear scale function. We allow each axis tick have an interactive control to be filtered out (minimize) or focused

(maximized). This is called *axis folding*, because it can be explained mentally by a folding paper. When minimized or folded, the visualization space is shrunk by applying linear scales instead of non-linear gather scales. This results in overplotting, as if a scatterplot was used for that axis. Maximization simply folds all other values except the value of the interest to assign maximum visual space to that value. Figure 4.9 shows axis folding applied to third class adult passengers in the Titanic dataset.

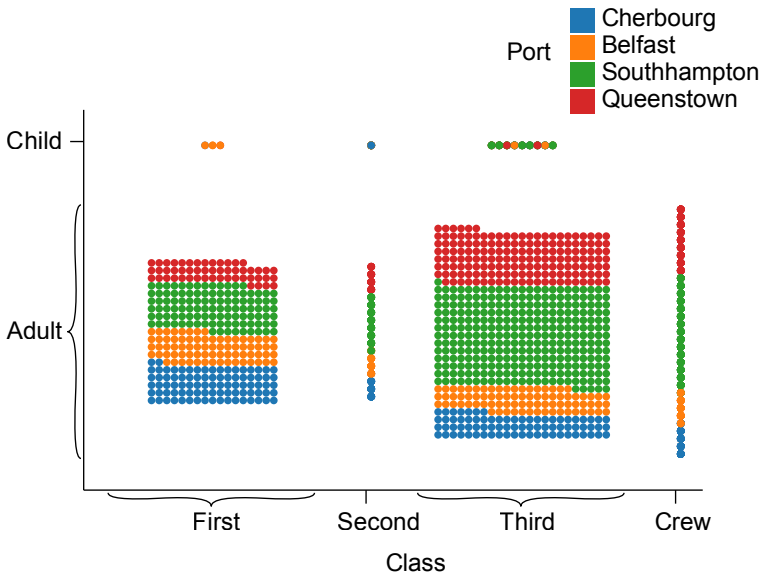


Figure 4.9: Survivors of the Titanic using gatherplots. The X axis is class of passengers, where second class passengers and crew are minimized. The Y axis is age, where the adult value is maximized. This view makes it easy to compare first class adults and third class adults. Note that even in the minimized state, we can get an overview about the second class and crew by the color line, which communicates the underlying distribution. This is due to sorting over the color dimension.

4.4 Implementation

We have implemented a web-based demonstration of gatherplots using D3.js[1] and Angular.js[2]. The prototype can be accessed online at http://www.gatherplot.org, and allows users to load various datasets into a gatherplot. The visualization can be compared to scatterplots and jittered scatterplots with a single click. In the top right area, an interactive guide is provided where users can follow step-by-step instructions in a guided tour of gatherplots.

4.5 Evaluation

The gatherplots technique was developed to overcome many of the limitations of conventional scatterplots. To validate its effectiveness, in particular its different layout modes, we conducted an evaluation study using the technique for categorical vs. categorical variables. Crowdsourcing platforms have been widely used and have shown to be reliable platforms for evaluation studies [89,90]. Therefore, we conducted our experiment on Amazon Mechanical Turk. [3] This also gave us the opportunity to study the utility of the technique for the general population, who do not have specific statistical training.

[1] http://www.d3js.org

[2] http://www.angularjs.org

[3] https://www.mturk.com

4.5.1 Experiment Design

Jittered scatterplots were selected as baseline condition, as it is a widely accepted standard technique maintaining consistency with scatterplots. We also wanted to measure the efficiency of different modes of gatherplots. Therefore, we designed the experiment to have four conditions: scatterplots with jittering (jitter), gatherplots with absolute mode (absolute), gatherplots with normalized mode (normalized), and gatherplots with one check button to switch between absolute and normalized mode (both). We adopted a between-subject design to eliminate learning effect by experiencing other modes.

4.5.2 Participants

A total of 240 participants (103 female) completed our survey. Because some questions asked about concepts of absolute numbers and probability, we limited demographic to the United States to remove the influence of language. To ensure the quality of the workers, the qualification of workers were the approval rate of more than 0.95 with number of hits approved to be more than 1,000. Only three of 240 participants did not use English as their first language. 119 people had more than bachelor's degree, with 42 people having high school degree. We filtered random clickers, if the time to complete one of questions was shorter than a reasonable time, 5 seconds. This yielded a total of 211 participants.

4.5.3 Task

As scatterplots can support various types of tasks, it is difficult to come up with a representative task. After reviewing tasks for categorical variables, we selected three types of tasks such as retrieving value as a low-level task, and comparing and ranking as a high-level task. For the comparing and ranking task, two different types of questions were asked: the tasks to consider absolute values such as frequency and tasks that consider relative values such as percentage. Therefore, for one visualization, 5 different questions were generated. For gatherplots, our interest is more about the difference between questions considering absolute values and relative values. The five types of questions are as follows:

- **T1:** retrieve value considering one subgroup.
- **T2:** comparing absolute size of subgroup between groups.
- **T3:** ranking absolute size of subgroup between groups.
- **T4:** comparing relative size of subgroup between groups.
- **T5:** ranking relative size of subgroup between groups.

To reduce the chance of one chart being optimal by luck for specific task, two charts of same problem structure were provided. Eventually, the resulting questions were 10 for each participant. Each question was followed by the question asking confidence of estimation with a 7-point Likert scale, and the time spent for each question was measured.

4.5.4 Hypotheses

We believe that different types of tasks will favor from different type of layouts. Therefore our hypotheses are as follows:

H1 For retrieving value considering one subgroup (Type 1), absolute, normalized, both mode reduces the occurrence of the error than jitter mode.

H2 For tasks considering absolute values (Type 2 and 3), the absolute mode reduces the error.

H3 For tasks considering relative values (Type 4 and 5), the normalized mode reduces the error.

4.5.5 Results

The results were analyzed with respect to the accuracy (correct or incorrect), time spent, and confidence of estimation. Based on our hypotheses, we analyzed the different modes of layout for each type of question: retrieve value, absolute value task, and relative value task.

4.5.5.1 Accuracy

The number and percentage of participants who answered correct and incorrect answers are shown in Figure 4.10. Eventually, we had 42 participants for jitter, 56 participants for absolute, 56 participants for normalized, and 57 participants for interactive mode.

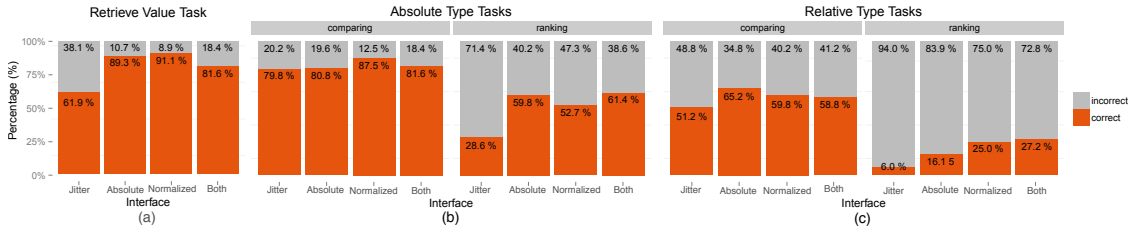


Figure 4.10: (a) The percentage of participants who have got the answer correct for retrieving value task. (b) The percentage of participants who have got the answer correct for absolute type tasks for comparing and ranking. (c) The percentage of participants who have got the answer correct for normalized type tasks for comparing and ranking.

As the measure for each question was either correct or incorrect, a logistic regression was employed. For the retrieving-value task (Type 1), both the the absolute mode and normalized mode had significant main effects (Wald Chi-Square = 18.58, $p < 0.01$, Wald Chi-Square = 21.05, $p < 0.01$, respectively) with a significant interaction effect (Wald Chi-Square = 19.53, $p = 0.03$) (H1 confirmed). For absolute-value tasks (Type 2 and 3), both the the absolute mode and normalized mode had significant main effects (Wald Chi-Square = 10.35, $p < 0.01$, Wald Chi-Square = 10.35, $p < 0.01$, respectively) with a significant interaction effect (Wald Chi-Square = 4.31, $p = 0.03$) (H2 confirmed). For relative-value tasks (Type 4 and 5), only the normalized mode had a significant effect (Wald Chi-Square= 5.10, $p = 0.02$) (H3 confirmed).

4.5.5.2 Completion Time

The time spent (in seconds) for each question was compared using mixed-model ANOVA with repeated measures. For the retrieving-value task, on average, the time spent (sec) for each interface was for jitter (44.26), absolute (56.84), normalized (52.45), and both (56.57). There was no significant difference between interfaces ($p > 0.05$ for all

cases).

For the absolute-value task (Type 2 and 3), on average, the time spent (sec) for each interface was for jitter (30.74), absolute (32.3), normalized (33.6), and both (47.91). The interface had a significant main effect ($F(3,207) = 11.5, p < 0.01$). However, when we conducted pairwise comparisons with adjusted p-values using simulation, the only significant difference in time spent was when using the both interface which took longer ($p < 0.01$ for all comparisons).

For relative-value task (Type 4 and 5), on average, the time spent for each interface was for jitter (26.6), absolute (31.12), normalized (31.38), and both (46.78). The interface had a significant main effect ($F(3,207) = 10.12, p < 0.01$). However, when we conducted pairwise comparisons with adjusted p-values using simulation, the only significant difference in time spent was when using the both interface which took longer ($p < 0.01$ for all comparisons).

4.5.5.3 Confidence

The 7-point Likert-scale rating was used for the level of confidence on their estimation. For the value-retrieving task (Type 1), Kruskal-Wallis non-parametric test revealed that the type of interface had significant impact on the confidence level ($\chi^2(3) = 74.57 p <$ 0.01). The mean rating for each interface was for jitter (4.8), absolute (6.3), normalized (6.0), and both (6.25). A post-hoc Pairwise Wilcoxon Rank Sum test was employed with Bonferroni correction to adjust errors. The jitter interface was significantly lower than the other three modes ($p < 0.01$ for all cases). There was no difference between absolute,

normalized, and both interfaces.

For absolute-value tasks (Type 2 and 3), Kruskal-Wallis non-parametric test revealed that the type of interface had significant impact on the confidence level ($\chi^2(3) = 18.32, p < 0.01$). The mean rating for each interface was jitter (5.4), absolute (5.7), normalized (5.0), and both (5.8). A post-hoc Pairwise Wilcoxon Rank Sum test was employed with Bonferroni correction to adjust errors. The interface with both mode was significantly higher than normalized and jitter mode ($p < 0.01$ for both), however, no difference with the absolute mode. The interface with absolute mode was significantly higher than normalized and jitter mode ($p < 0.01$).

For relative-value tasks (Type 4 and 5), Kruskal-Wallis non-parametric test revealed that the type of interface did not have significant impact on the relative tasks ($\chi^2(3) = 4.1, p = 0.2$). The mean rating was jitter (4.7), absolute (4.9), relative (4.9), and both (4.8).

One possibility explaining this result is that relative task is more difficult than other tasks. The low correct percentage of questions are also shown in Figure 4.10. To see that, we tested the confidence level between task types. Kruskal-Wallis non-parametric test revealed that the type of task had significant impact on the confidence level ($\chi^2(2) = 148.1, p < 0.01$). The mean rating for retrieving value (5.9), absolute (5.5), and normalized (4.8). The post-hoc Pairwise Wilcoxon Rank Sum test was employed with Bonferroni correction to adjust errors, and showed that all three task types were significantly different ($p < 0.01$ for all cases).

4.6 Discussion

Overplotting in scatterplots is a well-known problem, and several existing efforts have studied it, such as splatterplots [65] and GPLOMs []. Compared to these works, our main contribution is that gatherplots maintain the identity of objects. According to Ellis and Dix [62], gatherplots are distortion-based methods, while KDE, histograms, and violin plots are appearance-based methods. In this section, we will discuss trade-offs compared to appearance-based methods, as well as when gatherplots are appropriate and not.

4.6.1 Scalability

As datasets become larger, the scalability of a visualization becomes an important issue. Scatterplots support two main tasks: detecting correlations as well as outliers. Gatherplots are effective in showing correlations as the dataset grows, but this also causes the dot size to shrink, which makes detecting outliers becomes less plausible. Splatterplots [65] handle this by using two different visual representation for dense areas and sparse areas, whereas gatherplots have no such mechanism. In this sense, gatherplots do not scale to large datasets.

Also, as the dataset becomes large, individual object identification becomes less relevant, and the gatherplots resemble histograms or violin plots. Calculating layout of individual objects for gatherplots requires heavy computation compared to appearance-based methods. These computations are hard to justify for a large dataset, because the amount of output information is nevertheless same. However, according to Shneiderman's

visual information seeking mantra—*overview first, zoom and filter* [31]—even for the large dataset, zooming can make the application of gatherplots desirable.

4.6.2 Named vs. Anonymous Objects

In some multidimensional datasets, data points have names, whereas in others they not. For example, in datasets containing entities such as cars or digital cameras, it is a common task for users to search and identify cases that suit their needs. For this case, maintaining the identity of individual data points is important because it enables brushing and linking more easier than for aggregated forms such as histograms.

There are other datasets, such as the famous iris flower dataset, where individual data points do not have names. For such datasets, the benefit of maintaining object identity can be explained using frequency grids. According to Cosmides and Toody [80], the concept of relative percentage is new concept in human evolution, and explicitly showing distribution with discrete countable object is more comfortable for humans.

4.6.3 Visualizing Normalized Data

According to Im et al. [66], one tradeoff in designing GPLOMs was whether axes of the same variable should be scaled to the same range or not. If scaled to the same range, it would be easy to compare to adjacent charts, but results in large vacant spaces for sparse areas. This happens when there are severe imbalances in data distribution. Both options are valid depending on the task at hand, but it is difficult to represent it visually in histograms so that users can see it.

In gatherplots, these two options are supported as absolute vs. normalized modes. Because we show the individual objects as separate visual marks, it is feasible to deliver this information more explicitly. If the size of all data points are the same, users will understand that they all use the same scale, while rendering points with different sizes conveys the information that they are normalized.

4.6.4 Evaluation Limitations

Although scatterplots support several types of tasks such as detecting correlation, clusters, or outliers, in our experiment we decided to test a particular case with categorical data, which has distinctive views compared to conventional scatterplots. Even if this is a narrow case, the purpose of our study was to show the effectiveness of different layout modes in a quantitative way. The results indicated that the users could understand the visualization and accomplish the tasks that should be supported. However, we also observed that the difficulty level was different for each task type. In general, ranking tasks were more difficult than comparison tasks, and questions asking about relative values were more difficult than those about absolute values. Therefore, maintaining similar difficulty level among tasks should also be considered while designing the evaluation.

In our study, we selected scatterplots with jittering as the baseline for comparison because (1) it extends scatterplots to manage overplotting, (2) it maintains individual objects, and (3) it is a well-known technique. However, for future studies it would be also desirable to compare the performance with a purpose-specific technique, such as histograms or hieraxes.

4.7 Conclusion and Future Work

We have proposed gatherplots, an extension of scatterplots, which enable overview without overplotting for multidimensional data, particularly for categorical variables. We discussed several aspects of gatherplots including layout, coloring, tick format, and interactions. We also evaluated the technique with a crowdsourced user study showing that gatherplots are more effective than jittering, and absolute and normalized modes serve specific types of tasks better. We addressed these weaknesses and suggested possible remedies.

We believe that gathering is a general framework that captures the transition between overlapping and space-filling visualizations while maintaining object identities. In the future, we plan on studying the application of this framework to other visual representations. For example, overplotting is a common problem when visualizing categorical variables in a parallel coordinates plot. Parallel sets aggregate elements for the same value of a categorical variable into blocks, but loses the identity of objects. By applying the gathering framework, parallel sets can be reconstructed to render individual lines instead of block lines, which would enable combining both categorical and continuous variables. Also additional interaction techniques can be proposed with a gathering framework. One particular technique employing lens scheme is shown in Figure 4.11.

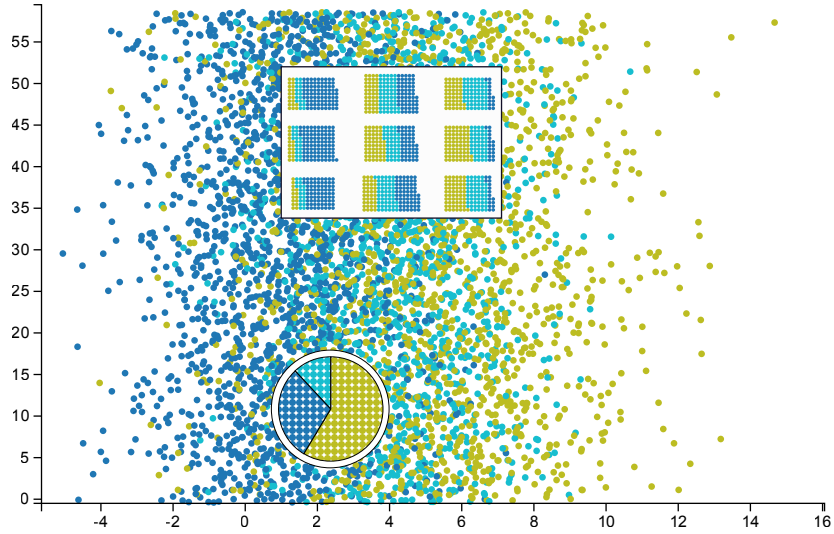


Figure 4.11: Design mock-up of gathering lens showing two random continuous variables. In the upper part, a rectangular gathering lens is applied, which arranges objects within the boundary so that it resembles histograms. In the lower part, a circular lens shows the distribution in a pie chart shape.

Chapter 5: CommentIQ: Building a Custom Mixed Axis

In ParallelSpaces, we used the existing attributes as an axis. However, there are many cases where the existing attributes are not effective. CommentIQ proposes an interactive process for building a custom axes by the mixture of the existing features.

Comments submitted by readers on online news articles provide valuable feedback and additional information for both other readers as well as for reporters. However, due to the varying quality and at times aggressive tone of online comments, news publishers moderate comments by both filtering out low quality comments, and increasingly by selecting and highlighting high-quality comments. This latter practice is motivated by findings that indicate that featuring high-quality comments increases the interest of the readers and promotes the general writing quality of the community. But due to the large number of submitted comments, exhaustive moderation is time-consuming and not scalable for all but the largest and most well-resourced news outlets. In this paper, we report on a design study focused on aiding finding valuable comments using visual analytics. Working closely with publishers, moderators, and reporters from the New York Times, the Washington Post, the Baltimore Sun, and other outlets, we first created a domain characterization for online comment moderation for news articles. We then derived a model for ranking high-quality comments using an annotated comment dataset. Using this model

we designed a web-based visual analytics tool called CommentIQ that generates a visual overview and custom ranked list of comments to be moderated. The CommentIQ tool was developed in an iterative fashion from early interface mockups to full-fledged system with input from domain experts. The full version of the system was evaluated with an extended panel of domain experts, and the feedback we received strongly supports the utility of the tool for moderating online news comments. We furthermore discuss our experience and design lessons for applying visual analytics to this domain.

5.1 Introduction

In September 2013, Vladimir Putin published an op-ed article in the New York Times (NYT).[1] It was an essay critical of the U.S.—some might even say prodding the public. As a result, the comments flooded in: 6,367 of them, in fact. Of those, 4,447 were eventually published along with the piece online, including 85 which were selected as 'NYT Picks', high quality comments with exceptional insights that are highlighted in the commenting interface. What makes this remarkable though is that each of these thousands of comments was read by a human moderator, a trained journalist, at the NYT *before* it was published. The New York Times uses a pre-moderation strategy and employs 13 community managers to read such comments, filter out inappropriate ones before publication, and select NYT Picks for highlight.[2]

Reader-contributed comments are a double-edged sword: while they increase user

[1] http://www.nytimes.com/2013/09/12/opinion/putin-plea-for-caution-from-russia-on-syria.html

[2] http://www.nytimes.com/times-insider/2014/04/17/a-comments-path-to-publication/

engagement, contribute to fostering an online community, and may even provide enriching content to both readers and reporters alike, not all comments are created equal. More specifically, reader comments are often low in quality (in terms of spelling, grammar, or composition), have a tone not commensurate with the news outlet (e.g., aggressive or obscene), and may be intentionally or unintentionally incorrect or misleading. For this reason, while there is a clear value to including reader comments on online articles, there is also a need for comment moderation to ensure that published comments are representative of the news outlets policies. Filtering out low-quality comments only addresses half the issue; top news sources are increasingly also selecting high-quality comments that contribute particularly well to the associated article and which set the tone for the site. Crowdsourced approaches have their own limitations: selections don't convey an editorial voice, there is no central oversight to ensure balance, and selections may exhibit undesirable popularity biases.

Managing and moderating online news comments is a particularly challenging task due to the overwhelming volume of content, as well as the nuance and context that the moderators sometimes need to understand and consider when dealing with sensitive or political issues. Various strategies for mitigating the scale issue have been tried: leaving comments unmoderated devolves quickly, so post-moderation is often employed to allow the community to flag or report low-quality or otherwise inappropriate comments. The pre-moderation strategy at the New York Times produces the highest quality of discourse but is perhaps the most resource intensive. As a result they must limit the number of articles where comments are even allowed, as well as the time window for commenting on those limited articles.

This work presents a design study of a visual analytic tool to augmenting comment moderators' capabilities to effectively scale the selection of *high quality* commentary on online news sites. We present the domain-centered design and development of a visual analytics tool called CommentIQ that is specifically geared towards the tasks and use-cases of the comment moderator. Our design process consisted of highly iterative requirements gathering from domain experts, prototyping, piloting, and development of analytics and visual representations that inform the selection of comments that are editorially interesting. The analytic criteria we incorporate are aligned with journalistic needs reported in the literature [91] and include article and conversational relevance of comments, readability, personal experiences, as well as user attributes like frequency of commenting. Given our understanding of the domain and tasks we explicitly designed CommentIQ, (1) to provide a broad overview of the comment space via various visual scheme, and (2) to be flexible in its comment scoring and ranking methodology so that the tool can be adapted by the end-user expert to fit the specific moderation circumstances.

Our design study is underscored by an evaluation with seven professional community editors and moderators at leading local and national news outlets who used CommentIQ to identify interesting comments in different articles. The evaluation provided insights into how the various analytic dimensions, filters, and visualizations that we built enabled and supported the comment moderation task. This evaluation with domain experts allows us to reason about current practices, our potential to augment these practices with visual analytics, and offers guidelines for the future design and development of computational journalism tools.

Our contributions include (1) the characterization of the comment moderation do-

main, which includes design needs, (2) a web-based moderation tool called CommentIQ that was designed and developed to meet these needs, and (3) the evaluation, which validates the use of criteria and various visualizations in the editorial context.

5.2 Background

This work contributes to a growing body of research in the area of computational journalism, which includes tools that are tailor- designed to suit journalistic tasks and workflows, and to take into account professional norms and use-cases of journalists. Related work in this vein includes the recently published Overview [92] system, which incorporates document visualization capabilities into an investigative journalism tool that has enabled important stories to be uncovered. Another visual analytic tool in this domain is the SRSR prototype [93] that utilizes social media analytics of twitter accounts to enable journalists to find eyewitness or other sources during various breaking news events. Following in the footsteps of these previous studies, in this work we present a design study in this domain. Yet, unlike previous design studies which have focused on reporters, we target our system—CommentIQ—at an underexplored but increasingly important task and sub-population within this domain: comment moderators.

In designing the CommentIQ system we were informed by related work in three areas which we detail further next, including *community moderation*, *analytics of comment quality*, and *discourse visualization*.

5.2.1 Community Moderation

Discourse quality and incivility in online comment forums is an issue that has not gone unnoticed in the research literature [94–96]. In fact, if low-quality commentary goes unchecked, research has shown that it can lead to detrimental and polarized risk perceptions of scientific information [26]. One approach for dealing with discourse quality is

that of post-moderation: a user community can flag or report comments that they deem inappropriate and these flags can then be reviewed by professionals [95] to determine whether they should be removed. Users can also rate, tag, and vote on comments which feed into end-user interfaces for sorting and filtering comments [97]. A substantial downside to this approach is that it can take a long time for good comments to be identified, and a reliable ranking depends on having enough votes in the system [98].

Studies have shown that users are more interested in engaging with discussion that is moderated [99]. Evidence is mounting which shows that by signalling norms and expectations for behavior, the overall tenor of discourse can be improved. For instance, lower levels of incivility and a greater use of evidence in comments was found when a reporter engaged directly in a news outlet's comment threads on Facebook [100]. In another study, thoughtfulness cues in comments led to participants posting longer comments, spending more time, and writing more relevant comments [27]. The practice of selecting high quality comments by outlets like the New York Times fits within this strategy of social signalling to set community standards. The CommentIQ system was specifically designed with this approach towards comment moderation in mind, thus we focus not on the removal of low-quality comments (which is still a valid problem in its own right), but on the identification of high-quality contributions that could act as cues for a positive feedback loop with the community.

5.2.2 Analytics of Comment Quality

Various efforts have been undertaken to measure and rank the quality of written texts and comments, including both low- quality [101] and high quality written outputs. Natural language processing of text content as well as data analysis of community information (e.g. user history and interactions) have been applied. For instance, Louis and Nenkova [102] predicted the article quality of science journalism based on lower level linguistic features, such as sentence structure. Other work in this domain has considered the measurement of dimensions of text readability [103]. Efforts have tried to automatically predict the quality of online comments, although the reported performance and accuracy of such models makes them difficult to apply practically [104, 105]. In addition to textual features such as informativeness and cohesion of text, user features can also be leveraged to rank comments, such as activity level, history of ratings, and degree to which other people respond [106]. In contrast to Hsu et al. [106], however, we do not use community ratings as ground-truth for quality as this can reflect popularity bias. For our ground truth of "quality" we instead utilize a source of professionally curated and selected comments: the New York Times "Picks" comments.

Studies in the literature describe journalistic efforts to identify high quality contributions from the public, including how letters to the editor are selected [107], how online comments are selected for print publication [108], and how on-air radio comments are chosen at NPR [109]. Specifically in the domain of online news comments, recent work by Diakopoulos [91, 110] has synthesized these journalistic criteria into a set of twelve human-centric candidate criteria including Argument Quality, Criticality, Emo-

tionality, Entertaining, Readability, Personal Experience, Internal Coherence, Thoughtfulness, Brevity, Relevance, Fairness, and Novelty. In this work, we utilize the validated analytic operationalizations of several of these criteria to score comments, including readability, personal experience, brevity (or length), and relevance (including article relevance and conversational relevance). We also derive user-based scores of quality by averaging these criteria over user history. We incorporate understanding from other literature in the domain of online reviews which suggests that a measure of user activity level will be usefully correlated to quality [111]. Finally, we train a model on a set of collected comments to arrive at default weights of these various analytic criteria that can orient moderators towards the top comment candidates.

5.2.3 Discourse Visualization

Previous work on comment or discourse visualization has often approached the issue from the end-user's perspective. For instance, the ForumReader tool [112] was designed to help orient and guide readers to areas of interest within large scale online forums, such as Slashdot. An evaluation found that moderator information such as tags that had been visualized was valuable in helping users navigate in the discussion. Another more recent effort in this area is the ConVisIt system [113], which utilizes flexible user-driven topic modeling to provide an interface that allows for exploration of asynchronous online discussions. The interface allowed users in an evaluation to find more useful and insightful comments than the more standard Slashdot interface. The Arkose system [114] was designed to help visually distill large online discussions into more succinct

summaries. A somewhat related effort is the Opinion Space system, which visualized comments by projecting sets of specially elicited scalar opinions relating to controversial statements [115]. An evaluation found that users reading comments in the system were more engaged in comparison to a baseline interface which simply listed the comments.

The discourse visualization systems presented in the literature are not oriented towards sensemaking of comments that can directly enable better moderation. In contrast, we designed the CommentIQ visual analytic system specifically for comment moderators and we present a persona and characterization of the task and use-cases involved which inform our design. More specifically we use analytics to score comments along various dimensions of interest to journalist moderators as discussed above, and we provide interactive visualizations of these scores including map-based and temporal views that align with user needs and requirements in the domain and help orient moderators towards comments that may be high quality.

5.3 Overview: Analytics for Comment Moderation

Our goal in this work is to explore how visual analytics can be applied to the comment moderation process employed by news outlets. For this purpose, we conducted a multi-phase design study with the purpose of applying both automatic text analytics algorithms as well as interactive visual interfaces to this domain. In this section, we give a general overview of our design process; the following sections discuss each of the phases in detail.

Our design study process was inspired by the *core* phase of the nine-stage frame-

work proposed by Sedlmair et al. [116]. More specifically, our work was organized into four distinct stages roughly matching the discover, design, implement, and deploy stages in that framework:

I **Domain Characterization:** (*discover*) characterizing personas, use-cases, and tasks;

II **Design and Analysis:** (*design*) developing design rationale as well as concrete visual, interaction, and algorithm design;

III **Prototyping and Implementation:** (*implement*) interface prototypes as well as client-side and server-side components; and

IV **In-Field Evaluation:** (*deploy*) validation through domain expert feedback in the field.

5.4 Stage I: Domain Characterization

In designing the CommentIQ, system we adhered to a human-centered design methodology, including undertaking several early semi-structured interviews with domain experts. In particular, we were interested in developing knowledge from our informants that would allow us to (1) build a persona of a comment moderator that would guide our design thinking, and (2) understanding the use-cases and tasks for comment moderation on news sites.

Our interviews were informal and targeted at individuals who were deeply embedded in newsrooms and who had experience moderating online comments and social media in the context of the news. We spoke to eight people from news organizations including

the Washington Post, National Public Radio (NPR), the New York Times, and the Wall Street Journal. Six of the interviews were conducted face-to-face, two were conducted via phone, and all were uncompensated and lasted roughly one hour. Copious notes were taken during the interviews and these were typed and analyzed afterwards to facilitate our design process. We approached the interviews with several questions in mind, aiming to understand more about the comment moderator workflow and goals, editorial criteria for identifying high quality selections, indications of how moderation decisions were made, and challenges, frustrations, and pain points with current tooling.

5.4.1 Persona Development

Personas can be useful design tools that work by capturing and communicating the objectives, motivations, behaviors, and expectations of a group of target users [117]. They are often useful as grounding artifacts to assess how different design options may impact users. Based on our initial informal interviews, we developed a persona of The Comment Moderator (TCM), an archetype reflecting our understanding of the comment moderators we interviewed.

Perhaps more than any other goal, TCM is motivated by a desire to produce a quality discussion. They want to not only remove off-topic, impolite, or critical and unconstructive comments, but also to identify and highlight original ideas, cogent points, or contributions that rise above the noise. They are interested in discovering local voices and top contributors to give them a spotlight and to create a feedback loop where good behavior is rewarded: readers should aspire to have their comment selected. Different

outlets employ different terminology for this idea: at the New York Times such selected comments are termed “NYT Picks”, at the Washington Post they sometimes badge users as “preferred”, and at the Wall Street Journal they refer to them as “featured” comments.

The New York Times is perhaps most sophisticated in their thinking about highlighting perspectives as NYT Picks.[3] The NYT Picks are the most popular comment queue and NYT TCMs try to select those with a broad range of viewpoints. In other words, they strive for diversity in the selections. This might include geographic diversity if that is relevant to the story, or it could other types of diversity such as along political perspectives. They want the selections to be representative but they dont necessarily need to be balanced between different viewpoints.

The overall goal of TCM is to set the tone and maintain the commenting policies to provide a positive atmosphere for discussion. TCM sees comments not as an appendage but as any other piece of journalistic content, and as such they apply an editorial eye: it’s not about finding the “most liked” comment. For instance, they strive for fairness in the editorial standards they apply, and are willing to turn comments off for sensitive topics or stories where they believe civil discussion is impossible. They also seek to build trust with and increase engagement with their community.

5.4.2 Use-Cases and Tasks

In our conversations with comment moderators, we learned of several use-cases and analytic tasks that may be accomplished with comments. These include: (1) exclusion of low quality comments, (2) selection, highlighting, or picking high quality comments, and

[3] http://www.nytimes.com/times-insider/2014/04/17/a-comments-path-to-publication/

(3) taking other journalistic actions based on comment content.

The first task, that of identifying and filtering out low quality comments is one that dominates the analytic workflow for moderators. To a large extent this is about upholding the community standards and providing a venue for discussion that is free from profanity, hate speech, or personal attacks. In many cases, moderators examine flags that are passed in by community (end-user) moderators, or in some cases by automated systems like KeepCon.[4] Sometimes moderators examine the context of a user to see if they are a habitual violator of community norms and as a result may take additional action such as blocking the user. While this is surely an important analytic task which has received some attention in the research literature [101], we instead focus the current work on the newer and underexplored strategy of selecting high-quality comments.

To select high-quality comments for highlighting on a site, moderators consider many different criteria. At the New York Times, they consider five criteria when choosing NYT Picks: overall quality, such as spelling and grammar, argumentation, and literary value, broad representation and diversity of perspective, conversation between two people making opposing points, unexpected short, funny, or unusual commentary, and personal stories and experiences that are relevant to the issue. Our interviews with moderators exposed the importance of flexibility and adaptability in applying these criteria. Different quality criteria apply for different stories and for different communities: there isn't a one-size-fits-all model for when to apply a given editorial criteria, but rather there are many contingencies. Interviews also elucidated an openness to employing automation to help uncover higher quality comments, with an acceptance of some errors and the

[4] http://keepcon.com/

understanding that a human moderator is making the final decision. As such, this task is well-suited to a visual analytic approach.

The final task which moderators might engage in involves some other journalistic action, which might include correcting a story based on a comment, or passing a comment on to a reporter for follow-up. Several moderators we interviewed believed that comments were valuable leads for news reporting. People often write fascinating stories about how they are personally impacted by an issue at hand, and this can fuel additional reporting by journalists. For niche communities or blogs, insiders may sometimes comment with valuable knowledge and insight that would otherwise be unavailable. This task is conceptually similar to the previous task insofar as it is about identifying comments of a specific ilk, but instead of choosing comments that should be published and highlighted it is about using the content of those comments for internal purposes. The essential difference in tasks is thus the final step being one of publication, or one of internal use.

5.5 Stage II: Design and Analysis

Our second stage, based on Sedlmair's *design* stage [116], involved pursuing additional insight and clarity into the development of tools to support comment moderators. Below we describe the design criteria that we developed based on our human-centered requirements gathering, as well as how the various pieces of the CommentIQ design and system support those criteria.

5.5.1 Design Rationale

Based on the survey with comment moderators (TCMs), we found that there are mainly two unmet requirements. First, TCMs need a way to manage a diverse set of qualifications for comments depending on the context of the workflow. The second requirement is to be able to maintain a balanced view when selecting comments. For example, many times moderators were looking for minority opinions when there were strong majority opinions about specific issues at hand; this task involves both requirements, i.e., maintaining balance (2) as well as adapting criteria to a specific workflow (1).

Based on these user requirements, we derived the following design rationale (DR), which are further reflected in Figure **??**.

DR1 **Custom ranked list:** Users should be able to customize rankings based on their own needs and their current context;

DR2 **Score by multiple criteria:** Comments should be able to be automatically scored by multiple user-controlled criteria;

DR3 **Overview and filter:** To represent balanced opinions, the system should show the distribution of both major and minor opinion groups;

DR4 **Learn from user actions:** The system should learn from actions taken by users to have a more refined model.

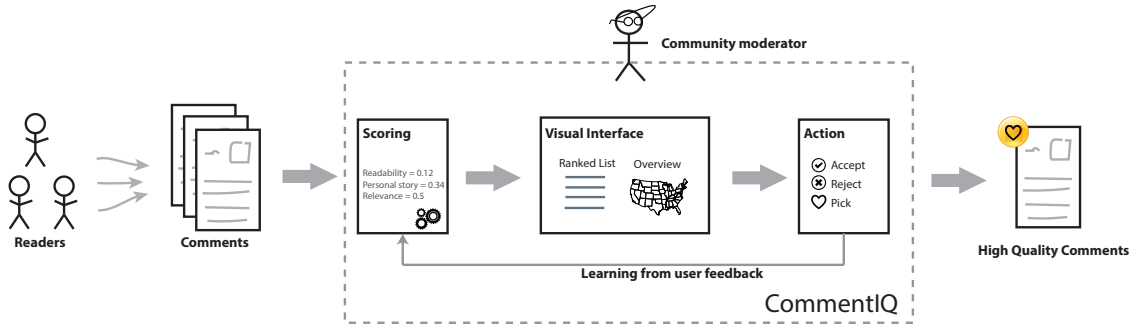


Figure 5.1: Our design learning suggests the following process for selecting high quality comments: (1) comments submitted by readers are scored using multiple criteria; using these scores, (2) we provide a ranked list of comments as well as the distribution of scores and other meta-data to give an overview and ability to filter; (3) user actions and selection rationale are fed back into the system so that we can learn from users

5.5.2 Analytics

A key insight from our human-centered domain characterization work is that the quality of comments is contextually determined. What makes a good comment on a fast-moving breaking news story is different from a topical blog with a more cohesive community around it. This informs our development of analytics-enabling algorithms for CommentIQ because it implies that a one-size-fits-all language model or classifier is not appropriate for predicting comment quality.

As a result, we designed CommentIQ to support different comment contexts flexibly. Instead of using a binary classifier, we employ a customizable weighted ranking of various features (**DR1, DR2**). By allowing the end-user to adjust weights, we put the power in their hands to decide when one feature may be more or less salient for their identification of "quality" in that particular context. In addition, we provide (1) a smart default weighting for the ranking [118], and (2) several presets to the weighting so that the end-user can quickly switch between contexts.

5.5.2.1 Selection of Criteria

Instead of computing a large number of textual features and then doing feature selection, we limited our selection of criteria to be legible by humans as motivated by our literature review. Thus, we selected criteria that were understandable in this specific editorial context. This allows us to provide end-user customization that is more straightforward than traditional text classification (**DR1**). The criteria are based on the content of comments and the history of the user:

- Criteria based on **comments**:

    1. **Article Relevance:** Score representing how relevant a comment is with respect to the article. Relevance is measured by looking at similarity of word feature vectors [110].
    2. **Conversational Relevance:** Relevance with respect to preceding comments. This relevance score is measured by looking at similarity of word feature vectors [110].
    3. **Length:** The size of a comment, measured in terms of the number of words.
    4. **Personal Experience:** Measure of the rate of use of words in Linguistic Inquiry and Word Count (LIWC) categories "I", "We", "Family", and "Friends", which reflect personal (1st and 3rd person pronouns) and close relational (family and friends) references [91].
    5. **Readability:** Measure of how readable a comment is according to a standard index of reading grade level known as SMOG [91, 119].

6. **Recommendation Score:** The number of recommendations a comment has received.

- Criteria based on **user history**:

1. **User Comments per Month:** The average number of comments per month a user has written.
2. **User Comment Length:** The average comment length score for a user across their entire history.
3. **User Personal Experience:** The average personal experience score for a user across their entire history.
4. **User Picks:** The average rate at which a user's comments are selected as NYT Picks
5. **User Readability:** The average readability score for a user across their entire history.
6. **User Recommendation Score:** The average recommendation score for a user across their entire history.

5.5.2.2 Development of Presets

Tuning our ranking thus involves modifying any of the 12 weights, one for each criterion above. In order to provide a smart starting point for the ranking, we trained a classifier to produce a set of default weights. We developed the default ranking weights using an annotated dataset collected via the New York Times' Community API[5]. Each

[5] https://developer.nytimes.com/

comment contains relevant metadata, such as a boolean value of whether the comment is an editor's pick, as well as the recommendation score as measured by community votes. To compute user scores, we also gathered all historical comments for each user.

The recommendation score was removed during the training of the classifier because it is highly correlated to the status of the comment as a NYT Pick. In other words, a comment may have a high recommendation score as a result of being picked, rather than as a reason for being picked. Using the remaining 11 scores as features and an editor's pick as target, we trained a binary classifier.

Since final scores for the comment is a weighted sum of weights and scores from each criteria, we compared output from both a linear support vector machine (SVM) as well as a logistic regression. We used 94 'picked' comments and 1,574 'not picked' comments. To compensate for the bias in samples, class weight corresponding to the ratio of samples was used. The average precision score using 5-fold cross validation was 0.13 ± 0.07 with 95 percent confidence interval using the linear SVM classifier, and 0.13 ± 0.08 with the logistic regression classifier. Average recall for SVM was 0.60 ± 0.39, and 0.60 ± 0.43 for logistic regression. High-quality comments are rare and we do not want to miss them, which explains the high recall with low precision. For both models, weights for the prediction were similar. Readability, article relevance, and user pick history were positively correlated with high quality, while the conversational relevance and length of comments were negatively correlated with picks. In the end, we used the SVM model parameter as the presets for default ranking.

Other ranking presents were also generated using heuristic methods. Informed by our interviews with domain experts, we identified the following additional presets that

we thought could be useful according to various commenting situations. The following shows the developed presets:

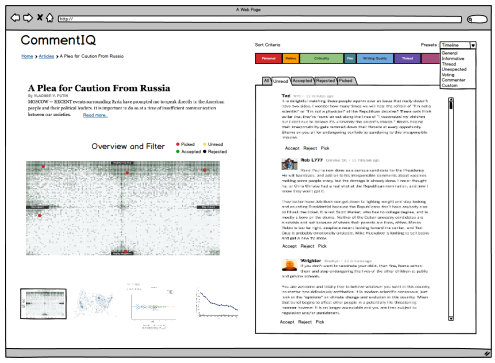
- **Default:** Prioritizes finding generally high-quality comments, trained using the NYT picks dataset as described above.
- **Personal story:** Sorting that favors personal anecdotes; a combination of personal experience and comment length.
- **Unexpected:** A ranking favoring short, unexpected comments. This is a combination of short length and high user reputation, such as average user picks and user recommendation. Therefore, it shows short comments from reliable users at the top.
- **Best user:** This ranking considers only the user reputation to find a comment written by reliable users.

5.5.3 Interaction and Visual Design

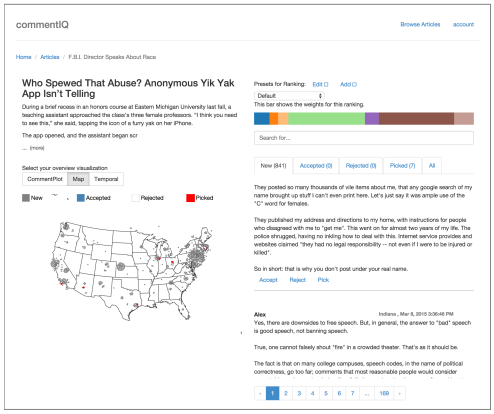
The CommentIQ system is composed of four interface components: article, overview visualization, custom ranking widget, and list of comments. The user can get an overview of comments from the visualization and filter based on making direct selection lassos on the overview visualization (**DR3**). The custom ranking widget provides a way to customize the ranking for one's needs. This section presents the design of these in detail. Figure 5.2 shows an intial sketch, an intermediate mockup, and the final interface. We solicited feedback on intermediate designs by meeting with domain experts before moving

to the implementation phase.

(a) Design sketch



(b) Design Mockup



(c) Developed Prototype

Figure 5.2: (a) Early design sketch during the design iteration, (b) the design mockup, and (c) a working prototype for CommentIQ.

Figure **??** shows the final interface that was designed and evaluated after soliciting feedback on intermediate designs.

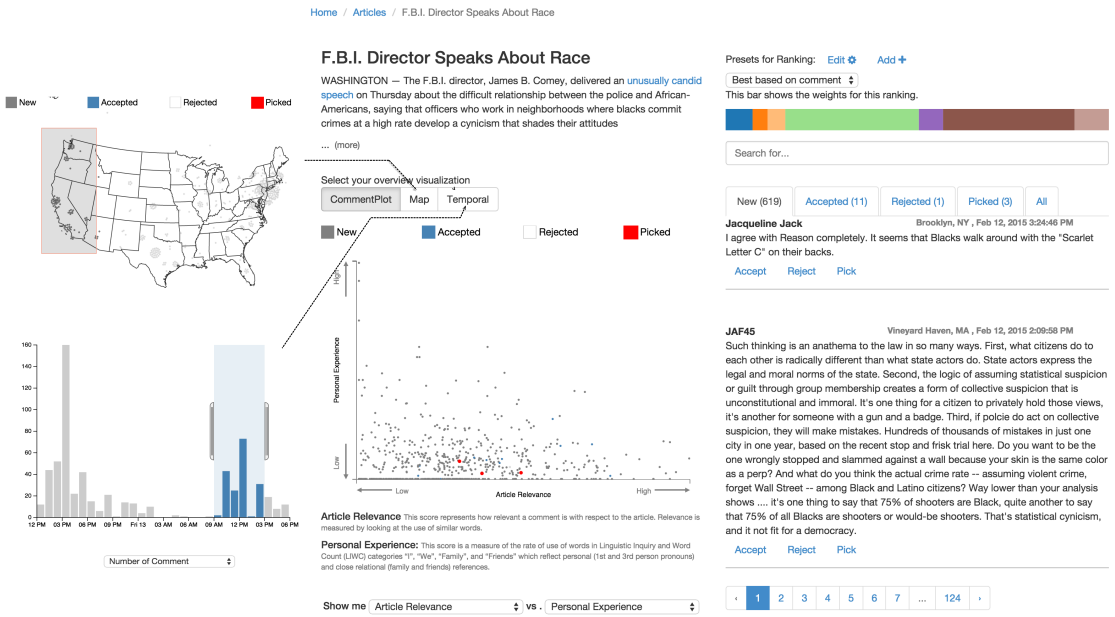


Figure 5.3: The CommentIQ UI showing toggleable visualizations such as scatterplot, map, and timeline (left) that enable overview and filtering of comments, as well as an adjustable ranking based on various weighted quality criteria (right).

5.5.3.1 Customizable Ranking Widget

The goal of the customizable ranking component is to make the custom ranking more intuitive and easy to adjust (**DR1**). The interface is designed for TCMs with various skill levels and goals. At its core is the preset drop-down where the the TCM can quickly select weightings for previously identified scenarios, or create a new weighting preset to meet an emerging need.

Since our presets obviously cannot cover all of the tasks, we provide a customization of presets. The users can change weightings to twelve criteria. For example, by giving weights to recent posting activity of a commenter, and also giving considerable weights to user recommendation score, we can get comments by community members who are very active and have a good reputation.

The different weights are presented with a stacked bar chart. It provides a visual signature for each preset acting as a visual marker. This design is inspired by LineUp, a multi-attribute ranking visualization by Gratzl et al. [120].

5.5.3.2 Overview Widget

The goal of the overview widget is to show a visual overview of comments according to different schemes so that moderators can achieve balanced moderation (**DR3**). It can also act as a visual filter, where people can select interesting subsets of comments. For example, when the user selects along the West Coast in the map view, only the comments from that region are shown in the comments list. Three types of overview are integrated and these can be switched by the user.

- **Map view:** This view shows the location of users. We geocoded the location reported by users as free-text metadata. The MapQuest Open Geocoding API Web Service[6] was used to get the associated latitude and longitude for the given user address. This enables the selection of comments from specific geographical regions to compare viewpoints. Because location is provided by users as free text, the locations are at various granularities (e.g., state, or city). We applied a force-layout algorithm to prevent dots in the same area (e.g. city) from overlapping severely.

- **CommentPlot:** The CommentPlot is a scatterplot of criteria scores for all comments. It was intended to provide a quick selection of comments across certain criteria. The axes of the scatterplots were left vague such as "lower" or "higher"

[6] http://developer.mapquest.com/web/products/open/geocoding-service

in lieu of precise value, because—based on design feedback—we determined that the relative distribution was more important to show than the absolute score on the overview.

- **Temporal view:** This view shows the aggregated scores according to chronological ordering. It can be used for the selection of the comments in a specific time window, as well as seeing change over time. Such changes, e.g., decreasing article relevance and increasing conversation relevance, may be used to make editorial decisions, such as determining when to close the comment functionality on an article.

All CommentIQ views can be brushed and linked. This feature can be used to sculpt queries in many views, i.e., selecting comments from the East Coast area with long length and high personal story score, and from specific times to find an informative personal anecdote for breaking news for that region.

5.5.4 Learning with User Feedback

Even though comments may be selected by editors for any number of reasons, the comment data from NYT is annotated simply as a 'Pick' and does not specify the reason for why it was picked. Comments could be picked based on multiple qualifications, such as the presence of personal anecdotes, informative content, or a short and unexpected viewpoint, for example. Furthermore, it is also very context-specific; in some contexts, one feature may have a positive correlation with quality, while for others that same feature would have a negative correlation. For example, a comment might be selected either for being short and unexpected, or instead for being long and informative. The length of

comments for the first case will be shorter than that of the second one. Thus, it is difficult for the classifier to generalize over the length feature.

To address this limitation of the data we have available, we designed CommentIQ to have a feedback loop (**DR4**). When a moderator designates a comment as a high-quality one, they are prompted for information about *why* it was selected. Along with some predefined options such as "well-written", "informative", "personal experience", "critical", and "humorous" derived from our literature review [91], the moderator can also provide additional free text reasons for selection. When operating at scale, the intent is that we can build a detailed taxonomy of comment selections and correlate the scores and features of those selections to different commenting contexts (e.g. breaking news, different topics or niches). The additional burden of tagging is not too severe because, according to our interviews, only about 1 in 20 comments might be selected.

Another way that we gather feedback from user decisions in CommentIQ is that when a user makes a new custom preset, it is recorded and stored. The recording reveals what the moderators are looking for in terms of relative weights of each of the specified criteria, and can provide data for future research by elucidating compositions of features and feature weights that match given use contexts. By analyzing the preset usage from multiple users, we anticipate deriving a task taxonomy for comment moderation in the future.

5.6 Stage III: Prototyping and Implementation

The CommentIQ system comprises backend components for computing comment scores, as well as front-end components for visualizing and presenting information to users. On the backend we compute scores such as article relevance, conversational relevance, personal experience, and readability [91, 110]. The vocabulary used for the relevance calculations is based on a sample of 3 months worth of comments from the NYT Community API. The backend is flexible and can recompute scores for comments as they become available on an article, or if a comment changes the scores can be updated.

We have open sourced our code, and thus allow others to install and run the comment scoring code on their own servers. The code is available at https://github.com/comp-journalism/commentIQ/tree/master/CommentAPIcode. The frontend of the system shown in Figure 5.2(c) is developed with D3[7] and the AngularJs framework[8]. Firebase[9] was used for hosting and real-time database support to cache articles and scored comments. The CommentIQ frontend is available online at https://commentiq.firebaseapp.com.

5.7 Stage IV: Field Evaluation

We conducted an exploratory evaluation of CommentIQ to gain an understanding of whether and how the tool was helpful to moderators looking to find high quality com-

[7] https://d3js.org/

[8] https://angularjs.org/

[9] https://www.firebase.com/

ments on news articles. We probed generally on the utility they found with the tool and also asked them to accomplish certain tasks to ensure that they had fully explored the functionality of the system. We piloted the study with a data journalist in order to iterate on the interaction design and, in particular, the labeling used in the user interface before we expanded the study to our target domain users of comment moderators. Based on this pilot feedback, we adjusted criteria labels (e.g., "brevity" became "length"), and adjusted the weighting interface to allow for negative weights.

5.7.1 Evaluation Design

We wanted to evaluate whether our approach could be used to improve current processes. Also, since this is design study, we wanted to know what components may have still been missing in the system, which can inform future research. Specifically, we set the following as evaluation goals:

- **Criteria:** Whether criteria and the meaning of weights were easy to understand, if they could be used to help in different editorial contexts, and what new criteria could be helpful;
- **Presets:** Was the goal and utility of presets clear to users, does the algorithm produce results as expected by users, and what might be useful presets to add;
- **Rank tuning:** Were users able to create their own custom ranking for their own goals; and
- **Overview:** How does the overview and filter approach change the moderation pro-

cess and what else might be an interesting aspect to show visually.

To assess these goals we conducted an in-field evaluation of CommentIQ with domain experts. The following reports our procedure and findings.

5.7.1.1 Procedure

The study was conducted on a laptop at the place of work of each participant, usually in an area adjacent to the newsroom. After consenting to be in the study, the CommentIQ interface was demonstrated to the participant. All of the features of the interface were explained, such as how to use and filter on the overview visualizations, how to adjust weights or use the presets on the customized ranking, and the meaning of the various criteria used to score comments. Then the participants were given a 5-minute freeform exploration where they could become acquainted with the system and ask questions if something was not clear. During the session we asked them to speak aloud what they were thinking as they used the system. Then we asked them to conduct specific tasks using the CommentIQ system. The sessions were audio recorded and transcribed for analysis. The specific tasks included:

- **Use of one criteria:** Select one or two criteria that are interesting. Using only that criteria, read the top and bottom comments and compare them. Why did you select that criteria? Are the results what you expected?;
- **Use of multiple criteria:** The participants were asked to create custom rankings using multiple criteria. We asked why they selected criteria and weights and whether the result was working as they expected;

- **Use of Commentplot:** We asked them to select two criteria for scatterplots and select regions of the visualizations that might be interesting;
- **Use of Map:** Find if there are any difference in perspectives according to geographical locations presented on the map;
- **Use of Temporal view:** Find meaningful patterns in the chronological dimension;
- **Picking high-quality comments:** The user was asked to set a goal for target comments that they were interested to find and then they used CommentIQ to find those comments;

After these tasks, we finished the session by asking for the user's general impression and opinion about the advantages and disadvantages of using system.

5.7.1.2 Content

We made use of three datasets for our evaluation: one for demo purposes, and two for the journalists to interact with. We selected news articles from a range of topics and with varying numbers of comments. One article, "What Is the Next 'Next Silicon Valley'?"[10] (147 comments), was selected because we thought that it would elicit comments from different geographies that might be interesting for the moderators to consider on the map visualization. Another article, "Who Spewed That Abuse? Anonymous Yik Yak

[10] http://www.nytimes.com/2015/03/05/upshot/what-is-the-next-next-silicon-valley.html

App Isn't Telling"[11] (848 comments), was chosen because we thought that it would elicit comments that included personal stories or perspectives that might be surfaced by our ranking, and it was a much larger and more challenging number of comments. Finally, we chose "F.B.I. Director Speaks About Race" (634 comments)[12] as a demo, since race relations are a hot topic in the U.S. and we thought that comments here might reflect different perspectives that the moderators would want to highlight.

5.7.1.3 Participants

We evaluated CommentIQ with working professional journalists who have direct responsibilities for comment moderation as part of their daily duties. Such professionals have knowledge of the real challenges in comment moderation, and the workflows and editorial criteria associated with evaluating online news comments. Combined with the in situ setting in which the study took place, this allows for more ecological validity to reflect on how the tool may be useful in practice. We recruited participants by soliciting industry contacts and asking for referrals. In all, we recruited seven participants (five male, two female) from local (Baltimore Sun) and topical (Wall Street Journal) as well as national (New York Times, Washington Post) outlets. The moderation workflows of the different outlets provided some variability and diversity for the evaluation as well. Only New York Times currently uses a pre-moderation workflow and actively promotes

[11] http://www.nytimes.com/2015/03/09/technology/popular-yik-yak-app-confers-anonymity-and-deliv
html

[12] http://www.nytimes.com/2015/02/13/us/politics/fbi-director-comey-speaks-frankly-about-police
html

high-quality comments with *NYT Picks*.

Our participants come from some of the most respected news outlets in the U.S. and are leading-edge practitioners. Their titles and roles include community moderators, community manager, a director of audience engagement development, and a director of digital news projects. Two of the participants are contributors to the Coral Project [13], which tries to improve communities on news sites through the development of an open source software platform. Table 5.1 shows affiliations and related work experience. We will refer to them as 'P1' or 'P2' when citing their comments in the rest of the paper.

Table 5.1: Experience and affiliations of in-field evaluation participants.

ID	**Organization**	**Field experience** (in years)	**Workflow**
P1	Washington Post	10	Post-moderation
P2		1	
P3		4	
P4	New York Times	4	Pre-moderation
P5		7	
P6	Wall Street Journal	4	Post-moderation
P7	Baltimore Sun	7	Post-moderation

5.7.2 Findings

Below we discuss the findings from our evaluation.

[13] http://coralproject.net/

5.7.2.1 General Utility

In general, participants were positive about the approach and capabilities of CommentIQ. P4 stated that CommentIQ is a great improvement to his existing interface. P3 stated that CommentIQ is a much more sophisticated and powerful tool than anything she has ever used for comment moderation. The ability to find personal anecdotes quickly and to select comments based on geographic regions received especially positive responses. This reaction was observed from outlets with both pre-moderation and post- moderation processes. Currently, most outlets are just filtering bad comments, but this is due to resource constraints rather than any resistance against instead selecting high-quality ones. Participants anticipated that CommentIQ would enable them to find high-quality comments, which is currently not supported by tools that assist in removing low quality comments.

5.7.2.2 Criteria

The participants were able to understand the meaning of each criteria and use multiple criteria to find interesting comments for their goals. The following quote from P6 shows how moderators think in terms of qualification for good comments and the effective use of custom rankings and the overview visualization to identify worthy comments:

> So I will look for someone who had used Yik Yak so that would be personal experience ... and I guess something that is relevant to the article. And that does not seem very relevant. But we will see what's going on here [selecting region on visualization]. So these are okay. So I want to find something

> more about how people use it. No, this is just old people being mad. What's going on here... Yes. This is a great comment you need. Talking about how she became aware of this app and it was relating to her sons at high school and this is perfect. This is exactly the thing we are looking for.

Participants were especially interested in *personal stories* (P1, P3, P4, P5, P6, P7). Initial use of the *personal experience* score created problems because it counts the frequency of words such as 'I' or 'we', resulting in short one line sentences with 'I' ranked higher. However, participants also reflected on the use of various strategies to remedy this by themselves, such as selecting comments of a certain length from the CommentPlot (P1, P6) or giving weights to the 'Length' criteria in sorting weights (P5).

P5 was able to successfully create his own ranking for personal stories. He gave high weight to personal experience criteria of comments, and at the same time gave weighting to the user picks and recommendations score to select good people. P3 was also able to find good comments for reporters to follow-up using a combination of map and ranking based on conversational relevance and user reputation.

The readability score of comments received mixed reviews. Many participants suggested readability was an important measure, but the result of the current algorithm were found to be confusing (P1, P4, P6). The readability was computed as the SMOG index [119], which is a measure of the grade-level difficulty of the text. However, the score requires further refinement to represent general readability in editorial contexts. As more general readability features, P1 and P4 suggested the proper use of paragraphs in a comment would make long comments more readable. This is because sometimes, bad

comments list meaningless sophisticated language without any hierarchy, resulting in a dense block of text. P6 suggested similar criteria, such as use of 'long ellipsis' or weird punctuations and spacing as a possible criteria for evaluation.

Article relevance was used frequently as a qualification. P1 made an interesting argument that a lower article relevance might also at times be interesting because it is not repeating the article but offering fresh view points. In a similar way, conversational relevance might not reflect the status of threading, if similar vocabulary is not used in conversational responses. P1 suggested the use of threading information, such as people replying to each other for the calculation of conversational relevance.

User activity history, such as average *user picks* or average *user recommendations*, were frequently combined with comment-based features as a strategy to find reliable people (P5, P6). Also there were requests for mapping more of user history, such as rejection history (P3), flagged history, or banned history (P6). P7 suggested the ability to pull out all the comments by the particular user, when he saw a very thoughtful user, so that he could use her comments for other articles. P1 suggested that a recent activity score might be useful to find users who are either new to the community or well-known members. P7 felt it could be useful to be able to filter out the ten to twenty people who are trolling by commenting on everything.

5.7.2.3 Overview Visualization

The map view received a lot of positive feedback as a key benefit of the system. Many usage scenarios for the use of the map feature were suggested. For example, P1

could find interesting comments from a specific region using the map view. P4 suggested its use to find personal stories, where you can find the comments from the geographic area where the news is coming from. P6 suggested that the map view could be useful for sport articles, where people are often passionate about their home or school team. Participants compared opinions of Silicon Valley with Florida in the article about Florida being the next Silicon Valley (P1). Also, people tried to compare conservative states with progressive states, using geography as a proxy for political perspective (P7). It was suggested that the granularity of the map might be made flexible to suit different demands: the local outlet wanted a more local map, while national outlets also wanted a global map (P3, P5, P6), with mechanisms to quickly select sub-regions (e.g., a state) using a single click.

In the tasks that used CommentPlots, people could select different criteria to get an insight from the scatterplots. Many people used *personal experience* (P4, P5) and *readability* (P4) to find interesting comments. P1 successfully used highly relevant but short comments for reporters to quickly find a quote to dress up their story with reactions from users. This shows the adaptability of our approach of leveraging various qualifications to accommodate different usages, which was one of goals of CommentIQ.

Elaborate usage scenarios for temporal view were also suggested. P5 suggested its use for breaking news items, where as new follow-up stories are uncovering, the ability to find comments from a certain time window can be useful to detect the change of tones and information. Because people do not want to read comments about out-dated news, TCMs can put their limited resources on more recent ones. P1 suggested the use of temporal information for getting changing responses of readers as sports events evolves.

For example, when there is a new score for an ongoing football game, the sentiment of readers might change accordingly. TCMs can use it to get the comments showing an exciting moment of reversal. P6 suggested that the temporal view can be useful when multiple TCMs are working together in coordination. P7 articulated that the first wave of comments will contain more personal anecdotes of the event and the last wave of comment will contain more diverse viewpoints about the issues.

5.7.2.4 Additional Use Cases

Many expressed that the current tools for comment moderation, which are usually based on chronological or recommendation-based scores, are sub-optimal (P6, P5). Some features of CommentIQ, such as the map-based view and the sorting presets feature, were suggested to even have potential for exposing to the readers or reporters themselves (P3, P5). P3 stated that, as a community manager, she wants readers to stick around, but the current comment list they use might scare them off. Given the ability to sort comments according to quality might make them stay longer. P5 articulated that he could program some presets for reporters so that reporters can use the intelligent sorts to find interesting content that may inform them. Selecting specific locations from comments is especially useful for reporters for finding potential sources of new information. P6 suggested that the map view can be useful for the general public, especially for sports articles.

5.8 Discussion and Future Work

Our evaluation of CommentIQ along the dimensions of scoring criteria, weighting presets, easy tuning, and overview visualizations showed that the visual analytic system that we designed was productive and useful for comment moderators. Users were able to effectively understand the criteria provided, compose them into conjunctions of scores that were editorially meaningful, and tune the results to provide access to higher quality comments of editorial interest. The visualizations themselves proved quite useful for providing an overview of the comment score space, and helped orient users towards comments according to geography, temporal characteristics, and score distributions. Our evaluation thus provides support for the design rationales DR1, DR2, and DR3. At the same time, assessing DR4 requires a much longer-term deployment to study whether the approach can yield a large enough quantity of feedback data to train better models, and is left for future work. Resource constraints and organizational barriers currently limit our ability to integrate CommentIQ fully with real newsroom data flows and content management systems, thus preventing a full at-scale deployment. However, we hope that the CommentIQ design can inspire future work and development in exploring the potential of user feedback data.

CommentIQ supports a transformational change to the moderation process. One of the more surprising results from our evaluation was the extent to which comment moderators were ready to begin thinking of moderation not as a policing function, but as a first-class editorial position in the newsroom. As P5 explained, CommentIQ positively changed the moderation workflow by "shifting moderating to a reporting research job"

It changes the role of moderators to editorial knowledge work, because they now think in terms of what is the qualifications for comments they are looking to publish or use. CommentIQ allows moderators to set up hypotheses and run experiments on presets for the workflows that work well in different contexts and for different types of journalism. Moderators can then publish or share that knowledge with others in the form of custom ranking presets, either internally with other moderators or reporters, or indeed ultimately also with their readers.

By framing comment selection as something that is done to identify interesting content to be published, our users articulated new use cases for comments. For instance, participants wanted to understand more about the people who had written various comments, including aspects such as profession or background knowledge of the commenter. Treating the comments as content, and the commenters as potential journalistic sources, opens new possibilities for leveraging comments in the news process. Deriving user models based on past commenting behavior and content would allow journalists to tap into comments in new ways, and is an exciting area for future work in analytics.

Our visual, interaction, and algorithmic design was entirely user-centered, and, as a result, none of our final design choices in these areas were particularly surprising or complex in nature. In particular, all of our visualizations were straightforward scatterplots, time-series plots, or map visualizations. Since our contribution is on the design study, domain characterization, and in-field evaluation aspects of the CommentIQ system, we do not think this is a problem. Rather, it is an indication that high visualization complexity has little intrinsic value, but needs to be matched to the needs and capabilities of the intended users.

The overarching take-away from our design study of CommentIQ is that journalists (and journalism as a domain) demand analytic solutions that place humans squarely in the sensemaking loop with the analytics. Visual analytics is thus an extremely well-suited approach for design in this domain. Journalists do not want editorial decisions made *for* them, but rather seek designs that enable and enhance their own decision-making functions so that they can adapt to new situations and contexts and apply human judgment to editorial decisions. Flexibility was seen as extremely important, as well as not having the analytics make strong classification decisions automatically.

Future work in this domain should be oriented towards developing new and better scores, including for dimensions such as novelty, criticality, thoughtfulness, and others [91]. Much additional work on natural language processing and analytics needs to be done to develop such metrics so that they are understandable and useful to moderators. Based on our initial interviews, we identified the desire to maintain diversity and balance in the selected comments. While CommentIQ does provide this to some extent, moderators in our evaluation were looking for minority opinions in terms of sentiment, political thought, or even religious affiliation. Even though our system can provide limited access to this, such as by proxying political thought to a map view, the current CommentIQ system lacks the ability to show these dimensions in terms of semantic analysis such as sentiment or political position. Thus, future work should also strive to develop advanced analytics for dimensions such as political affiliation so that this can be represented explicitly in the overview visualizations.

Given that there were at least small teams and in one case a quite large team of moderators in the organizations where we tested the system, another avenue for future research

is how to explicitly support collaboration in the comment moderation process. Because CommentIQ uses a real-time database, the system already supports multiple users moderating a comment database together, but we think there are additional mechanisms that could streamline and coordinate a collaborative moderation process.

5.9 Conclusion

We have described a design study aimed at supporting community managers for online news sites in filtering reader-submitted comments to news articles using a combination of visual interfaces and automatic algorithms. Our study spanned a significant time period during which we explored the domain, design, implementation, and deployment of a visual analytics system called CommentIQ for this purpose. Our contributions include the domain characterization in comment moderation, the CommentIQ system, and results from an in-field evaluation featuring working journalists from major newspapers.

Chapter 6: ConceptVector: Building a semantic axis

Here we propose a method to create a semantic axis by building a dictionary for the custom concept and applying it to create a numeric representation of the text for the concept.

Central to many text analysis methods is the notion of a *concept*: a set of semantically related keywords characterizing a specific object, phenomenon, or theme. Advances in word embedding allow building a concept from a small set of seed terms. However, naive application of such techniques may result in false positive errors because of the polysemy of natural language. To mitigate this problem, we present a visual analytics system called ConceptVector that guides a user in building such concepts and then using them to analyze documents. Document-analysis case studies with real-world datasets demonstrate the fine-grained analysis provided by ConceptVector. To support the elaborate modeling of concepts, we introduce a bipolar concept model and support for specifying irrelevant words. We validate the interactive lexicon building interface by a user study and expert reviews. Quantitative evaluation shows that the bipolar lexicon generated with our methods is comparable to human-generated ones.

6.1 Introduction

We live in a society that routinely produces more textual data on a daily basis than can be comfortably viewed—let alone analyzed—by a single person in virtually any given domain: finance, journalism, medicine, politics, and business, to name just a few examples. As a result, automatic text analysis methods, such as sentiment analysis, document summarization, and probabilistic topic modeling, are becoming increasingly important [121]. Central in virtually all such methods is the focus on textual *concepts*, defined as a set of semantically related keywords describing a specific object, phenomenon, or theme.[1] The benefit of this unified view is that concepts can be created once and then be shared and reused many times. However, constructing high-quality concepts for the purpose of classifying and organizing document collections is a challenging task that cannot be easily automated.

Motivated by this challenge, we present a visual analytics system called CONCEPTVECTOR that guides users in interactively building such textual concepts in a highly efficient and flexible manner and then applying them to document corpora. The ConceptVector model includes both unipolar (e.g., crude oil, immigration, genetics) and bipolar concepts (e.g., positive or negative reviews, liberal versus conservative politics, Trekkie versus Star War fans, etc). Furthermore, the model lets users select specific keywords to be irrelevant to their concept, which will further improve the quality of the concept. Finally, concepts created in this manner can then be applied to a document corpus

[1] More formally, a textual *concept* is a set of keywords associated with the degree of their corresponding relevance to the concept.

to analyze its characteristics, such as analyzing product reviews based on sentiment, blog posts based on political affiliation, or trade articles based on business sector. We demonstrate this technique with a web-based text analysis system for analyzing comments for the New York Times articles available at http://conceptvector.org.

To our knowledge, the ConceptVector system is one of the first visual analytics systems applying the state-of-the-art word embedding techniques [122], which made recent breakthroughs in many tasks in traditional natural language processing and text mining, to practically useful applications in a user-driven interactive environment. The specific contributions of our work include the following:

- The novel ranking algorithm that generates a concept relevance score given a user-specified set of seed words;
- A visual interface where users can interactively generate, refine, and share specifications for custom concepts, and then use these concepts to analyze document corpora;
- Results from quantitative experiments showing that our algorithm can generate rankings of words that are highly correlated with human-generated ones;
- Results from a user study evaluating the performance of participants generating concepts using our visual interface compared to using WordNet or a simple thesaurus; and
- A visual analytics system called CONCEPTVECTOR that demonstrates our novel process of text analysis in the domain of comments analysis in news articles.

6.2 Related Work

Numerous previous studies have attempted to scale up human capability to make sense of a text corpora. ConceptVector is a visual analytics system that uses word-level semantics using a lexicon for concepts. In this section, we discuss current research related to our work from three perspectives: (1) manual approaches for constructing word relationships and hierarchies, (2) automatic word-embedding approaches, and (3) visual analytics approaches for word-level content analysis.

6.2.1 Building Word Relationships and Hierarchies

Manually building a lexicon with coherent semantics has long been an active area of research. LIWC [123] is an example of a manually built lexicon that characterizes various concepts. The General Inquirer[2] is a comparable line of research that builds lexica in diverse concepts. Beyond building a lexicon for a particular purpose, researchers have also developed sophisticated structures that store relationships and hierarchies of words.

Unlike these methods, which rely on a small number of experts to compose a lexicon, the Hedonometer project [124] employed crowdsourcing to build a lexicon for sentiment ranking. One benefit of this approach is its large-sized lexicon, containing the ranked list of 7,000 words in terms of the degree of happiness.

Although these manually built databases, which store relationships and hierarchies of words, provide high-quality information for various natural language understanding and text analysis tasks, the main problem is the significant human effort needed to create

[2] http://www.wjh.harvard.edu/~inquirer/

and validate them. This makes it difficult for users to efficiently create a lexicon for their own purpose. Because of this high cost, only a limited number of widely applicable concepts can be built, and building a domain-specific custom lexicon has not been well-supported. This has motivated a slew of automatic methods to craft a lexicon of custom concepts.

6.2.2 Word Embedding

Word embedding computes semantically meaningful vector representations of words in a high-dimensional space. Compared to traditional methods of representing a word as a vector, such as the bag-of-words representation [125] or latent semantic indexing [126], recent word embedding methods such as word2vec [122] and GloVe [127] have two noteworthy advantages in terms of high-level semantics: meaningful nearest neighbors and linear substructures [127]. Regarding the first, these techniques satisfactorily capture semantically related words as the nearest neighbors of a particular word in a vector space. As for linear substructures, the vector obtained by subtracting two words in a vector space often yields semantics that contrast the words. For instance, if we subtract a word vector 'queen' from 'king' and then add 'girl,' the resulting vector corresponds to 'boy.' This stems from the fact that the vector from 'king' to 'queen' and from 'boy' to 'girl' are similar, commonly representing the notion of gender (from male to female).

Since such word embedding techniques have shown their advantages in numerous tasks in natural language processing and information retrieval, advanced word embedding techniques have recently been actively studied. Ling et al. proposed the use of mul-

tidimensional transformation matrices to flexibly capture different semantics of a single word [128] leading to better representations for part-of-speech tagging tasks. Similarly, assigning more weight to a particular word than other words in a sentence produced better word embeddings by extending the continuous bag-of-words model [129]. The weights are computed by an attention model, yielding better performance than neural network models [130]. Tian et al. integrated an expectation-maximization (EM) algorithm with the continuous skip-gram model to handle the polysemy problem [131]. For example, the word 'bank' can have multiple vector representations corresponding to 'a place related to money' and 'a place where water runs,' respectively. Besides transforming word-level embeddings, several efforts extended this technique to document-level embeddings that yielded good performance in information retrieval tasks [132, 133]. Other notable recent studies applied the technique to machine translation [134, 135]. Additionally, the skip-gram idea of word2vec has been applied in generating the embeddings of entities in other domains, e.g., bibliographic items in scientific literature [136] and nodes in network analysis [137]. Finally, and most relevant to this work, Fast et al. [138] showed that word embedding can be used to expedite lexicon-building so that users can easily create their own concepts.

6.2.3 Word-Level Content Analysis

The use of a coherent set of keywords for characterizing a particular concept has wide applicability in various document analysis tasks. For instance, the problem of sentiment analysis has been tackled by identifying a set of keywords expressing the positive (or

the negative) sentiment, possibly with different degree values, and this is also known as a lexicon-based sentiment analysis [139, 140]. In topic modeling, such as latent Dirichlet allocation (LDA) [50], a topic represents a set of semantically related keywords found in a document corpus, e.g., sports- or science-related topics, generated from a large amount of news articles. Recent studies by Kim et. al [141, 142] are particularly notable because they introduced a continuous embedding space similar to *concepts* as considered in this paper, although they only covered emotion-related concepts.

Topic modeling has also been actively employed in visual analytics approaches for document analysis. TIARA [143] is one of the first systems that integrated LDA with interactive visualization. This system visualizes the topical changes of documents over time in a streamgraph view reminiscent of ThemeRiver [84]. Other studies, such as ParallelTopics [144] and TextFlow [145], also focused on visualizing topical changes over time in document data by using different visualization techniques, such as parallel coordinates and custom glyphs, respectively. In most of these studies, the key information for understanding the visualized topics is a set of dominant keywords associated with each topic. However, the number of topics can be as large as several hundreds or thousands [146]. This makes manual interpretation of topic characterization or topic labeling a main bottleneck for the topic modeling. To facilitate this task, Termite [147] provides an interactive visualization with which a user can explore topics in terms of their dominant keywords, as well as the overlapping patterns of keywords among different topics. In addition, various interactive capabilities that can steer the topic modeling process in a user-driven manner have been studied. iVisClustering [148] allows a user to perform a user-driven topic modeling process by interactively constructing topic hierarchies and

changing keyword weights of a topic. Chang et al. introduced a interactive clustering system based on knowledge-graph embeddings [149]. More recently, non-negative matrix factorization [150] has been proposed as an alternative topic modeling method that can flexibly support user needs such as splitting and merging topics, creating a new topic via particular keywords, and supporting user-driven topic discovery [151].

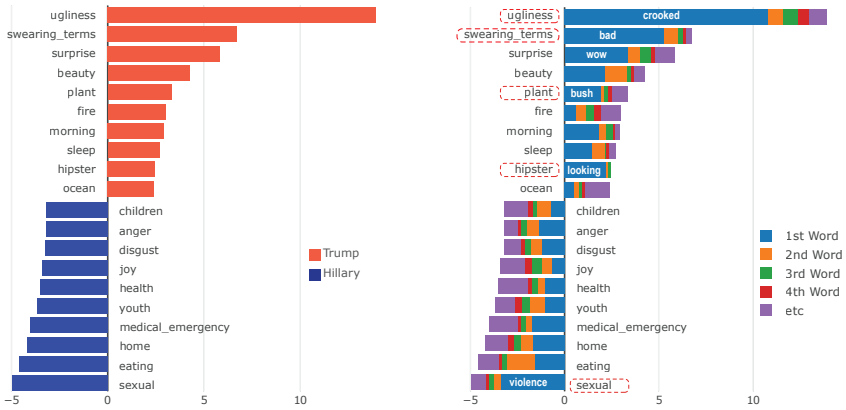
Our ConceptVector work in this study has much in common with topic modeling: both try to summarize documents, and both express words and documents as high-dimensional vectors. However, they differ in whether humans or the document corpus itself drive the latent semantics behind each dimension. Topic modeling, therefore, is better-suited for finding hidden underlying topic clusters, while ConceptVector provides better interpretability and transferability. In this sense, topic modeling and ConceptVector are complementary.

Lexicon-based document analysis has also been applied in various application domains. For instance, Kwon et al. [152] used a manually built lexicon to identify online health community postings that share personal medical experiences. In most of these previous studies, document analysis relied on lexicons of properly chosen words that were created for a specific purpose. The ConceptVector system aims to help users easily create such lexicons.

6.3 Motivation: Concept-Based Document Analysis

Here we describe two real-world examples where concept-based document analysis was performed by using Empath and Jupyter Notebook.[3] First, we show how concepts can reveal the underlying differences in two document sets, such as tweets from Hillary Clinton and from Donald Trump, highlighting the importance of integrating the lexicon-building process with its refinement during the document analysis. Second, we demonstrate how NASDAQ 100 companies can be clustered using the differences in concepts and how each cluster can be interpreted using tweets mentioning them.

6.3.1 Tweets by U.S. 2016 Presidential Candidates



(a) Top 10 categories between Trump and Hillary

(b) Detailed analysis of top words in each category

Figure 6.1: Comparison of tweet messages from Hillary Clinton and from Donald Trump during the U.S. 2016 presidential election. The odd ratios of the top 10 categories show differences between the two candidates in (a). The analysis on actual keywords contributing to their corresponding category scores reveals limitations of using the prebuilt lexicon in (b). Red dotted categories do not make sense, because an irrelevant top word is counted dominantly. For example, keywords such as 'bush' in the 'plant' category and 'looking' in the 'hipster' category are not relevant to their categories.

[3] http://conceptvector.org/#/twitter

Empath [138] provides prebuilt lexica of various concepts that can be used to compare two document groups. Using these 194 prebuilt concepts provided by Empath, we analyzed two sets of tweets composed by Hillary Clinton and Donald Trump[4] respectively, each of which contains about 3,000 tweets. Figure 6.1(a) shows the top ten categories statistically significantly different from each other ($p < .01$). For example, Trump mentioned more terms in the 'ugliness' (13.9 odds), 'swearing terms' (6.7 odds), and 'surprise' (5.8 odds) categories, whereas Hillary used more in the 'sexual' (4.97 odds), 'eating' (4.6 odds), and 'home' (4.2 odds) categories. Interestingly, Trump used more casual language while Hillary's tweets contained words related to 'anger' and 'disgust.'[5]

However, further examination reveals numerous false positives. Figure 6.1(b) shows the most dominant keywords corresponding to each concept. While some keywords make sense, e.g., 'wow' in the 'surprise' category, less meaningful words exist in other categories. For example, Trump was shown to talk more about the 'plant' concept because of the term 'bush,' which in fact indicates Jeff Bush. 'crooked' in the 'ugliness' concept means 'deformed', whereas Trump is using it in his catchphrase 'Crooked Hilary' to mean 'not straightforward; dishonest.' Besides, another strong concept 'hipster' emerged because of the use of the term 'looking,' while 'swearing terms' emerged because of the use of 'bad.' In Hillary's case, the 'sexual' concept appeared owing to the use of 'violence,' which did not make much sense. After removing these words from the corresponding concepts, these concepts no longer show significant differences between the two.

[4] https://www.kaggle.com/benhamner/clinton-trump-tweets

[5] http://graphics.wsj.com/clinton-trump-twitter/

6.3.2 Tweets from NASDAQ 100 Companies

Concepts can be also used to extract meaningful features from documents. Given tweets about NASDAQ 100 companies,[6] our goal in this work was to find meaningful clusters and their distinct characteristics by using concepts as features. That is, for a set of tweets belonging to each company, we obtained its 194-dimensional feature vector by computing the occurrence count of words contained in each of the 194 prebuilt concepts. Afterwards, we performed *k*-means clustering and 2D embedding via principal component analysis (PCA) [153].

The results (Figure 6.2(a)) reveal that words from the company name affect the results, e.g., 'cooking' and 'restaurant' categories for Dish Network Corporation. Companies containing 'technology' in their names form a cluster because of similar reasons. After removing these words from the lexicon of the corresponding concept and recomputing feature vectors, the clustering results are shown to be more reasonable (Figure 6.2(b)). For example, Marriott and TripAdvisor form a single cluster owing to the high frequency of words in 'tourism,' 'warmth,' 'sleep,' and 'vacation' mainly because of the use of the words 'hotel' and 'hot.' Companies with their tweets containing negative sentiments such as 'ridicules,' 'neglect,' 'kill,' or 'hate' are clustered together.

This example shows that document analysis using concepts as a feature extractor is useful, but that existing systems such as Empath lack the integrated support for concept construction and refinement, as well as interactive concept-based analysis itself.

[6] http://www.followthehashtag.com/datasets/nasdaq-100-companies-free-twitter-dataset/

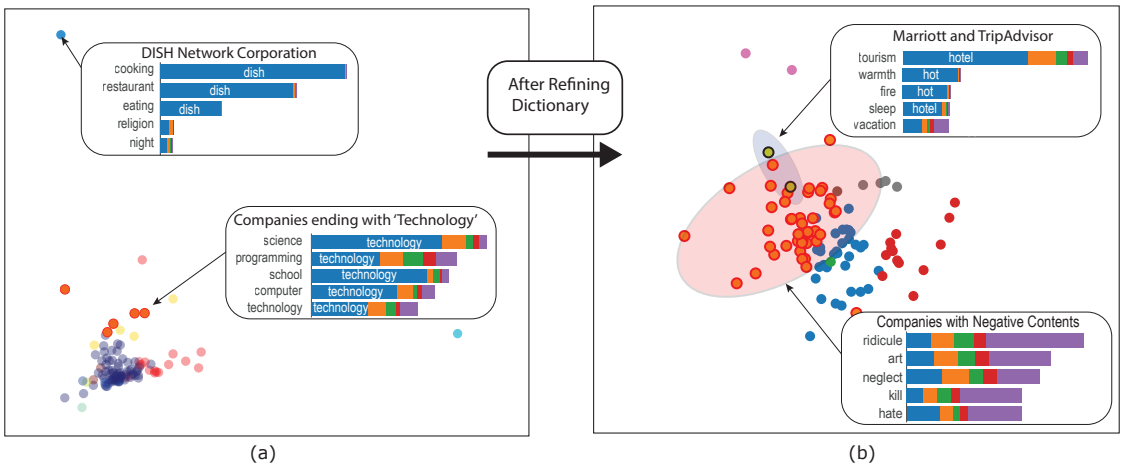


Figure 6.2: PCA 2D projection of NASDAQ 100 companies with their k-means clustering labels color-coded, where the feature vector of each company is computed from its tweets' word count in each of 194 concepts. The clustering using the prebuilt lexica shows some outliers (a), where further investigation of contributing words shows that the company name itself acts trivially as strong signals, such as 'dish' in Dish Network Corporation. Another cluster is shown to be formed because of the common word 'technology' in their names. After excluding them in the initial lexicon, more meaningful clusters are revealed. For example, Marriott and TripAdvisor form a cluster because of words in 'tourism,' 'vacation,' and 'sleep' concepts (olive green with a black border). Companies with negative sentiments such as 'ridicules,' 'neglect,' 'kill,' and 'hate' were also clustered together (bright red dots with red border).

6.4 ConceptVector in Action

To address the limitations of using prebuilt lexica, ConceptVector aims at facilitating user-driven concept building as well as the subsequent concept-based document analysis in a seamless manner.

While the previous examples started with prebuilt lexica, we now present how ConceptVector can be used to build custom concepts in the task of journalistic curation of user comments on online news. Moderation of online comments can follow various approaches, and often includes mechanisms to remove uncivil, profane, or otherwise inflammatory comments. That is, however, not our focus here; instead we consider the approach

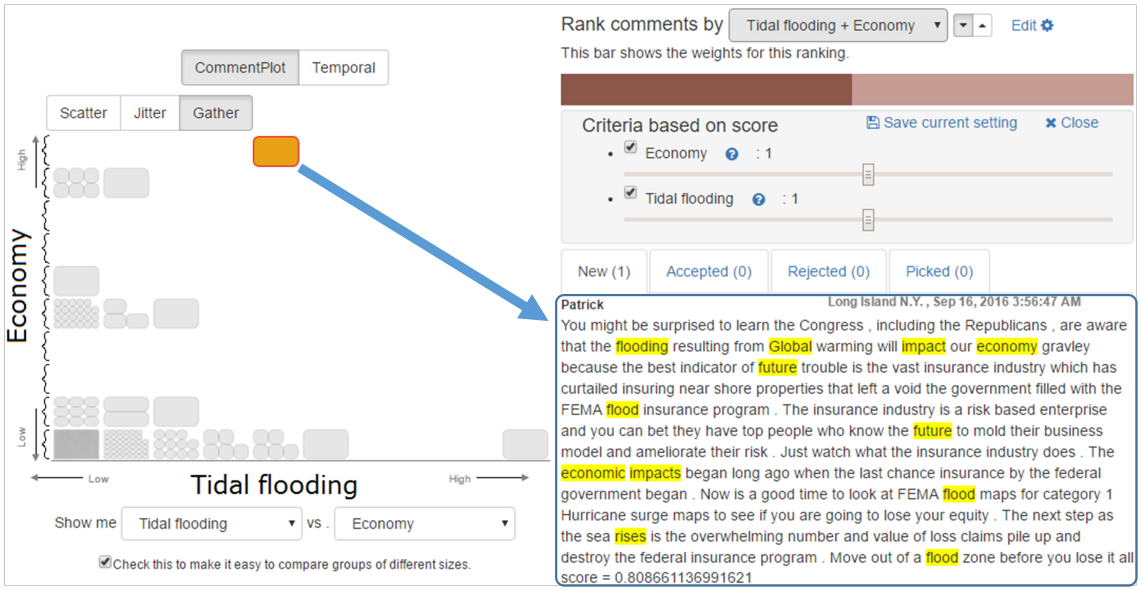


Figure 6.3: Distribution of comments across the 'tidal flooding' (X-axis) and the 'economy' (Y-axis) concepts. A comment that has scored relatively high on both concepts is selected (orange box). The content of the corresponding comment within this dataset is shown.

championed by the New York Times, in which editorially interesting and insightful comments are selected and highlighted on the site as "NYT Picks" comments. Below we present a scenario showing how an expert community moderator from an organization such as the New York Times could leverage the capabilities of ConceptVector to define and deploy those concepts useful for finding and selecting "NYT Picks" comments.

It is helpful to understand the general editorial attitude and approach—the persona—of an online news moderator. Prior research has enumerated several dimensions of editorial interest for finding high-quality comments including factors such as comment relevance, argument quality, novelty, and personal experience [91]. Importantly, different articles or subcommunities on a site demand different approaches to moderation and the application of different editorial criteria [13]. Diversity is a dimension of utmost importance to comment moderators; it is a difficult task to select high-quality comments that

also reflect the diversity of voices available in a comment stream. ConceptVector is well-suited to enabling such diverse selection because of its capabilities to allow moderators to develop content-specific or even article-specific concepts to apply to different contexts, and to see how comments are scored when applying that concept.

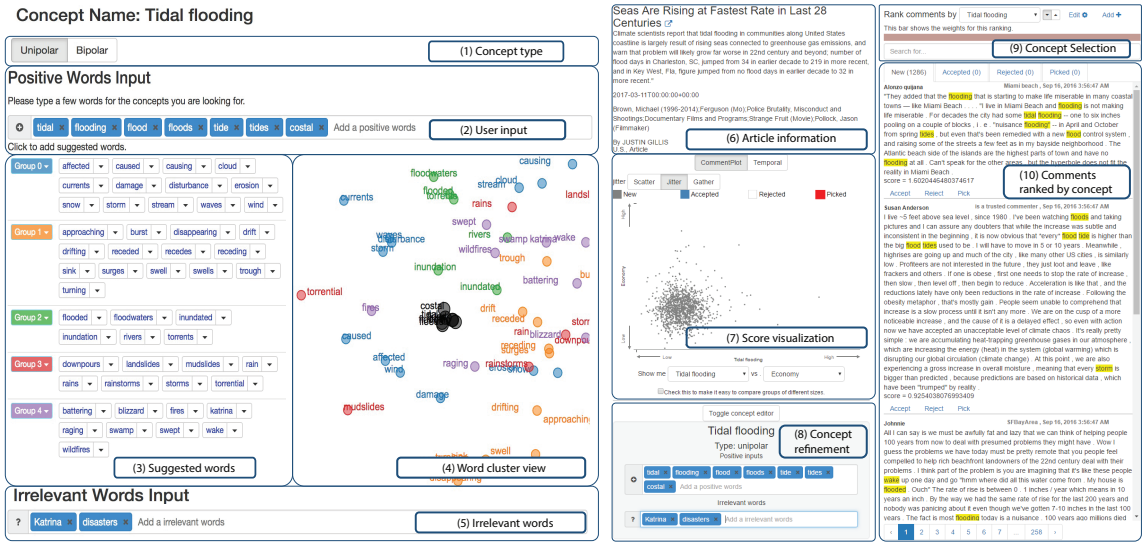
Let us follow Laurie, a hypothetical comment moderator at the New York Times who is trying to moderate comments on several different articles. Her task is to pinpoint diverse but representative comments to highlight on the site as "NYT Picks."

The article she is examining is entitled "Seas Are Rising at Fastest Rate in Last 28 Centuries," which has over 1,200 comments when she logs on.[7] She is really not looking forward to moderating the comments for this article, because an article like this always brings out the global warming skeptics who can cause quite a ruckus. The article is specifically about the idea of 'tidal flooding,' i.e., the notion that coastal areas will be flooded more often as sea levels rise. Using ConceptVector, she first wants to develop a tightly defined concept on this specific idea of 'tidal flooding' so that she can find comments maximally relevant to the article.

Laurie creates a unipolar concept for 'tidal flooding' by typing in its relevant keywords, starting with the words 'tidal' and 'flooding.' She then sees related words as recommendations in the scatterplot that help her flesh out the concept by adding related terms such as 'flood,' 'floods,' 'tide,' and 'tides,' as shown in Figure 6.4.

She examines the clusters of other terms generated, and decides to avoid words related to specific instances of tidal flooding, such as 'katrina,' or those associated with

[7] http://www.nytimes.com/2016/02/23/science/sea-level-rise-global-warming-climate-change.html



(a) Concept-Building View (b) Document Analysis View

Figure 6.4: ConceptVector supports interactive construction of lexicon-based concepts. Here the user creates a new unipolar concept (1) by adding initial keywords related to 'tidal flooding' (2). The system recommends related words along with their semantic groupings (3), also shown in a scatterplot (4), revealing word- and cluster-level relationships. Irrelevant words can be specified to improve recommendation quality (5). Concepts (9) can then be used to rank document corpora (10). Document scores can be visualized in a scatterplot based on concepts such as 'tidal flooding' and 'money' (7). Users can further refine concepts based on results (8).

storms and hurricanes, such as 'storm,' 'raging,' or 'swell.' She wants to keep this a general-purpose concept. Moving on to the second phase, she applies the concept to the comments on the article and immediately notices other key terms, e.g., 'storm,' highlighted as yellow in the retrieved comments. She then adds them to the relevant keyword set of the concept using the integrated concept editor.

Based on her understanding of media framing, Laurie knows that people often discuss complex issues in terms of specific frames relating to definitions, causal interpretations, moral evaluations, and solutions [154], as well as using topical perspectives like economic, political, or scientific. She decides to find a comment to highlight that deals with tidal flooding from the perspective of economic implications. Similar to how she de-

veloped the unipolar concept for 'tidal flooding,' she develops another unipolar concept relating to economic implications. She starts with 'economic,' and the scatterplot of recommended words leads her to add related terms such as 'economy' and 'economies,' as well as some of the negative implications that she wants to include, such as 'crisis,' 'impact,' 'turmoil,' and 'instability.' Her economic concept is thus tuned towards negative economic impacts that could arise.

To apply a combinations of these two concepts, Laurie checks the distribution showing all comments plotted against the relevance scores to each of the two concepts (Figure 6.3). Here she maps the 'tidal flooding' concept on the x-axis and the 'economy' concept on the y-axis. She then brushes on the scatterplot to find comments containing both concepts, and these comments are filtered into the ranked list. She finds an insightful comment she likes that perfectly combines the two concepts, discussing coastal flooding in terms of impacts to the economy as exposed through the insurance industry. She marks the comment as a "NYT Pick" and it gets highlighted on the site.

She then begins to read those comments with high scores from the top of the list and quickly finds an insightful one indicating that some of the coastal flooding in Virginia has actually been shown to be a result of subsidence of land. Laurie thinks that highlighting this will deepen the discussion online by pointing out the diverse factors that society needs to grapple with as it confronts global warming. Therefore, she marks this comment as an "NYT Pick" as well.

6.5 The ConceptVector System

Motivated by the limitations of using prebuilt lexica for concept-based document analysis, we designed ConceptVector as a visual analytics system that tightly integrates concept building and refinement with direct support for concept-based document analysis. In detail, our design rationale behind ConceptVector is as follows:

D1 **Supporting diverse user needs in concept building.** Users may have diverse meanings in mind for defining their concepts. Thus, users should be able to construct the lexicon of a concept from scratch and/or refine a prebuilt one to suit to their exact requirements.

D2 **Supporting integrated analysis of iterative lexicon refinement and concept-based document analysis.** As seen from our motivational examples (Section 6.3), even carefully curated lexica need to be adjusted depending on a document corpus. Thus, the concept-based document analytics system should provide interactive refinement capabilities of a lexicon as well as dynamic document analysis based on the updated lexicon.

D3 **Revealing lexicon word context in documents.** The system should allow users to understand how the words in a lexicon are used in documents in terms of their context.

In this section, we explain how our front-end interfaces and the back-end computational modules support these tasks, and associate each component with design guidelines.

6.5.1 Front-end Visual Interface

Based on our design rationale, the text analytics process in ConceptVector is composed of two iterative processes: concept building and document analysis (Figure 6.5). We introduce the two views that allow the user to interactively build concepts and analyze documents.

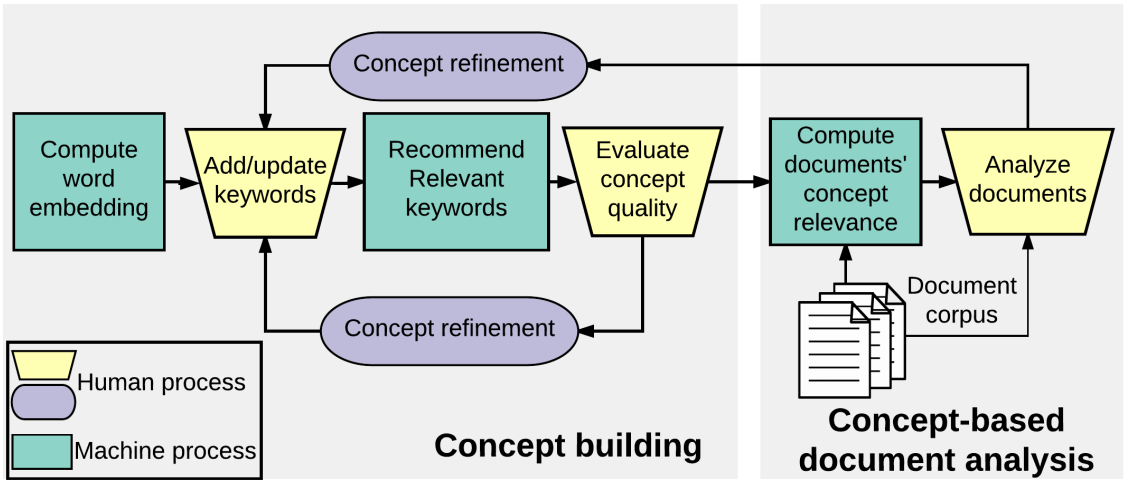


Figure 6.5: ConceptVector includes both human-guided and automatic steps. Green blocks represent human steps and orange blocks represent automatic steps. The process includes two iterative loops: an initial *construction loop* where users manually add new or recommended words to the concept until the quality is acceptable, and a *refinement loop* where users modify the concept based on scoring a document collection using the model.

6.5.1.1 Concept Building View

As shown in the left pane of Figure 6.5, the *concept building* process allows a user to interactively build the keyword sets describing a user's intended concept. Figure 6.4 shows a screenshot of our front-end interface that was taken during this process when the user was building the 'tidal flooding' concept.

We define two types of concepts: bipolar and unipolar. Bipolar concepts have two

nontrivial polarities, e.g., positive vs. negative sentiments, happiness vs. unhappiness, etc., while unipolar concepts have a single polarity, e.g., work-related (or not), biology-related (or not), etc. To support both concept types, ConceptVector models a particular concept using three different sets of keywords: positive, negative, and irrelevant (**D1**). In the case of unipolar concepts, the positive keyword set contains those keywords relevant to a concept of interest, while the negative set is an empty set. For both types, the irrelevant keyword set includes the words marked explicitly as irrelevant by the user.

The user starts building a concept by adding seed keywords to describe the concept. ConceptVector then recommends keywords that are potentially relevant to the seed keywords for each positive and negative keyword set, and performs k-means clustering, where we set k as 5, based on their word embeddings. Keyword clusters are presented to the user (Figure 6.4(3)), along with their 2D embedding view, computed by t-distributed stochastic neighbor embedding (t-SNE) [58] (Figure 6.4(4)). Checking these recommendation results, the user can either expand the initial keyword set by (1) adding individual words, (2) adding a keyword cluster of them, or (3) move words to the irrelevant set by marking them as irrelevant (**D1**). This iterative *concept building* continues until the user is satisfied with the constructed keyword set.

As relevant (or irrelevant) keywords often appear together in a single cluster, processing words at the cluster level makes the concept building process much more efficient than without clustering (**D1**). For example, if a user enters 'happy' as the only keyword for a concept, irrelevant words such as 'everyone,' 'anyway,' 'yes,' and 'anymore' are recommended as a single cluster, while semantically relevant words such as 'glad,' 'good,' and 'thrilled' form another cluster. When the semantic distinction among words is not

clear, users can tag individual words in the cluster. The t-SNE embedding space has very strong neighboring effects [122, 127], placing similar words closely to each other, and hence the 2D embedding view shows the distribution among user-initiated keywords and recommended ones. Users can enter/remove keywords in the t-SNE view as well (**D1**).

6.5.1.2 Concept-Based Document Analysis View

The concept-based document analysis view, shown in the right pane of Figure 6.4, allows the user to analyze a document corpus with respect to the constructed concepts. See Section 6.4 for a detailed description.

Given a single or multiple user-selected concepts, ConceptVector computes the relevance scores of documents for each concept and retrieves/ranks those documents with high score values (Figure 6.4(10)), which would be meaningful to the user who created/selected the corresponding concept. To help the user understand why these documents have high scores, the significantly contributing keywords are highlighted in yellow color (**D3**). Please note that our relevance scoring algorithm is not limited to the keywords registered in the positive/negative/irrelevant sets, but that other keywords potentially relevant to the concepts are considered as well. We will describe the algorithm further in the following section.

Additionally, ConceptVector provides two different views: a temporal view showing the concept strength over time, and a scatterplot showing the distribution of documents according to the relevance scores for the two different concepts, e.g., 'tidal flooding' vs. 'economy' concepts (Figure 6.3). According to the Jänicke et. al., extraction, evolu-

tion, and clustering are the three main tasks in visual text analysis [155]. The temporal view supports the temporal tracking of the topic signal evolution, while the scatterplot allows mapping/clustering documents in semantic space. Users can assign user-defined concepts as axes of the scatterplot to explore the distribution of the semantic meaning of documents (**D2**). Note here that we use a modified version of a scatterplot, where both dimensions are binned and dots are scaled to fill the assigned space [156]. This improves the visibility of outliers and densely overplotted areas. In these views, the user can brush over a time axis or data items to filter data in the ranked retrieval results.

During the process, the user may add additional words to the relevant and the irrelevant keyword sets of the concept (**D2**). For example, when applying the 'tidal flooding' concept shown in Figure 6.4 to a document corpus, the word 'disaster' was highlighted owing to its high relevance score to the concept. Since this word is not related to the 'tidal flooding' concept, the user can add it to the irrelevant keyword set to revise the concept and update the ranking of documents accordingly. This interaction allows in-situ concept refinement.

Note that the two analysis tasks of concept building and document analysis are not separate but tightly connected in ConceptVector, so that the user can fluidly switch between concept building/refinement and document analysis based on concepts.

6.5.2 Back-end Relevance Scoring Model

ConceptVector is built upon the vector representations of words generated by word embedding techniques such as word2vec [122] or GloVe [127]. In this step, the train-

ing corpus for word embeddings could be a generic one such as Wikipedia articles or a corpus within a particular domain, so that the trained vectors can better reflect the semantics of the domain. ConceptVector currently adopts pretrained vector embedding using Wikipedia articles by GloVe.[8] ConceptVector represents a concept C as the three set of keywords: the positive, the negative, and the irrelevant ones—L_p, L_n, and L_i, respectively. Given a word or a document, ConceptVector computes its relevance scores to the concept, based on the probability of a given word belonging to each of L_p, L_n, and L_i using a kernel density estimation (KDE) method.

In detail, let us denote q as the vector representation of a query word, l as that of the keyword contained in the keyword set L, where L can be one of L_p, L_n, or L_i. We define the probability of q belonging to L as

$$p(q|L) = \frac{1}{|L|} \sum_{x \in L} k(q, l), \tag{6.1}$$

where $k(q, l)$ represents a kernel function computing the similarity value between the two word vectors q and l. That is, Eq. (6.1) computes the average similarity values between q and each word l contained in a particular keyword set L. The reason for using a kernel function instead of a simple similarity measure such as cosine similarity is because this provides not only a user-controllable, flexible similarity measure but also a principled probabilistic framework of incorporating multiple similarities of q with L_p, L_n, and L_i, as will be described later.

The choice of the kernel function $k(q, l)$ can vary, but in ConceptVector, we adopted

[8] http://nlp.stanford.edu/projects/glove/

a Gaussian kernel defined as

$$k(q,l) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|q-l\|_2^2}{\sigma^2}\right),$$

where σ^2 is the bandwidth parameter that determines how quickly the similarity decreases as the L^2 distance increases. A small bandwidth value gives a high similarity only on the words exactly contained in L, which is suitable when L contains many words and a user does not want to consider other words outside L as relevant to the concept. A large bandwidth, on the other hand, will consider many of the outside words as relevant to L, which is useful when a user wants to define the concept in a broad and flexible manner, not just limited to those words contained in L.

Viewing $p(q|L)$, which is computed by Eq. (6.1), as the likelihood in a Bayesian context, we can define the prior probability $p(L)$ and the posterior probability $p(L|q)$, respectively, as

$$p(L) = \frac{|L|}{|L_p| + |L_n| + |L_i|}, \text{ and}$$

$$p(L|q) = \frac{p(L) \cdot p(q|L)}{p(L_p) \cdot p(q|L_p) + p(L_n) \cdot p(q|L_n) + p(L_i) \cdot p(q|L_i)}.$$

Using these, the final relevance score $r(q,C)$ of a query word q to the concept C is computed as

$$r(q,C) = (1 - p(L = L_i|q)) \cdot (p(L_p) \cdot p(q|L_p) - p(L_n) \cdot p(q|L_n))$$

Basically, $r(q,C)$ computes the differences between the joint probabilities $p(q,L_p)$ and $p(q,L_n)$, ranging between -1 and $+1$, and furthermore, as $p(L = L_i|q)$ increases, $r(q,C)$ becomes close to zero, indicating irrelevance to the concept.

In the case of a unipolar concept, the relevance score is computed in the exact same manner by setting $L_n = \emptyset$. These bipolar scores and unipolar scores are used for recommendation of relevant words.

Finally, the relevance score of a document to a particular concept is computed by simply taking the average relevance score among all the words contained in a document.

6.5.3 Implementation Details

ConceptVector was implemented as a web-based application using D3 and AngularJS. We employed the New York Times online article comments as our corpus; naturally, the approach can be applied to any document corpus. We selected articles with more than 300 comments from the most popular articles during the period August to September 2016. Articles and comments were collected using the NYT API.[9]

The back-end computational modules were implemented using Python with the Flask framework.[10] The key computation shown in Eq. (6.1) for recommending relevant words requires computing the one-to-all distances for all words in the current keyword set (either positive, negative, or irrelevant). Computing a single one-to-all distance repeatedly due to frequent user interaction may slow down the overall process. We instead compute the one-to-all distance incrementally with a cache that contains recently computed pairs. This is possible because the user incrementally adds a single word at a time to the keyword set. To this end, a least recently used cache of size 10,000 word pairs was employed, resulting in a speed-up of efficient user interactions.

[9] http://developer.nytimes.com/

[10] http://flask.pocoo.org/

6.5.4 Quantitative Evaluation of Bipolar Concepts

We validate the bipolar concept model supported by ConceptVector to address the following two questions:(1) Does our proposed approach generate relevance scores comparable to human judgments? and (2) How many input words are required to properly model concepts? To answer these questions, we conducted a quantitative analysis.

6.5.4.1 Experiment Setup

Validation of a lexicon requires ground truth. For unipolar concepts, the prior work from Fast et al. compared the result with "golden standard dictionaries" such as LIWC and GI [138]. While many lexica for unipolar concepts have been developed, bipolar lexica are rare. In this study, we adopted a keyword database available from the Hedonometer project[11] [124]. This database contains a ranked list of 10,200 keywords in terms of their relevance to the concept of 'happiness,' where the ranking was determined by crowdsourcing. The word ranking begins with the happiest word and ends with the saddest word. From this database, we selected 9,600 words from the intersection of the Hedonometer ranking and the vocabulary set from the Wikipedia corpus[12] used to train our word embedding model. From the Wikipedia corpus, we removed 71,697 documents that no longer exist, and used the resulting 171,729 articles. We then removed the words containing nonalphanumerical characters as well as those appearing less than ten times in the entire document corpus, resulting in 142,275 keywords in total.

[11] http://hedonometer.org/

[12] https://cs.fit.edu/~mmahoney/compression/textdata.html

The goal of our experiments was basically to evaluate how well the ranking of words computed by our back-end algorithm matches with the ground truth ranking, given a subset of top and bottom k words as positive and negative sets, respectively, to form a concept. As the methods to generate word vector representations, we used two different word embedding techniques—word2vec [122] and GloVe [127]—as well as a baseline method, latent semantic indexing (LSI) [126]. Additionally, in each vector space, we compared our KDE-based algorithm against logistic regression for computing the word-to-concept relevance score and the associated word ranking. As an evaluation measure, we computed Spearman's rank correlation coefficient between the ranking of the ground truth and that from each different case.

6.5.4.2 Comparison Results

Figure 6.6 shows the Spearman's rank correlation coefficients obtained for various word embedding and relevance scoring methods by varying the value of k in the top k and the bottom k keywords used to train each model. In general, given a small number of input keywords less than 200, the algorithm was shown to generate a reasonably good rank correlation of more than 0.4. In addition, as we increase k, the rank correlation increases in all cases, indicating that more information helps the model learn the intended concept (happiness in this case). Between the two word embedding methods and LSI, the former showed a rapidly increasing performance even with a small number, e.g., around 100, of keywords necessary for training. Between our KDE-based scoring method and logistic regression, the former outperformed the latter method when the size of the keyword set is

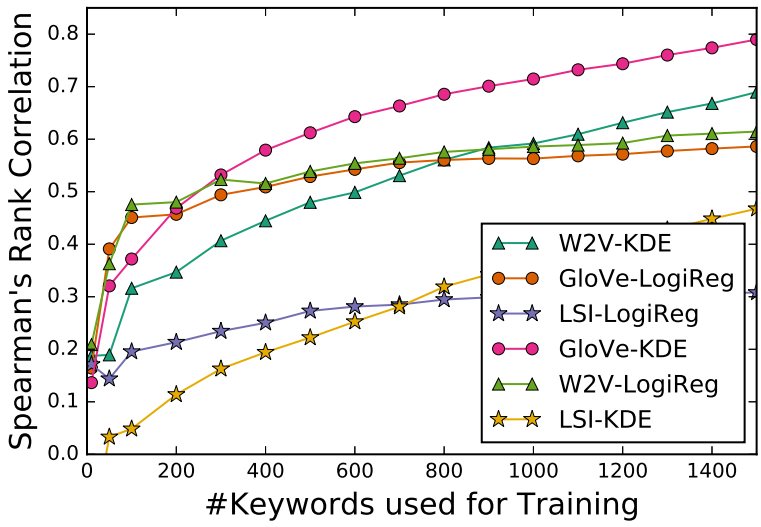


Figure 6.6: Spearman's rank correlation coefficient results with respect to the number of keywords used for training. `KDE` stands for kernel density estimation, `LogiReg` for logistic regression, `W2V` for word2vec [122], `LSI` for latent semantic indexing, and `GloVe` for GloVe [127].

sufficient, e.g., more than 300. Furthermore, the performance of our KDE-based method consistently increases by a large margin compared to competing methods.

Among different word embedding techniques, the GloVe model followed by the KDE-based method achieved the best rank correlation performance of around 0.8. Word2vec performed relatively well, but it was inferior to GloVe in our task. On the other hand, traditional methods such as LSI do not perform well in this task, showing a rank correlation of 0.45 even with large k values. Finally, the overall performance gain due to the increase of the embedding dimensions was not significant.

Our experiment involves only bipolar concepts (no unipolar ones), and we did not examine the effect of an irrelevant keyword set. In this case, logistic regression may not be applicable at all. In addition, we found that the ground truth ranking is not always correct, especially among the mid-ranked unclear words. However, the results presented here

highlight the potential superiority of our proposed KDE-based scoring approach combined with GloVe, and in the next section, we present results from an expert review to show the effectiveness of the system when used in practice.

6.6 Evaluation

Visual analytics systems comprise many interconnected components, and this complicates their overall evaluation. Here we separate the visual interface and the back-end computation and evaluate them individually with a user study and a quantitative evaluation, respectively. For the front-end, we focus on the effectiveness of the concept-building view because document analysis requires analysts with domain knowledge and is subjective to inter-analyst differences. For the back-end, we validate the effectiveness of supporting the process of building bipolar concepts. Although we did not evaluate a unipolar case, we generally expect the same level of effectiveness since the process of building unipolar concepts is similar yet simpler than the process of building bipolar concepts. Finally, we also include results from an expert review comparing ConceptVector to Empath [138] to show ConceptVector's performance in relation to the state of the art.

6.6.1 Evaluation of Concept Building

We conducted a user study to evaluate how users generate lexica with ConceptVector compared to WordNet [157][13] and Thesaurus.com[14] as baselines. WordNet is known for its large-scale lexical database, and Thesaurus.com is an online thesaurus containing

[13] http://wordnetweb.princeton.edu/perl/webwn

[14] http://www.thesaurus.com/

Table 6.1: Precision, recall, and average number of keywords per concept for three methods constructing user-defined concepts. The values in parentheses indicate the standard deviation. See Section 6.6.1.2 for details.

Metrics	ConceptVector	Thesaurus	WordNet	F value	Pr > F
Precision	**0.6363** (0.1701)	0.3099 (0.3773)	0.3794 (0.3637)	5.22 [2, 40]	0.0096
Recall	**0.0789** (0.0308)	0.0333 (0.0385)	0.0275 (0.0242)	5.25 [2, 40]	0.0094
Mean word count	**15.6667** (7.4536)	13.8000 (6.0685)	8.2667 (3.3360)	5.40 [2, 40]	0.0084

exhaustive synonyms and antonyms for the English language. We employed the following performance metrics: (i) the completion time for building concepts, and (ii) the quality of the resulting concepts.

6.6.1.1 Methodology

We recruited 15 graduate students (1 female and 14 males) majoring in computer science to participate in the study. All participants reported high computer skills.

Each study session lasted 15–25 minutes and involved three systems: ConceptVector, WordNet, and Thesaurus.com. Before starting the session, a test administrator briefly explained how to use the systems and allowed the participant to spend enough time to familiarize themselves. Participants were then asked to build a lexicon for three concepts: 'family,' 'body,' and 'money,' which we selected as relatively neutral and easily compre-

hensible by all participants. Each participant was randomly assigned to a system for each concept so that at the end of the study they had used all three conditions. Each concept-building task was capped at three minutes. All three systems, including ConceptVector, were accessed by their official websites. We recorded both the lexicon each participant created as well as the number of keywords in it as a dependent variable.

As the ground truth lexicon for each concept, we selected three dictionaries from Linguistic Inquiry and Word Count (LIWC) 2007 [123]. The ground truth lexicon sizes of the three concepts are 65 words for 'family,' 180 for 'body,' and 173 for'money.' We adopted widely used information retrieval evaluation metrics, precision and recall, where precision is the fraction of correct answers over the total number of answers given, and recall is the fraction of retrieved correct answers out of all correct ones. The null hypothesis assumes that the difference of methods does not affect the precision, recall, or average number of words in the resulting lexicon.

6.6.1.2 Results

Table 6.1 shows precision, recall, and average total words generated for the three methods. ConceptVector achieved the highest scores in all three metrics, indicating that the user-created lexicon using ConceptVector is the most accurate and most time-efficient. We further analyzed the effect of employing ConceptVector using mixed linear model analysis, where the fixed effect is the choice of methods (ConceptVector, WordNet, and Thesaurus.com) and the random effect is the choice of specific concepts ('family,' 'money,' and 'body').

Figure 6.7 shows boxplots for precision, recall, and total words generated. We used a pairwise Tukey HSD method to test statistical significance between different methods. There was a significant performance boost of employing ConceptVector on recall ($F(2,40) = 5.25, p = .0094$). Pairwise Tukey HSD between ConceptVector and the other methods showed significant differences ($p < .05$). There was also a significant main effect for technique T on precision ($F(2,40) = 5.22, p = .0096$). Pairwise comparisons with a Tukey HSD showed significant differences ($p < .05$) between ConceptVector and Thesaurus.com. Finally, there was a significant main effect for technique T on the number of total words generated ($F(2,40) = 5.40, p = .0084$). Pairwise comparisons with a Tukey HSD showed significant differences ($p < .05$) for ConceptVector and WordNet.

Recall rates in all three systems are relatively low compared to the high precision rates. This is mainly because the size of the ground truth lexicon is much larger than the average size of the lexicon a person can create within a short period of time (three minutes in our case). As seen in Table 6.1, the average size of the created lexicon was around 8 to 15 depending on the system.

Since the current experimental design does not consider polysemy or subtle nuance differences, this experiment could be improved further by employing more sophisticated ground truth data instead of the current ones obtained from LIWC. For example, the 'family' concept may diverge in terms of its subtler meanings to different people. On the one hand, it may correspond mainly to the members of a family such as 'mother,' 'grandfather,' and 'son.' On the other hand, it may correspond to emotional words such as 'love,' 'rest,' and 'nursing.' Since our ground truth lexicon from LIWC was mostly composed of the keywords from the first case, the user-generated keywords from the second case were

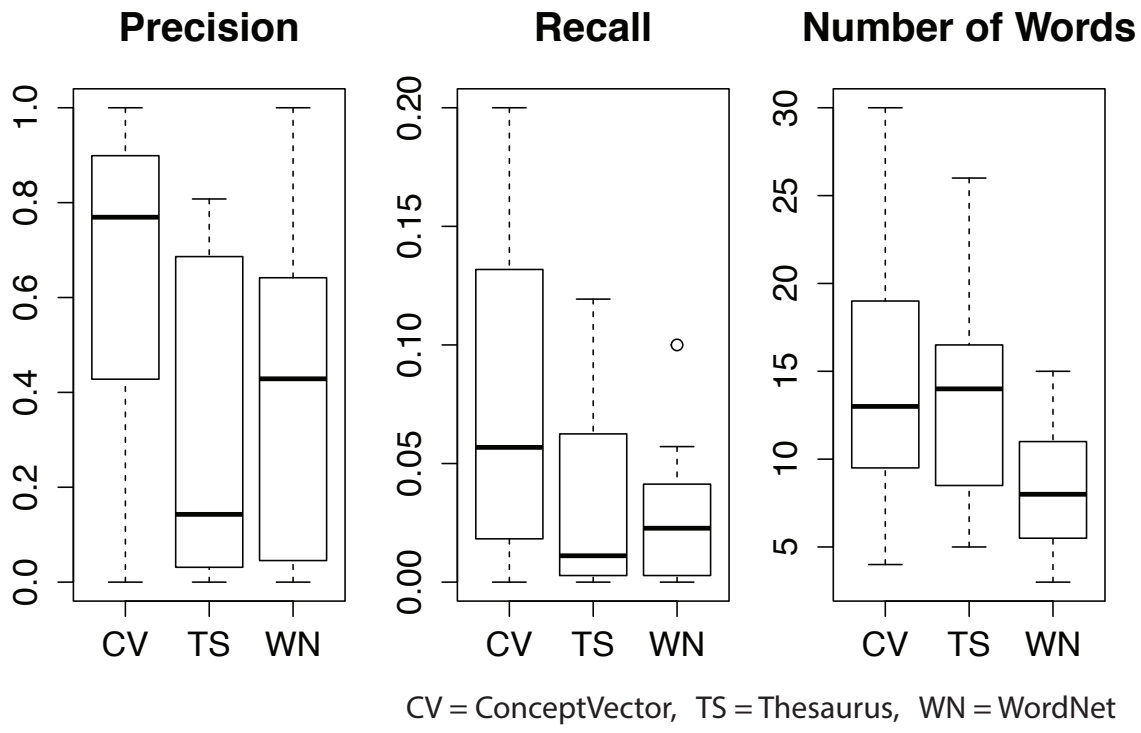


Figure 6.7: Boxplots comparing the three methods in terms of precision, recall, and the size of resulting lexicon. ConceptVector shows the best result.

treated as false positive words. We expect that ConceptVector will perform better if we find ground truth data that enable us to measure these richer relations, and this will be one of our future directions. Furthermore, regardless of which type of concept a user had in mind, ConceptVector properly supported the concept-building process by recommending suitable keywords for different cases. This indicates the flexibility and the affordance that ConceptVector offers compared to other, more rigid, systems.

6.6.2 Expert Review

To evaluate the visual interface of ConceptVector in depth, we engaged two experts, one in text analytics (P1) and one in visual analytics (P2), to provide qualitative feedback on ConceptVector compared to the Empath system. Both were postdoctoral researchers,

the former of which conducts research on social networks, crisis analytics, and credibility in social media, while the latter studies the healthcare domain that frequently involves text mining and visualization (e.g., electronic medical records). We began with a 15-minute tutorial introducing our system and Empath. Afterwards, the experts built their own custom concepts using both systems. Those concepts were used in the analysis of New York Times comments. During the process, we gathered the feedback on both the model and the visual interface of the system. The online version of Empath was used as a reference.[15]

Both P1 and P2 agreed that the recommended keywords given by ConceptVector, which are shown in a semantically meaningful grouping in a scatterplot, were easier to grasp than the simple list given by Empath. The scatterplot helped them digest the generated words by providing a high-level overview (P2) or chunking the words into semantically homogeneous groups (P1). It was especially useful in the early stage of concept building, because irrelevant words formed a separate group in many cases, allowing the user to spot them easily and mark them as irrelevant. The word clusters (Figure 6.4(3)) were good for reading words quickly (P2), and was used during most of the concept-building process (P1). Furthermore, the t-SNE view (Figure 6.4(4)) provided an additional benefit of showing the similarities between words (P1) and the relationships between the input terms and the recommended terms (P2). For example, whether the input words form tight clusters or not gives a visual clue as to whether the generated concept is consistent (P1). At the same time, P1 noticed that an input term was actually an outlier compared to other input terms forming a packed cluster. After examining this word, he

[15] http://empath.stanford.edu/

removed it because it had a very broad meaning and thus dilutes the clarity of the concept.

Both experts noted that the difference between the corpora used to train word embedding affects the concept quality. Empath used modern amateur fiction data, but ConceptVector used the Wikipedia dataset. For example, when P1 used 'politics,' 'voting,' and 'elections' as seed terms in Empath, the generated words contained several words such as 'shipping' and 'readers' which did not really make sense. According to P1, Empath also generated more 'high school'-related words. This does not necessarily mean that one system is better but rather that using word embedding trained by a corpus suitable for target corpus to analyze is important. After building a concept about 'grievance,' P1 noted *"The recommended words for the grievance concept is different from what I saw on social media. That is, many legalese and lengthy words related to grievance were recommended, but very unlikely to show up on social media."* P2 suggested using ConceptVector as a tool to evaluate multiple versions of word embedding models during iterative model development.

P1 and P2 both agreed that comparing Empath and ConceptVector is challenging, because the main focus of Empath is not its user interface. P1 thought the visual interface in ConceptVector was useful to explore the semantic space. Being able to look around and select words that are not originally shown to him helped to expand the lexicon. P2 pointed out that the document analysis feature of Empath is more of a blackbox and felt uncomfortable with trusting the result. For example, when analyzing the Wikipedia page about 'Ramen,' the 'friends' category was ranked as the 6th, but it is not clear which words in the friends category were counted.

P1 noted that the word-highlighting feature of ConceptVector allows for the easy

spotting of false positives, but detecting false negatives is not currently supported. P2 appreciated the concept score scatterplot (Figure 6.4(7)) that showed the distribution of comments with respect to custom concepts as axes. It revealed outliers and enabled filtering of comments based on semantic contents. After using ConceptVector, P2 said that it could be useful to build a concept for drugs by adding related symptoms and using a positive/negative sentiment as another axis to visualize the sentiment for a particular drug. P2 also liked that the concept dictionary can be refined by trial and error.

P1 expressed concern about fundamental limitations of both systems. Both systems use word embedding based on the assumption that word co-occurrence statistics reflect semantic similarities, which might not be always true in real-world text analysis. P1 pointed out that while the color coding of words to highlight newly recommended words is an improvement over Empath, it was still difficult to follow the word changes according to the input terms. P2 liked the bipolar concepts feature because it helps in building more sophisticated concepts. As an alternative design, P2 suggested showing the words interpolating positive and negative terms. Those interpolated words will reveal the validity of a concept, as suggested in Axisketcher [158].

6.7 Discussion

ConceptVector is a novel approach for text analysis that falls somewhere between sentiment analysis performed using manually constructed dictionaries, and topic modeling performed by automatic algorithms. This unique position brings new benefits as well as limitations.

In general, when achieving a particular analytic goal, an interesting tradeoff between quality and efficiency can be considered. That is, human efforts secure the quality of the outcome, while automated approaches can significantly boost the efficiency of our efforts. For concept building, purely manual approaches such as LIWC and Hedonometer can be viewed as extreme cases, where the task relies completely on human effort. Thus, the resulting dictionary is of high quality, but it is achieved by an inefficient, costly process without automation. On the other hand, purely automated approaches such as topic modeling, which generate multiple sets of semantically coherent words, maximize the efficiency of the task, but the quality of the outcome cannot be controlled by the user. Human labor is still needed to interpret the results that such fully automatic approaches generate.

In this sense, our approach in the ConceptVector system can be viewed as a balanced—or hybrid—case, where both efficiency and interpretability are achieved via a synergetic blending of both human efforts and automated machine computations. That is, our main steps of adding and removing keywords to construct a particular concept are all confirmed by humans, and in this manner, a high quality outcome is maintained. However, our system significantly accelerates these human-guided processes by crucial automated approaches, including word recommendation based on word embedding, followed by word grouping and visual presentation. Also, after users build a specification, this specification is used to build the concept model, which calculates the relevance scores of all words with this particular concept. In this respect, our system represents an illustrative example for properly achieving human-machine collaborations. As it happens, this is also precisely in line with the visual analytics philosophy, where automatic algorithms and visual

interfaces create synergies .

6.8 Conclusion and Future Work

Current text analytics methods are either based on manually crafted human-generated dictionaries or require the user to interpret a complex, confusing, and sometimes nonsensical topic model generated by the computer. In this paper we proposed ConceptVector, a novel text analytics system that takes an visual analytics approach to document analysis by allowing the user to iteratively define concepts with the aid of automatic recommendations provided using word embeddings. The resulting concepts can be used for concept-based document analysis, where each document is scored depending on how many words related to these concepts it contains. We crystallized the generalizable lessons as design guidelines about how visual analytics can help concept-based document analysis. We compared our interface for generating lexica with existing databases and found that ConceptVector enabled users to generate concepts more effectively using the new system than when using existing databases. We proposed an advanced model for concept generation that can incorporate irrelevant words input and negative words input for bipolar concepts. We also evaluated our model by comparing its performance with a crowdsourced dictionary for validity. Finally, we compared ConceptVector to Empath in an expert review.

The text analysis provided by ConceptVector enables several novel concept-based document analysis, such as richer sentiment analysis than previous approaches, and such capabilities can be useful for data journalism or social media analysis. There are many limitations that ConceptVector does not solve. Among these, the selection/integration

of multiple heterogeneous training data according to the target corpus and the automatic disambiguation of multiple meanings of words according to the context are promising avenues of future research.

Chapter 7: Conclusion

We will first summarize the interaction patterns from five design studies. Then, the limitations of the visual analytics approach for open-ended tasks will be discussed. Finally, we conclude with a plan for future work.

7.1 Summary of Projects

The five projects presented here were our efforts to improve the state of the art for the visual analytics approach for open-ended tasks in text mining. While these projects deal with specific target usage scenarios, it would be helpful if we can extract lessons that can be applied to broader domains. These may overlap with many previous design guidelines that existed before and that we have tried to cite in the preceding chapters. Still, our restatement here might contribute supporting evidence for this pioneering work.

- **Interactive Overview:** A user characterizes document clusters interactively by assigning meaning to the locations via (1) using an existing attribute, (2) building a new custom axis as a mixture of existing attributes, or (3) defining a semantic axis using a dictionary of intended words and using it to layout the documents.

- **Unit Visualization:** Maintaining individual objects rather than aggregating them is

helpful to enable unknown discovery.

- **User Interaction as Additional Input to the Algorithm:** A user provides additional parameters, such as a region of interest, that the algorithm can use to optimize computation time and interpretation time.

- **Collaboration by Sharing Contributions:** A user creates and shares reusable by-products of the analysis.

In the following subsections, each design pattern will be discussed.

7.1.1 Interactive Overview

Recommender systems are highly developed domains with many algorithms. While these algorithms are capable of recommending one item to one user, they fail to provide an overview of the user's preferences. Visualizing the overview is also helpful because it can mitigate the bias of the user. Further, previous overview visualization methods tend to use layouts where similar objects are closer, but the location in the visual space does not have semantic meaning. This makes spotting clusters or groups simple but interpreting them difficult. For example, community moderators read thousands of comments to select high quality instances. They are afraid that they might develop biases while reading them, which affects their ability to select high-quality comments fairly.

Users can see an overview of document clusters with scatterplots using MDS. But the static overview has the limitation that it derives from dimensional reduction, meaning much information is lost. Interactivity can help here. In ParallelSpaces, users can change axes to explore the semantics of each cluster. In CommentIQ, community moderators

can see an overview distribution of the comments in terms of geographical, temporal and feature spaces. ConceptVector supports the creation of user-driven semantic features to enrich the analysis. In TopicLens, interactivity provides a balance between overview and details by dynamically changing the number of the clusters in the overview.

7.1.2 Unit Visualization

Previous statistical graphics tend to create aggregated entities. While there is benefit to this in terms of computational resources and summarization, we claim that maintaining individual data objects in the visual space has many benefits for open-ended tasks in text mining.

ParallelSpaces allows examination according to individual preference. However, a scatterplot with categorical variables creates an overplotting issue. Gatherplots provide a way to mitigate this problem. In CommentIQ and ConceptVector, individual comments are marked with a graphical object. In the geographic view, marks for comments with the same location, such as New York, are spread out using force-directed layouts to avoid overlap.

7.1.3 User Interaction as Additional Input to the Algorithm

The "no free lunch" theorem suggests that the computational cost to solve a problem cannot fundamentally be reduced. However, by providing additional context or constraints, certain algorithms will have advantages over others. TopicLens is an example of utilizing a user's interaction data as an additional input to the algorithm. The user pro-

vides a region of interest as context. The algorithm can use this input to lower calculation time and interpretation time.

7.1.4 Collaboration by Sharing Contributions

The power of visualization is that it lowers the barriers of computational analysis. This leads to the democratization of data analysis and encourages participation from the public. The visual analytics system should thus encourage the sharing of byproducts of the analysis in order to promote collective intelligence, or the *wisdom of the crowd*.

In CommentIQ, a user can build a custom model of high-quality comments and share it with other moderators, amounting to sharing domain knowledge among the community. Also, after high-quality comments are selected, the system allows tagging of the comments with the reason for selection. This tag data can be used for future research.

In ConceptVector, the byproduct of the analysis is the custom concepts. These concepts can be used for analysis by other users.

7.2 Limitations of VA Approaches

While the visual analytics approach complements the statistical machine learning approach, it also has limitations. Interestingly, these limitations originate from the unique characteristics that make each visual analytics approach effective: (1) Domain/Application specific systems, (2) Back-end algorithms, and (3) Human engagement. In the following subsections, we will explore each of these.

7.2.1 Domain/Application specific systems

Building a visual analytics systems requires a significant amount of resources. Yet, visual analytics systems are usually domain-specific. The nine-stage design study methodology framework developed by Sedlmair et al. begins with the precondition stage, where researchers learn about the domain [116]. The knowledge gained from these steps cannot be transferred to other domains. Furthermore, the subsequent iterative development stage adapts to feedback from the users. This results in a highly specialized system for the specific domain. Even though the participatory nature of visual analytics is helpful at first, the development cost for the specialized system makes wide adoption challenging.

Let's compare it with the ideal human assistant. He/she will learn from the curriculum in school and apply what was learned to wider areas of problem solving. He/she accumulates knowledge about the world; therefore, initial investment in education on a specific topic results in the exponential growth of the ability.

7.2.2 Back-end algorithms

The main idea behind the visual analytics approach is to use the computational power of machine learning to assist reasoning by humans. In the text mining domain, you have to convert raw text into numbers before visualizing it. A lot of information loss happens in this step.

For example, many topic modeling algorithms, such as pLSA, LDA, and NMF, rely on the bag-of-words assumption. In this model, all syntactic information is lost, limiting the semantic analysis.

While the visual analytics approach relies on human capability to overcome such limitations, this usually results in many false positives for a human to detect, which increases the required human effort. As a metaphor, let's assume we have an assistant, but one with very limited knowledge. To prevent errors, this assistant asks for our confirmation before making uncertain actions. The frequency of asking confirmation will then inversely correlate with the assistant's knowledge. If knowledge is too low, the assistant will not be helpful because we will be bugged too much. By increasing our assistant's knowledge, we can focus on more important things while allowing our assistant to work on assigned tasks.

7.2.3 Human involvement

As machines improve, they are taking more responsibility for mission-critical tasks such as driving, loan approval, or criminal prediction. For some applications, having a human in the loop makes sense for legal responsibility. However, in many cases, rather than being a desired property of the system, the human in the loop is a consequence of the unreliable and unsatisfactory nature of the machines. Visual analytics approaches are, in this regard, like a medicine that mitigates the symptoms of a disease rather than curing it fundamentally.

7.3 Future Work

Our society is becoming increasingly dependent on the ability to analyze its own data and extract knowledge and insights from that data. I will focus on the various human

aspects of these processes, collectively called Human-Data Interaction. Building on previous research, there are multiple opportunities for follow-up study. Here I propose some research ideas that I would like to pursue in the future.

Discourse Identity Mining will extract open personal information about the authors of social media texts, such as job, age, and gender. For example, from a news article about the Ferguson shooting, comments from police officers and African Americans can be meaningful. Therefore, the ability to see the demographic information, and to search for comments from specific groups, can be valuable. Interestingly, many people identify themselves in their comments to give relevance for their arguments. I envision three major problems for developing systems for authorship information. The first is constructing a schema for the personal identity. Humans can possess many aspects of identity, for example, in the case of Ferguson, a commenter has identified herself as "a white mom with two black kids." This includes gender, race, and family structure. Secondly, identity can be expressed in many natural language forms. For example, a person's occupation can be expressed in various ways. Developing a parser for such information is thus another challenge. Finally, once I have structured demographic information about authorship, the third challenge will be presenting the information effectively, so that the end-users of the system can get an overview of the distribution and find interesting demographic segments.

CaptionViz will try to explore large numbers of captions for images. Recent advances in automatically describing images with natural language are motivating, because the task is a multi-modal one crossing both the language and vision domains. However, evaluating model performance is difficult, because it is an open-ended task with unstructured data. Previous methods for automatic evaluation generate a single aggregate score.

While these scores help rank model performance, they do not guide researchers in identifying problematic areas of the models or recommend how to improve the models. In CaptionViz, I will model an image captioning task as a five-stage process that includes object recognition, attribute detection, relation detection, salience detection, and sentence generation. Errors and problems in each stage will be measured and visualized with multiple levels of detail to enable exploration of model performance.

Bibliography

[1] Don Norman. *The design of everyday things: Revised and expanded edition*. Basic Books (AZ), 2013.

[2] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

[3] Saleema Amershi. *Designing for effective end-user interaction with machine learning*. PhD thesis, University of Washington, 2 2012.

[4] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343. IEEE, 1996.

[5] William N Dunn. *Public policy analysis*. Routledge, 2015.

[6] Susan Havre, Beth Hetzler, and Lucy Nowell. ThemeRiver: Visualizing theme changes over time. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 115–123, 2000.

[7] Trevor F. Cox and Michael A. A. Cox. *Multidimensional Scaling*. Chapman & Hall/CRC, London, 2000.

[8] William S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 30 1985.

[9] Jock Mackinlay. Automating the design of graphical presentations of relational information. *Acm Transactions On Graphics (Tog)*, 5(2):110–141, 1986.

[10] Jeffrey Heer and Michael Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 203–212. ACM, 2010.

[11] Deokgun Park, Jungu Choi, and Niklas Elmqvist. Parallelspaces: Simultaneous exploration of feature and data for hypothesis generation. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 1437–1445. IEEE, 2016.

[12] Jingu Kim and Haesun Park. Sparse nonnegative matrix factorization for clustering. 2008.

[13] Deokgun Park, Simranjit Sachar, Nicholas Diakopoulos, and Niklas Elmqvist. Supporting comment moderators in identifying high quality online news comments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1114–1125, 2016.

[14] Deokgun Park, Seungyeon Kim, Jurim Lee, Jaegul Choo, Nicholas Diakopoulos, and Niklas Elmqvist. Conceptvector: text visual analytics via interactive lexicon building using word embedding. *IEEE transactions on visualization and computer graphics*, 24(1):361–370, 2018.

[15] Minjeong Kim, Kyeongpil Kang, Deokgun Park, Jaegul Choo, and Niklas Elmqvist. Topiclens: Efficient multi-level visual topic exploration of large-scale document collections. *IEEE transactions on visualization and computer graphics*, 23(1):151–160, 2017.

[16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[17] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

[18] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[19] DL Applegate, RM Bixby, V Chvátal, and WJ Cook. The traveling salesman problem. *ISBN: 0-691-12993-2*, 2006.

[20] Marco Dorigo and Luca Maria Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66, 1997.

[21] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[22] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[23] S. P. Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 2:129–137, 1982.

[24] Lisha Chen and Andreas Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219, 2009.

[25] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[26] Ashley A. Anderson, Dominique Brossard, Dietram A. Scheufele, Michael A. Xenos, and Peter Ladwig. The "nasty effect:" online incivility and risk perceptions of emerging technologies. *Journal of Computer-Mediated Communication*, 19(3):373–387, 2014.

[27] Abhay Sukumaran, Stephanie Vezich, Melanie McHugh, and Clifford Nass. Normative influences on thoughtful online participation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 3401–3410, 2011.

[28] Hyunmo Kang, Catherine Plaisant, Bongshin Lee, and Benjamin B. Bederson. NetLens: iterative exploration of content-actor network data. *Information Visualization*, 6(1):18–31, 2007.

[29] Tatiana von Landesberger, A. Kuijper, Tobias T. Schreck, Jörn Kohlhammer, Jarke J. van Wijk, Jean-Daniel Fekete, , and Dieter W. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011.

[30] D.A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 8(1):1 –8, jan/mar 2002.

[31] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[32] Ivan Herman, Guy Melançon, and M. Scott Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[33] Ben Shneiderman and Aleks Aris. Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):733–740, 2006.

[34] John W. Tukey, Mary Anne Fisherkeller, and Jerome H. Friedman. PRIM-9: An interactive multi-dimensional data display and analysis system. In William S. Cleveland and Marylyn E. McGill, editors, *Dynamic Graphics for Statistics*, pages 111–120. Wadsworth & Brooks/Cole, 1988.

[35] Richard A. Becker, William S. Cleveland, and Ming-Jen Shyu. The visual design and control of trellis display. *Journal of Computational and Graphical Statistics*, 5(2):123–155, 1996.

[36] Matthew O. Ward. XmdvTool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the IEEE Conference on Visualization*, pages 326–333, 1994.

[37] Deborah F. Swayne, Duncan Temple Lang, Andreas Buja, and Dianne Cook. GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43(4):423–444, 2003.

[38] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008.

[39] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, May 1987.

[40] Martin Theus and Simon Urbanek. *Interactive Graphics for Data Analysis: Principles and Examples*. Chapman & Hall/CRC, 2008.

[41] Chris Weaver. Building highly-coordinated visualizations in Improvise. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 159–166, 2004.

[42] Chris Stolte, Diane Tang, and Pat Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[43] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*, pages 110–119, 2000.

[44] Jonathan C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of the International Conference on Coordinated Multiple Views in Exploratory Visualization*, 2007.

[45] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computer Surveys*, 31(3):264–323, September 1999.

[46] Jacques Bertin. *Semiology of Graphics*. University of Wisconsin Press, Madison, Wisconsin, 1983.

[47] Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proceedings of the International Conference on Machine Learning*, pages 46–54, 1998.

[48] Samuel Kaski, Timo Honkela, Krista Lagus, and Teuvo Kohonen. WEBSOM - self-organizing maps of document collections. *Neurocomputing*, 21:101–117, 1998.

[49] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, January 2004.

[50] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[51] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, pages 175–186, 1994.

[52] Joseph Konstan and John Riedl. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22:101–123, 2012.

[53] Tomoharu Iwata, Takeshi Yamada, and Naonori Ueda. Probabilistic latent semantic visualization: topic model for visualizing documents. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pages 363–371, 2008.

[54] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, volume 5, pages 2–4, 2005.

[55] M. P. Wand and M. C. Jones. Comparison of smoothing parameterizations in bivariate kernel density estimation. *Journal of the American Statistical Association*, 88(422):520–528, 1993.

[56] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[57] D. Kim, S. Sra, and I. S. Dhillon. Fast Newton-type methods for the least squares nonnegative matrix approximation problem. In *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM07)*, 2007.

[58] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[59] William S Cleveland and Marylyn E McGill. *Dynamic Graphics for Statistics*. CRC Press, 1988.

[60] Jessica M Utts. *Seeing Through Statistics*. Duxbury Press, 1996.

[61] Jean-Daniel Fekete and Catherine Plaisant. Interactive information visualization of a million items. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 117–124, 2002.

[62] Geoffrey P. Ellis and Alan J. Dix. A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1216–1223, 2007.

[63] Marjan Trutschl, Georges Grinstein, and Urska Cvek. Intelligently resolving point occlusion. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 131–136, 2003.

[64] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of the IEEE Conference on Visualization*, pages 43–50, 1999.

[65] A. Mayorga and Michael Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 19(9), September 2013.

[66] Jean-François Im, Michael J McGuffin, and Rock Leung. GPLOM: The generalized plot matrix for visualizing multidimensional multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2606–2614, 2013.

[67] Shumin Zhai, William Buxton, and Paul Milgram. The partial-occlusion effect: utilizing semitransparency in 3D human-computer interaction. *ACM Transactions on Computer-Human Interaction*, 3(3):254–284, 1996.

[68] Alan Dix and Geoffrey Ellis. By chance - enhancing interaction with large data sets through statistical sampling. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*, pages 167–176, 2002.

[69] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[70] Niklas Elmqvist and Jean-Daniel Fekete. Hierarchical aggregation for information visualization: Overview, techniques and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2010.

[71] Anastasia Bezerianos, Fanny Chevalier, Pierre Dragicevic, Niklas Elmqvist, and Jean-Daniel Fekete. GraphDice: A system for exploring multivariate social networks. *Computer Graphics Forum*, 29(3):863–872, 2010.

[72] Ben Shneiderman, David Feldman, Anne Rose, and Xavier Ferré Grau. Visualizing digital library search results with categorical and hierarchical axes. In *Proceedings of the ACM Conference on Digital Libraries*, pages 57–66, 2000.

[73] Benjamin B Bederson, Ben Shneiderman, and Martin Wattenberg. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854, 2002.

[74] Heike Hofmann, Arno P. J. M. Siebes, and Adalbert F. X. Wilhelm. Visualizing association rules with interactive mosaic plots. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pages 227–235, 2000.

[75] R. Kosara, F. Bendix, and H. Hauser. Parallel sets: interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, 2006.

[76] S. S. Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, June 1946.

[77] Hadley Wickham and Lisa Stryjewski. 40 years of boxplots. *American Statistician*, 2011.

[78] Leland Wilkinson. Dot plots. *The American Statistician*, 53(3):276–281, 1999.

[79] Tuan Nhon Dang, Leland Wilkinson, and Anushka Anand. Stacking graphic elements to avoid over-plotting. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1044–1052, 2010.

[80] Leda Cosmides and John Tooby. Are humans good intuitive statisticians after all? Rethinking some conclusions from the literature on judgment under uncertainty. *Cognition*, 58(1):1–73, 1996.

[81] Luanna Micallef, Pierre Dragicevic, and Jean-Daniel Fekete. Assessing the effect of visualizations on Bayesian reasoning through crowdsourcing. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2536–2545, 2012.

[82] Samuel Huron, Romain Vuillemot, and Jean-Daniel Fekete. Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, 2013.

[83] Steve Haroz and David Whitney. How capacity limits of attention influence information visualization effectiveness. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2402–2410, 2012.

[84] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[85] Bryan McDonnel and Niklas Elmqvist. Towards utilizing GPUs in information visualization: A model and implementation of image-space operations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1105–1112, 2009.

[86] Edward R Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, 1983.

[87] D. B. Carr, W. L. Nicholson, R. J. Littlefield, and D. L. Hall. Interactive color display methods for multivariate data. In *Naval Research Sponsored Workshop on Statistical Image Processing and Graphics*, pages 215–250, 1983.

[88] Herman Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

[89] Gabriele Paolacci, Jesse Chandler, and Panagiotis Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5):411–419, 2010.

[90] Wesley Willett, Shiry Ginosar, Avital Steinitz, Björn Hartmann, and Maneesh Agrawala. Identifying redundancy and exposing provenance in crowdsourced data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2198–2206, 2013.

[91] N. Diakopoulos. Picking the NYT picks: Editorial criteria and automation in the curation of online news comments. *ISOJ Journal*, 2015.

[92] Matthew Brehmer, Stephen Ingram, Jonathan Stray, and Tamara Munzner. Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2271–2280, 2014.

[93] Nicholas Diakopoulos, Munmun De Choudhury, and Mor Naaman. Finding and assessing social media information sources in the context of journalism. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 2451–2460, 2012.

[94] Kevin Coe, Kate Kenski, and Stephen A. Rains. Online and uncivil? patterns and determinants of incivility in newspaper website comments. *Journal of Communication*, 64(4):658–679, 2014.

[95] Nicholas Diakopoulos and Mor Naaman. Towards quality discourse in online news comments. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 133–142, 2011.

[96] Jane B. Singer. Quality control: Perceived effects of user-generated content on newsroom norms, values and routines. *Journalism Practice*, 4(2):37–41, 2010.

[97] C Lampe, E Johnston, and P Resnick. Follow the reader: filtering comments on slashdot. *Proc. Conference on Human Factors in Computing Systems (CHI)*, pages 1253–1262, 2007.

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

[99] Kevin Wise, Brian Hamman, and Kjerstin Thorson. Moderation, response rate, and message interactivity: Features of online communities and their effects on intent to participate. *Journal of Computer-Mediated Communication*, 12(1):24–41, 2006.

[100] N. J. Stroud, J. M. Scacco, A. Muddiman, and A. L. Curry. Changing deliberative norms on news organizations' facebook sites. *Journal of Computer-Mediated Communication*, 2014.

[101] S. O. Sood, E. F. Churchill, and J. Antin. Automatic identification of personal insults on social news sites. *Journal of the American Society for Information Science and Technology*, 63(2):270–285, 2012.

[102] Annie Louis and Ani Nenkova. What makes writing great? first experiments on article quality prediction in the science journalism domain. *Transactions of the Association for Computational Linguistics*, 1:341–352, 2013.

[103] Emily Pitler and Ani Nenkova. Revisiting readability: a unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 186–195, 2008.

[104] Dirk Brand and Brink Van Der Merwe. Comment classification for an online news domain. In *Proceedings of the International Conference on the Use of Mobile Informations and Communication Technology*, 2014.

[105] Nayer Wanas, Motaz El-Saban, Heba Ashour, and Waleed Ammar. Automatic scoring of online discussion posts. In *Proceedings of the ACM Workshop on Information Credibility on the Web*, 2008.

[106] Chiao-Fang Hsu, E. Khabiri, and J. Caverlee. Ranking comments on the social web. In *Proceedings of the IEEE Conference on Computational Science and Engineering*, pages 90–97, 2009.

[107] Karin Wahl Jorgensen. Understanding the conditions for public discourse: four rules for selecting letters to the editor. *Journalism Studies*, 3(1):69–81, 2002.

[108] Kathleen McElroy. Where old (gatekeepers) meets new (media). *Journalism Practice*, 7(6):755–771, 2013.

[109] Bill Reader. Air mail: NPR sees "community" in letters from listeners. *Journal of Broadcasting and Electronic Media*, 51(4):651–669, 2007.

[110] N. Diakopoulos. The editor's eye: Curation and comment relevance on the New York Times. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 2015.

[111] Saeideh Bakhshi, artha Kanuparthy, and David A. Shamma. Understanding online reviews: Funny, cool or useful? In *Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 1270–1276, 2015.

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

[113] Enamul Hoque and Giuseppe Carenini. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pages 169–180, New York, NY, USA, 2015. ACM.

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

[115] Siamak Faridani, Ephrat Bitton, Kimiko Ryokai, and Ken Goldberg. Opinion space: a scalable tool for browsing online comments. *Proc. Conference on Human factors in Computing Systems (CHI)*, pages 1175–1184, 2010.

[116] Michael Sedlmair, Miriah D. Meyer, and Tamara Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[117] Dan M. Brown. *Communicating Design*. New Riders, 2010.

[118] Nicholas Diakopoulos, Stephen Cass, and Joshua Romero. Data-driven rankings: The design and development of the ieee top programming languages news app. In *Proceedings of the Symposium on Computation + Journalism*, 2014.

[119] G. H. McLaughlin. SMOG grading - a new readability formula. *Journal of Reading*, 12(8):639–646, 1969.

[120] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics*, 2013.

[121] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[122] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[123] James W Pennebaker, Martha E Francis, and Roger J Booth. *Linguistic inquiry and word count: LIWC 2001*. Lawrence Erlbaum Associates, 2001.

[124] Peter Sheridan Dodds, Kameron Decker Harris, Isabel M Kloumann, Catherine A Bliss, and Christopher M Danforth. Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. *PLOS ONE*, 6(12):e26752, 2011.

[125] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[126] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41:391–407, 1990.

[127] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[128] Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. Two/too simple adaptations of Word2vec for syntax problems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, 2015.

[129] Wang Ling, Yulia Tsvetkov, Silvio Amir, Ramon Fermandez, Chris Dyer, Alan W Black, Isabel Trancoso, and Chu-Cheng Lin. Not all contexts are created equal: Better word representations with variable attention. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1367–1372, 2015.

[130] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[131] Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of the International Conference on Computational Linguistics*, pages 151–160, 2014.

[132] Seungyeon Kim, Joonseok Lee, Guy Lebanon, and Haesun Park. Local context sparse coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2260–2266, 2015.

[133] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*, pages 1188–1196, 2014.

[134] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013.

[135] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

[136] M. Berger, K. McDonough, and L. M. Seversky. cite2vec: Citation???driven document exploration via word embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):691???–700, Jan 2017.

[137] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.

[138] Ethan Fast, Binbin Chen, and Michael S Bernstein. Empath: Understanding topic signals in large-scale text. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 4647–4657, 2016.

[139] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[140] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Journal Computational Linguistics*, 37(2):267–307, 2011.

[141] Seungyeon Kim, Joonseok Lee, Guy Lebanon, and Haesun Park. Estimating temporal dynamics of human emotions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 168–174, 2015.

[142] Seungyeon Kim, Fuxin Li, Guy Lebanon, and Irfan Essa. Beyond sentiment: The manifold of human emotions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 360–369, 2013.

[143] Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X. Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. TIARA: a visual exploratory text analytic system. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, pages 153–162, 2010.

[144] Wenwen Dou, Xiaoyu Wang, Remco Chang, and William Ribarsky. ParallelTopics: A probabilistic approach to exploring document collections. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pages 231–240, 2011.

[145] Weiwei Cui, Shixia Liu, Li Tan, Conglei Shi, Yangqiu Song, Zekai Gao, Huamin Qu, and Xin Tong. TextFlow: Towards better understanding of evolving topics in text. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2412–2421, 2011.

[146] Edmund M Talley, David Newman, David Mimno, Bruce W Herr II, Hanna M Wallach, Gully APC Burns, AG Miriam Leenders, and Andrew McCallum. Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods*, 8(6):443–444, 2011.

[147] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*, pages 74–77, 2012.

[148] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. ivis-clustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, 31(3pt3):1155–1164, 2012.

[149] Shuo Chang, Peng Dai, Lichan Hong, Cheng Sheng, Tianjiao Zhang, and Ed H Chi. AppGrouper: Knowledge-based interactive clustering tool for app search results. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 348–358, 2016.

[150] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[151] Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. UTOPIAN: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013.

[152] Bum Chul Kwon, Sung-Hee Kim, Sukwon Lee, Jaegul Choo, Jina Huh, and Ji Soo Yi. VisOHC: Designing visual analytics for online health communities. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):71–80, 2016.

[153] Ian T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[154] Robert M Entman. Framing: Toward clarification of a fractured paradigm. *Journal of Communication*, 43(4):51–58, 1993.

[155] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann. Visual text analysis in digital humanities. *Computer Graphics Forum*, pages n/a–n/a, 2016.

[156] Deok Gun Park, Sung-Hee Kim, and Niklas Elmqvist. Gatherplots: Extended scatterplots for categorical data. Technical Report HCIL-2016-10, University of Maryland, College Park, 2016.

[157] George A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[158] Bum Chul Kwon, Hannah Kim, Emily Wall, Jaegul Choo, Haesun Park, and Alex Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, 2017.