VISUALIZATION OF TEXTUAL CONTENT FROM SOCIAL MEDIA AND ONLINE COMMUNITIES

A Thesis
Presented to
The Academic Faculty

by

Mengdie Hu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology
May 2018

VISUALIZATION OF TEXTUAL CONTENT FROM SOCIAL MEDIA AND ONLINE COMMUNITIES

Approved by:

Dr. John T. Stasko, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Rahul C. Basole
School of Interactive Computing
Georgia Institute of Technology

Dr. Duen Horng (Polo) Chau
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Alex Endert
School of Interactive Computing
Georgia Institute of Technology

Dr. Michelle X. Zhou
Co-founder & CEO
Juji, Inc.

Date Approved: Jan 4, 2018

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Motivation

As activities on social media and online communities increase explosively, users on these platforms produce a new type of text document. Examples include Tweets, Facebook posts, YouTube comments, and Yelp reviews. These text documents are authored by numerous ordinary people and encode authentic opinions on various topics. Many of them are immediate reactions to time-sensitive subjects. Most of these text documents are free and open for the public to browse, and many platforms even offer APIs (application programming interface) for programmatically querying the data on their sites. The unprecedented abundance and the free and easy access make these text documents a goldmine for people interested in understanding the opinions and emotional state of the masses.

One example consumer of social media documents is the branding and customer relationship manager. Before the era of social media, customer service engaged with users through private channels such as phone calls and emails. Today, many consumers prefer to post about their experience publicly through review sites or social media. Given how quickly these stories can pick up volume and gain attention through up-voting and sharing, experts say that brands must engage with these consumer-generated brand stories to ensure their success in the marketplace [36]. They suggest a strategy of stimulating and promoting positive consumer-generated brand stories, while quickly addressing negative consumer-generated brand stories [36]. To accomplish this, brands need to closely monitor different channels on social media and quickly identify, understand, and react to relevant posts.

Another example consumer of social media documents is a journalist. Journalists often want to report how the public feels about a topic or event. This is traditionally done through surveys, interviews, and on-site reporting, which are limited in scale and take a lot of time and effort to conduct. With the rise of social media, journalists can easily access large numbers of personal and spontaneous accounts in real-time. They can also use social media as a platform to identify posts to cite in their reports and recruit subjects for further in-depth interviews.

Although the data from social media and online communities are relatively easy to access, gaining insights from them is not always easy. The vast scale and the fact that these documents are usually in free-form text present an interesting paradox for those trying to analyze them: on the one hand, the vast scale means the reader gets many, many data points. On the other hand, it is very time-consuming to go through all the posts. Similarly, the free text form means people can express what's on their mind without any restraints. But without any guidance, much of what they post will be useless to the analysts. In other words, the increased volume and spontaneity is both a blessing and a curse.

Many researchers have sought to uncover the social dynamics behind a social media text collection by studying meta-information such as the network structure or descriptive characteristics such as activity volume fluctuations [57, 30]. This is very useful knowledge, but only part of the puzzle. To uncover equally valuable information such as the opinions and emotional states of masses requires a deep delve into the textual content itself.

Given the scale of the data, it is near impossible to have human analysts manually read through every single post. Two primary solutions help analysts get useful information without reading the entire dataset:

1. Show a sample of most representative posts.

2. Show some kind of aggregation (e.g., keywords) of the textual content of the dataset.

In order to select a sample of posts to show, some kind of criteria needs to be provided. The easiest criteria to implement is probably picking the posts that are shared most often. The obvious problem with this approach is it will only highlight the most popular posts from a few popular outlets, defeating the purpose of gathering diverse information (see this issue demonstrated by the Osama Bin Laden study in Section 4.2.8). To address this, one can first summarize high-level categories and display the most popular posts from each category. Both projects in my thesis work follow this idea.

Solution 2 is also widely used, and the most popular aggregation type is frequently mentioned words and phrases. For example, review sites such as tripadvisor.com often show a short list of keywords and phrases extracted from reviews. Topic modeling results also can display a list of representative words for each topic. Since people cannot always make sense from keywords and phrases, they are often used as a navigational tool to direct readers to posts that interest them. In other words, one can combine solution 2 and 1 to help people discover representative posts to read.

While algorithmic solutions exist for both categorizing topics and selecting popular posts, they still face challenges that make the accuracy rarely acceptable to human users. The accuracy issue is both due to the natural complexities of human languages and to the fact that machine learning algorithms usually require intensive training with domain-specific data [73, 77]. For example, sentiment analysis is a popular topic in the natural language processing (NLP) research community. For even the most straightforward task of assigning polarity of sentiment, computer programs still face challenging issues such as understanding negation and sarcasm which makes it hard for even the best techniques to reach high accuracy [59]. On top of the challenges to understanding natural language, social media text brings its own set of issues:

compared to traditional text documents (e.g., book chapters, news articles) a social media text collection typically contains much more redundant information and much more informal and unexpected use of the language. So it is harder to train the computer program, and the noise-to-information ratio in social media text is higher than that in traditional text collections.

As user studies have found [95], inaccuracies by algorithms make users lose confidence in the algorithms and question if they can trust the results presented to them. So one challenge I seek to address in this thesis is user trust for computational results. Also, social media analysis is very new and attracts the interest of not only researchers but also casual Internet users. The analytic tasks are often exploratory with no clear goal or hypothesis. While computer programs might be good at finding answers, they are not known for asking good questions. So another issue is how to support open-ended exploratory analysis with social media text.

Given all the challenges with pure algorithmic solutions, a natural course of action is to involve a human analyst. Humans can raise questions, use the computational resources to research these questions, and make the final decision. Interactive data visualization systems provide an interface between the human analyst and the algorithm. They are designed with the goal of presenting the data in a way that is easy for the human to consume and ask important questions, while leveraging computational algorithms for analytical tasks that are too complicated for the human brain. In this thesis, I explore ways to apply information visualization techniques to facilitate analysis of text from social media and online communities.

1.2 Research method

My work answers the following research questions:

R1: What are the data characteristics and typical analytic tasks associated with text documents from social media and online communities?

R2: In order to best leverage the strength of both computational analysis and human judgment, can we identify tasks or aspects of tasks in social media analysis that are better suited to the human analyst than algorithms? What are limitations to algorithmic solutions and can they be addressed by the human analyst?

R3: How do we use interactive data visualization to address tasks or aspects of tasks identified in R2? What are efficient ways to visually communicate computational analysis results to the human viewer and for the human to interact with the views in order to gain insights about the underlying text data?

By answering R1, we gain a better understanding of how the data and analytic tasks for social media text differ from those for traditional text documents. This knowledge helps us identify existing tools that can be applied to social media data analysis as well as the missing pieces. Before trying to build tools to support new data and tasks, it is vitally important that we answer R2, so we know the strength and weakness of both human and machine and find the best leverage between them. R3 is answered by designing and prototyping visualization systems that support effective interactions between the human analyst the the computer program. We design novel visualization metaphors that best communicate the text analysis results to the human, and novel interaction techniques that use human feedback to improve the system.

The full spectrum of social media text documents spans many domains, document types, and tasks associated with these materials. It is impossible to iterate through all document types and tasks or build prototypes for each domain. For my thesis, I focus on two domains and try to answer the three research questions for each domain. The two domains are consumer reviews as represented by Amazon reviews and social media documents as represented by Twitter posts. I argue that not only are both domains very important, but they also represent some extreme characteristics of text documents produced by social media and online communities. Understanding these

two domains can help us create tools that apply to other domains that share similar characteristics.

Table 1: This table highlights differences between consumer reviews and social media posts.

	Consumer reviews	Social media posts
Length	long (paragraphs)	short (sentences)
format	guided	free
Language	relatively formal	usually very informal
Spontaneity	relatively deliberated	relatively spontaneous
Analytical tasks	well-defined	exploratory

Table 1 highlights the differences between consumer reviews and social media posts. Among documents produced by social media and online communities, consumer reviews are relatively similar to traditional text documents: they are usually structured on a given topic (the reviewed product or experience), and they usually contain well-formed sentences since their audience is other consumers and the merchant. The high-level goal for reading consumer reviews is also relatively well-defined, namely, understand the pros and cons of the product in order to make a purchasing decision. In my thesis, I describe different tasks for supporting this goal, outline the design considerations based on the tasks, implement a working prototype system called OpinionBlocks with novel block-like visual metaphors for opinion snippets, and perform a user study to find the values and limitations of the design.

The second study is in the domain of social media text such as tweets. The high-level goal of social media analysis is usually not well-defined, and it is even less clear what tasks should be performed. In my thesis, I describe a case study on an interesting social media news spread phenomenon. I walk through the steps involved in a real exploratory analysis scenario and the different tools I build for each task. Based on the knowledge gathered from the study, I identify content visualization as a crucial piece that was missing from existing tools, and I design a new visualization technique

called SentenTree for open-ended exploration of social media text. I implement a prototype of the system, and through user study found out about the use cases and limitations of this new technique.

Based on my research in both domains, I show that reading the original text document is critically important to analytic tasks because analysts rely on the original text document to gain insights into complex concepts, confirm hypotheses, and provide evidence to substantiate their claims. Instead of trying to eliminate reading of the original text by providing results through algorithms, I propose that we should build systems that help people organize the documents and identify the most interesting ones to read. I further summarize three critical steps that help analysts identify useful document to read, and highlight the importance of addressing errors from NLP algorithms.

1.3 Thesis Statement

Interactive visualization can bridge the gap between human analysts and large quantities of text from social media and online communities by using computational methods to organize the text document in a way that allows analysts to quickly form impressions, discover interesting facets, and locate relevant documents to read.

1.4 Contributions

In this thesis, I explore design principles for interactive visualizations that facilitate analysis of large quantities of text documents from social media and online communities. I focus on two domains of text: consumer reviews and social media posts. I summarize the data characteristics associated with text documents from social media and online communities and challenges they bring. I also explore the typical analytic tasks in each domain and propose that a critical task for a computational system to support exploratory analysis is to help analysts identify useful documents to read. I further break down this task into three key steps that a visualization system should

support: 1) form impressions, 2) discover interesting facets, and 3) locate relevant documents.

I also present two prototype visualization systems, OpinionBlocks for consumer reviews, and SentenTree for social media text. Both systems were designed following the outlined principles. Both contain novel visualization metaphors. SentenTree, in particular, can be adapted to be used in many other text visualization systems. User studies with these systems demonstrate their effectiveness in helping analysts organize and find documents useful for their workflow.

1.5 Organization

This thesis document is organized in the following manner: In Chapter 2, I discuss related work on text analysis visualization and techniques. Then I describe my research projects in the domain of consumer reviews (Chapter 3) and social media (Chapter 4). For each project, I describe a study of the domain data and tasks, the design rationale and the implementation details of my visualization prototype. Then I explain what I learned from testing my prototype with users. In the final chapter (Chapter 5) I summarize my overall contributions to text visualization for social media and online communities.

CHAPTER II

RELATED WORK

In this section, I review existing work on text visualization techniques and systems, with a focus on those that apply to social media text. The body of work is very large, so I organize the sections around major analytic tasks. Each task is usually associated with domain data from certain sources and of certain scale. I order the sections by the scale of associated text data in decreasing size. Text visualization also inevitably involve natural language processing (NLP) and other data analysis techniques from other fields such as information retrieval and machine learning. Since the focus of this thesis is on advancing data visualization techniques, I will not comprehensively review text analysis techniques from fields outside of visualization. Instead, as I review each visualization task and technique, I will point out which data analysis techniques are used in combination with the visualization, why they are valuable, and what their limitations are. Also note that many text visualization systems use a suite of text analysis and visualization methods to examine the data at multiple angles, so these systems might be covered in multiple parts in this section. Due to the scope of this thesis, I try to summarize the work in each section briefly. For a more comprehensive review, refer to Cao and Cui's book chapter on Text Visualization [19], which is based on papers collected in the Text Visualization Browser [53] up to 2016.

2.1 Mapping topics

When trying to gain a high level view into the entire corpus in a domain or subject, a common task to to create a "landscape" of topics or concepts, how these topics or concepts relate to each other, and which documents mention which topics or concepts. Visualizations often map concepts and documents to a 2D/2.5D space and utilize

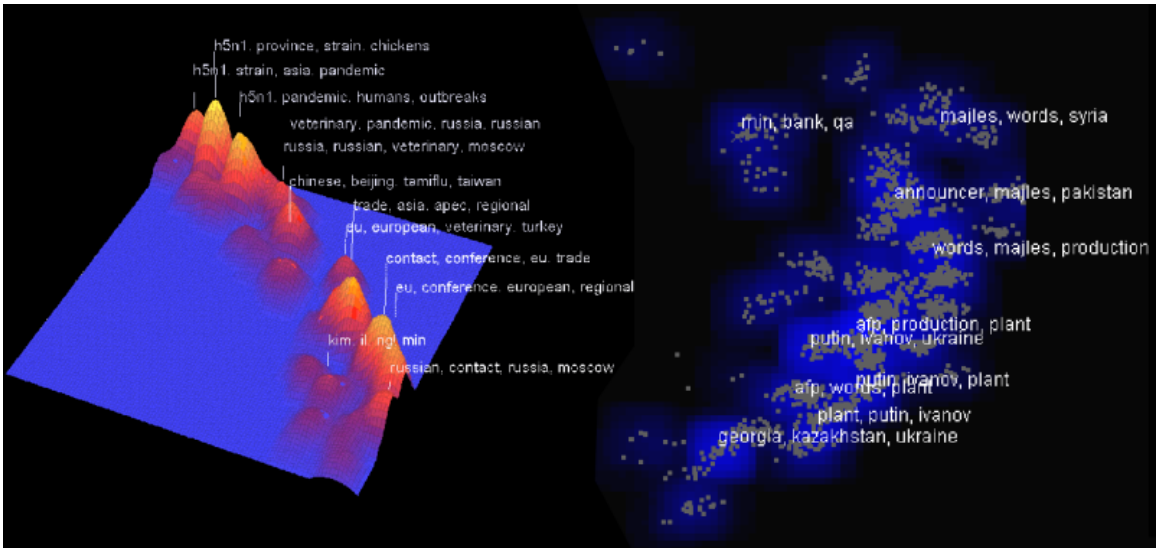


Figure 1: ThemeView and Galaxy View from IN-SPIRE. Image from `http://vacommunity.org/IN-SPIRE`

spatial proximity to imply relationships between concepts and documents. And by examining the size of related document cluster and sometimes color intensity the user is able to tell the popularity of a topic or concept.

Notable visualization systems include SPIRE/IN-SPIRE [87], as shown in Figure 1. On the left is a ThemeView, which displays major clusters of concepts in a 2.5D terrain map. On the right is a galaxy metaphor with each star representing a document and the spatial layout of the stars representing relevance between documents. The Stanford Dissertation Browser (Figure 2) focuses on a single academic department at a time by placing it at the center of a radial layout. It shows the relative similarity between this centered department and all other departments at Stanford by the distance between them [24] .

Visualizations that map the major topics and concepts in large and diverse document collection is used to provide a high-level summary of the different subject areas within a domain. They focus on highlighting the similarity or connections between subject areas, instead of showing details about each subject. In order to cluster

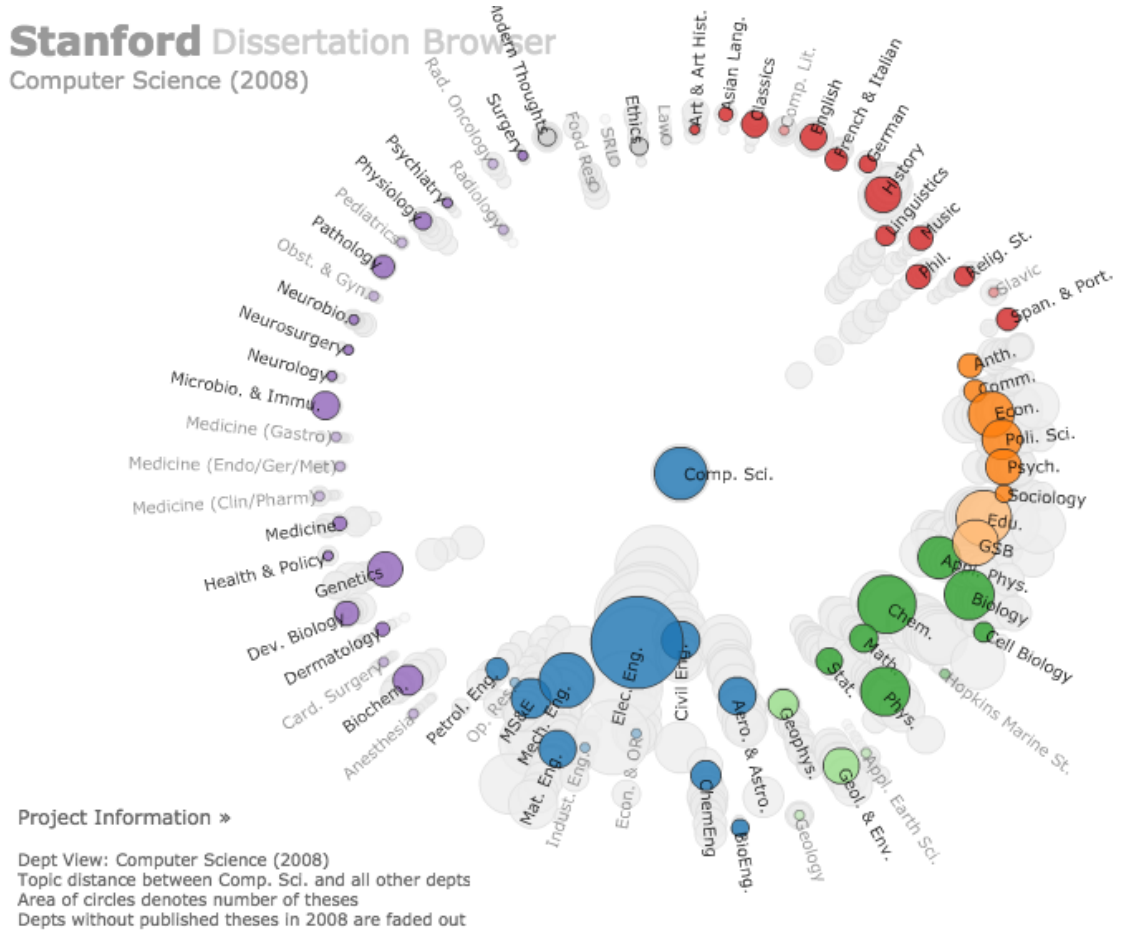


Figure 2: Stanford Dissertation Browser. Screenshot taken from `http://nlp.stanford.edu/projects/dissertations/browser.html`

documents and extract subject areas, complex clustering and language modeling techniques are applied. Dimensionality reduction is often used to assign a spatial position for each subject based on many dimensions of data.

2.2 Linking entities

Similar to the the previous group of visualization techniques, visualizations discussed here also focuses on showing relationships between concepts and documents. However the concepts in question are concrete lower-level entities such as names and locations. And instead of grouping a lot of documents into clusters, a single documents is often considered as its own unit. These visualizations help analysts uncover hidden

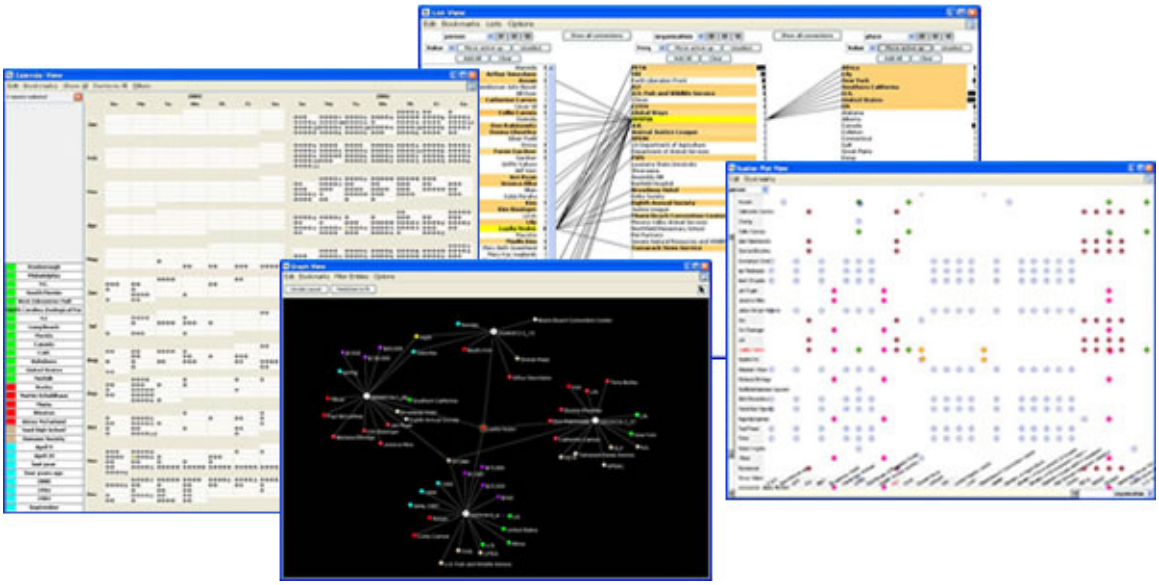


Figure 3: Selected views from Jigsaw. Image from `http://www.cc.gatech.edu/gvu/ii/jigsaw/`

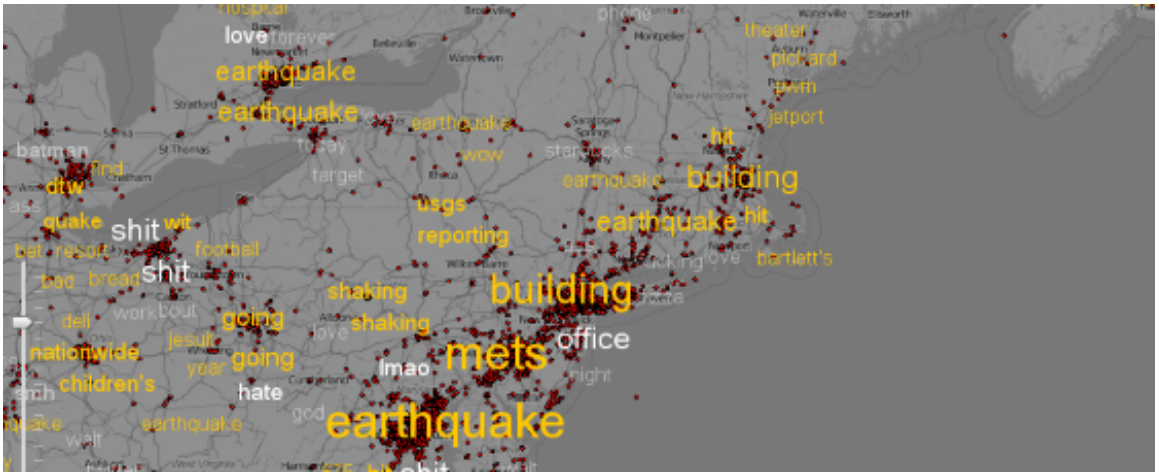


Figure 4: Scatterblogs shows the major concepts for each geo-location on a map. Image from `https://www.scatterblogs.com/en/`

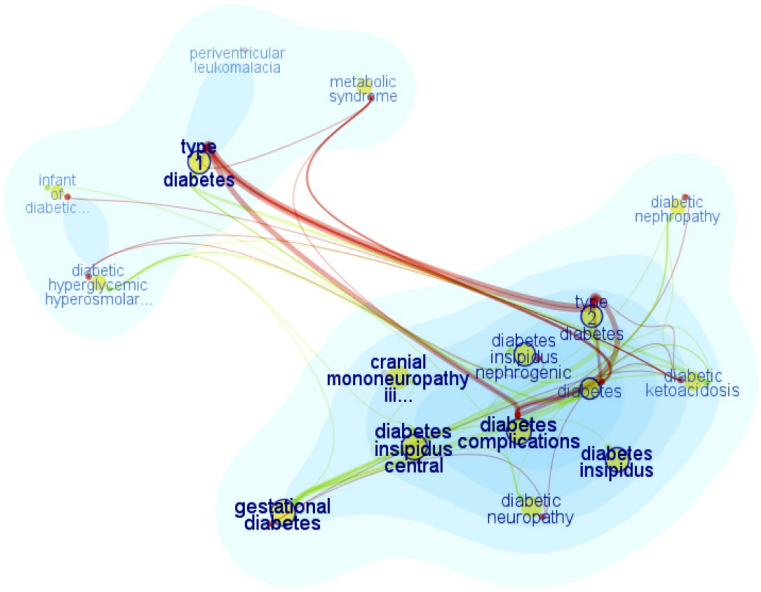


Figure 5: FacetAtlas. Image from `http://gotz.web.unc.edu/research-project/facetatlas/`

relationships between different entities by examining their connections at different dimensions or facets. It also connects the dots between different documents through shared entities and direct the analyst to useful documents in the collection. These visualizations are often used in investigative analysis scenarios.

Notable examples include Jigsaw [76], a multi-view system for intelligence analysis that focuses on entity co-occurrence within the same document. Figure 3 shows some of the most important views of Jigsaw, including the Graph View that links important entities and allows the analyst to expand from one entity to its connections, and the List View that puts entities in parallel lists based on their type and highlight their inter-type connections between entities as well as entity-to-document connections. Scatterblogs (Figure 4) is a system that extracts keywords from social media text and overlays them on a map to show the origins of the text, which can be used for social media alert and monitoring [17]. FacetAtlas [21] is another visualization system that looks for complex multifaceted entity relationships. It uses spatial proximity and density maps to show cliques of concepts, and edges to show relationships between

concepts in different cliques (Figure 5).

Named entity extraction is a key data analysis techniques used in many of these systems. Based on the characteristics of each entity, custom views are also designed. For example Jigsaw (Figure 3) has a calendar view for dates, while Scatterblogs (Figure 4) shows locations on a map. FacetAtlas performs topic modeling before named entity extraction to uncover hidden "facets" in topics and documents [21].

2.3 Temporal topical change

Many text visualizations focus on the temporal changes of topics. The most common applications of these visualization include studying topical popularity and influence over time, and detecting event spikes. Since time is crucial for temporal visualization, almost all existing work display time on the x-axis.

The arguably most popular visual metaphor used for this task is the ThemeRiver [38]. As shown in Figure 6, ThemeRiver uses a stacked graph to encode topic volume changes over time. MemeTracker [57], TIARA [86], TextFlow [27], and Opinion-Flow [90] are just a few of the many visualizations based on the ThemeRiver technique. Most of them employ advanced topic-modeling techniques to detect different topic themes and some even detect and build hierarchical topic trees for topical changes over time [90]. MemeTracker [57] on the other hand detects short, distinctive phrases and their variations from news articles and blogs, and clusters a phrase and its variations into a "meme". The visualization successfully demonstrates that the life circle of memes follow a pattern of spiking and diffusion (Figure 8).

The original design of ThemeRiver displays volume changes but not details about each topic. It shows a label next to each topic and annotates major events on the timeline (Figure 6). Similarly, MemeTracker also manually label each meme next to its stacked bar. On the other hand, systems such as TIARA (Figure 7) that rely on topic modeling to generate topics usually have a set of representative words for each topic

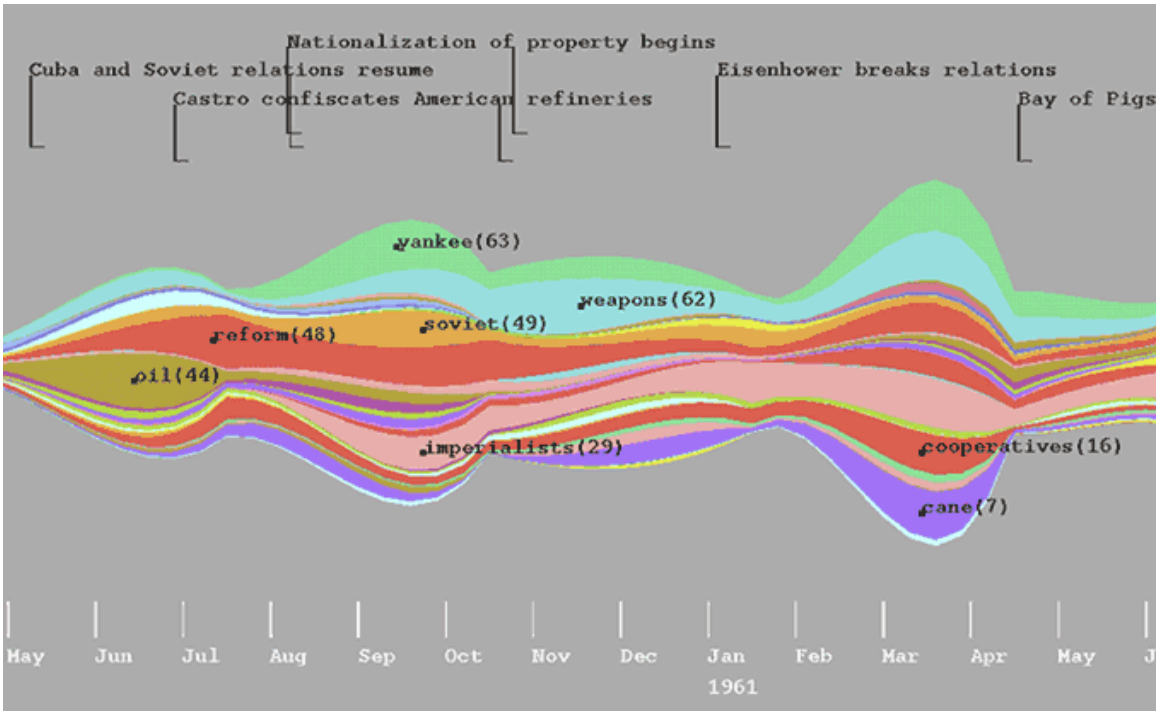


Figure 6: The original ThemeRiver visualization. Image from `http://vis.pnnl.gov/research_themeriver.stm`

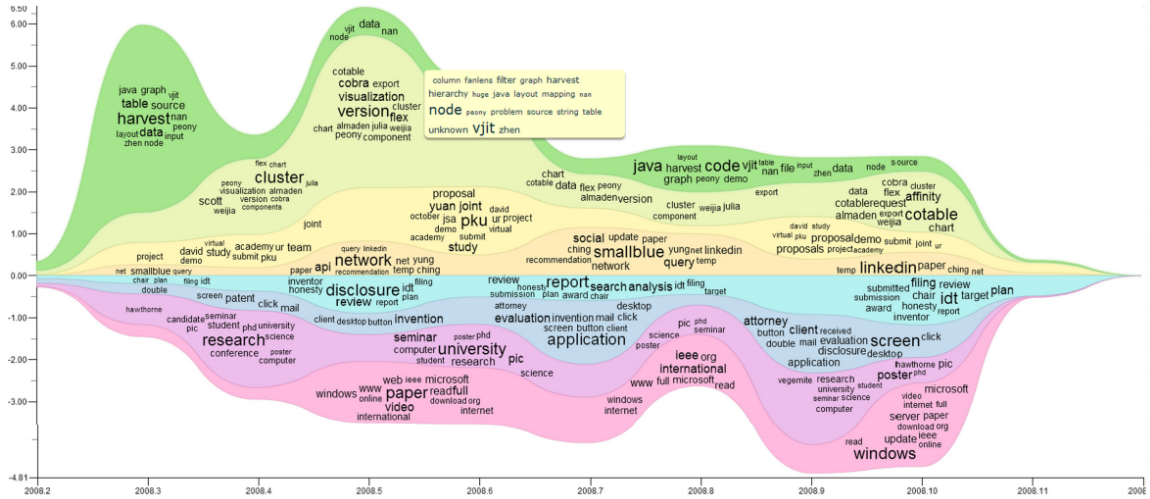


Figure 7: TIARA, a text visualization system based on latent Dirichlet allocation for topic modeling [86].

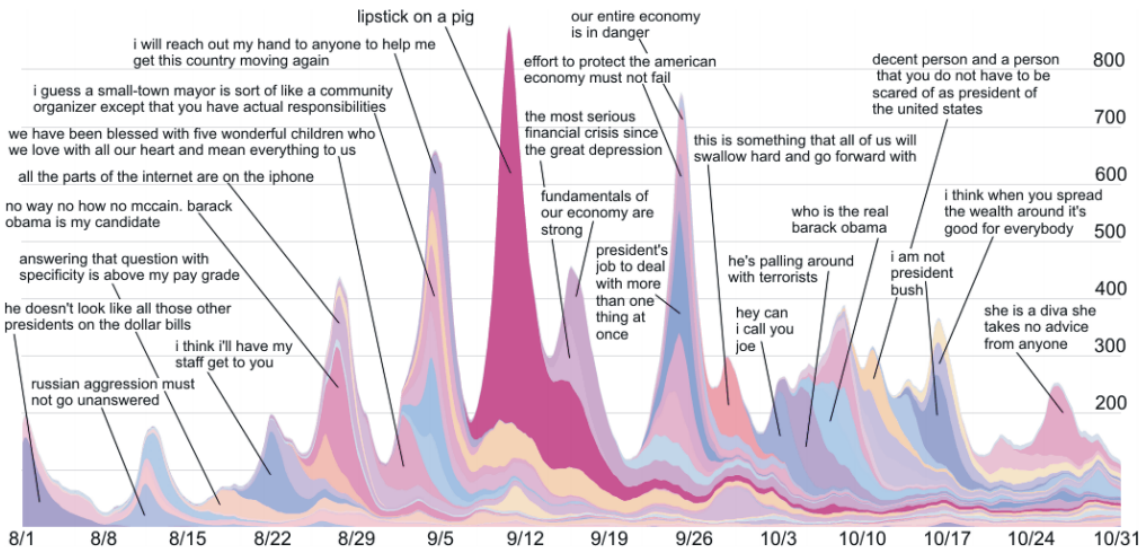


Figure 8: ThemeRiver-based visualization of memes in the news circle [57].

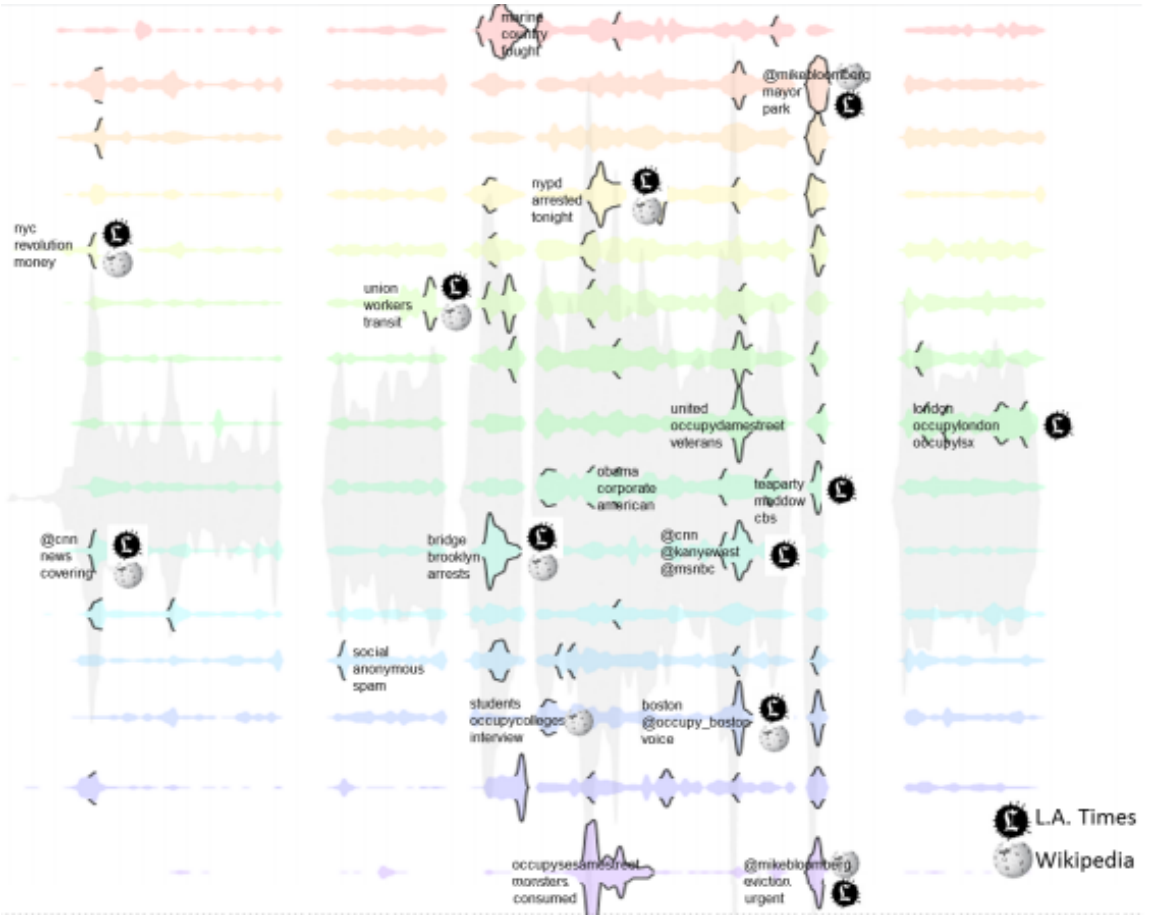


Figure 9: Leadline highlights event bursts in streams of topics and shows related entities next to each burst [30].

at different time points. They often fill the space of the stacked bars in ThemeRiver by overlaying a word cloud on top of the bars to show the most representative words for each topic at a given time. This is an efficient use of space and gives the viewer a quick summary of the content of each topic.

A number of temporal visualization systems focus on detecting event spikes. EventRiver [62], Leadline [30], and Cloudlines [51] all use bubble-like visual elements to show event bursts detected from text streams. These visualizations also focus on the changing shape of the event streams rather than details about each topic. Some of them (Figure 9) annotate a few keywords next to each topic or event to provide context. Event visualization systems generally integrate topic modeling, event detection, and named entity recognition techniques.

Finally, a few visualizations display a snapshot of the content of topics at different time points. By comparing the difference between snapshots the viewer can easily tell how the content changed over time. Parallel Tag Clouds [26] displays a vertical word cloud for each facet (e.g. time) of the data, and links related words in different facets, forming a parallel coordinates-like visualization (Figure 10). Though Parelle Tag Clouds is not specifically designed to show temporal changes, when the facet is time it can be very effective in tracking and comparing keyword changes in documents. Story Tracker [52] clusters news stories and displays a set of keywords for each news story at a different time to show how the news stories evolve over time.

2.4 Information diffusion

Somewhat related to topical changes is work on information diffusion in social networks. Most visualizations for information diffusion are for story-telling purposes and focused on showing the diffusion networks with very limited text visualization component (e.g. Google+ Ripples as shown in Figure 11), a native visualization using a mix of circular tree map and node-link diagram [81] and Whisper which tracks

Figure 10: Parallel Tag Clouds link the same word across document sets and show how its ranking changed over time [26].

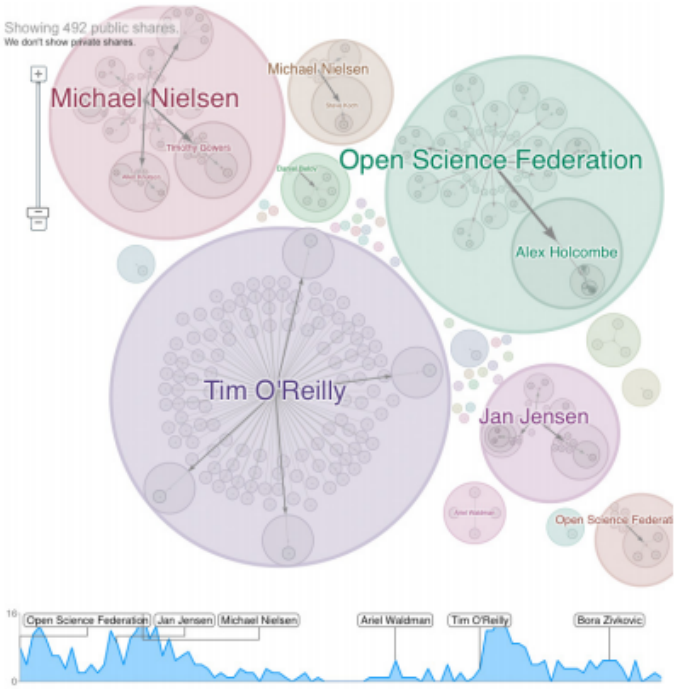


Figure 11: Google+ Ripples uses a mix of node-and-link and circular treemap metaphors to show patterns of sharing for a given topic [81].

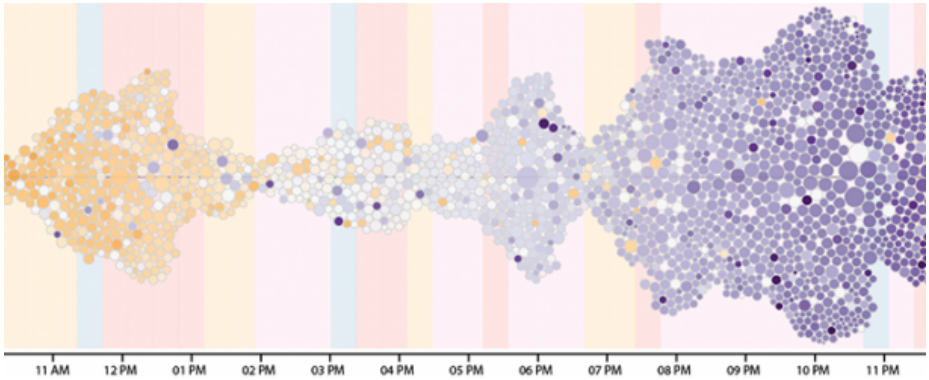


Figure 12: FluxFlow visualizes anomalous information spreading on Twitter [96].

location of retweets [20]). TextWheel is a fun story-telling visualization consists of a "wheel" of keywords and a "transportation belt" of social media documents that play out the news stream. MemeTracker [57] which I discussed in an earlier section is sometimes seen as depicting the diffusion and evolution of viral information online.

One important application combining information diffusion research and text analysis is the detection of viral misinformation. Due to challenges associated with classification misinformation, work in this field has been limited. FluxFlow [96] stands out as an interactive visualization that combines advanced machine learning algorithms to detect anomalous tweets and interactive visualizations that help the human analyst understand the detected information. The main visualization is a bubble-like view similar to EventRiver [62], Leadline [30], and Cloudlines [51] with an embedded circular treemap to show individual tweets (Figure 12).

2.5 Opinion mining

Opinion mining is a well-published field with researchers from information retrieval, data visualization and user interface design, etc. The most popular application of opinion mining is to consumer review analysis. Usage of consumer review analysis includes helping consumers make informed decisions and helping manufacturers understanding consumer feedback. Some good examples include Review Spotlight, which presents a word-cloud summary of online reviews in noun-adjective pairs [95], and color-code noun-adjective pairs based on the perceived sentiment (Figure 13). Carenini and Rizoli built a multimedia interface that facilitates the comparison of different reviews [22]. More recently, Huang et al., presented RevMiner, an interactive system that sum-marizes reviews in noun-adjective pairs to be presented in a compact mobile phone interface [45]; and Rohrdantz et al., designed a visualization system that supports feature-based sentiment analysis of time-stamped review documents [71]. More recently, Xu et al. built a visualization system focused on

highlighting controversy in consumer reviews by extracting aspects with conflicting sentiment and displaying a word cloud with keywords from both sides for each aspect [92]. The general take-away from these systems is that user care deeply about the major aspects mentioned in the reviews, as well as the overall sentiment towards each aspect. Displaying noun-adjective pairs is a very efficient way to communicate aspects and related sentiment to the user.

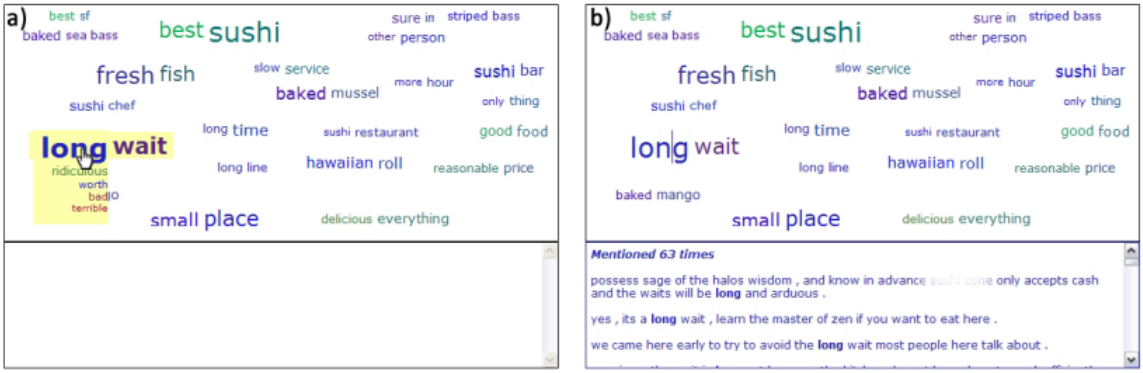


Figure 13: Review Spotlight summarizes restaurant reviews in noun-adjective pairs, and uses color to highlight positive and negative sentiment. It also allows the user to click on a noun-adjective pair and bring up the original review text [94].

Beyond consumer reviews, visualization systems have also been developed for opinions on public discourse, etc. For example, Faridani et al. created Opinion Space, an interactive tool that allows users to visualize and navigate collected opinions [32] with a focus on surfacing opinion polarity.

Major text analysis techniques used in opinion mining include part of speech analysis to extract the noun phrases that represent important topics or facets, and adjectives that describe the sentiment towards the topic or facet. Sentiment analysis based on labeled dictionaries is also widely used to understand opinions. The user study of Review Spotlight shows that [95] issues with NLP analysis, especially wrongly assigned sentiment, have a very negative role on users' trust towards the visualization system.

2.6 Content visualization based on words

Most of the tasks mentioned so far focus on providing a summary of the corpus by clustering the text document into topics, themes, facets, etc. As discussed in previous sections, many visualizations uses a "main" view to show the topics, volume changes, and so on, but also want to overlay a visualization of the actual text content.

Tag cloud/word clouds are arguably the most widely used visualization method to display content of text documents. Popularized by websites such as Flickr, the original tag clouds list frequently used tags and vary their font sizes according to their usage frequency [83]. As people started to apply the visualization technique to other text documents, the name "word cloud" and sometimes "text cloud" are used interchangeably with "tag cloud". We will use "word cloud" to refer to this visualization technique from now on.

Wordle [82] provides an algorithm to automatically generate compact layout and aesthetically pleasing visual encodings. Wordle is very popular with the casual user for non-analytics tasks. For analytics tasks, word clouds are often used in combinations with other visual metaphors to provide some context or detail. As we discussed while reviewing some of the earlier visualization techniques, many visualizations displaying topic and topical changes often overlay a word cloud on top of the "main" topic visualizations. Given that many systems use topic modeling to cluster documents into topics, and topic modeling results in representative words for each topic, word cloud is a natural visual metaphor to use for such systems.

Despite their popularity, word clouds have been found to be lacking for analytical tasks [83]. Hearst and Rosner [39] pointed out three major problems with word clouds: 1) the size encoding is not accurate due to different word length, which makes it hard to compare words, 2) the physical layout is not meaningful, therefore words appear in discrete form and there is no context next to them, and 3) there is no natural "flow" for reading, the viewer just looks at random words in the visualization. Supporting

the final two points, Yatani et al. found that humans tend to verbalize opinions in short expressions rather than discrete words [94].

A number of projects try to extend word clouds and address some of these concerns. Chuang et al. [23] applied natural language processing techniques to improve word selection. Multiple research efforts in semantic-preserving word clouds try to address the second problem by positioning words/terms closely to other related ones. Context-preserving word clouds [28], Seam-carving word clouds [91], ProjCloud [69], and ReCloud [84] all pursue this approach in different ways (see example in Figure 14). In general, they focus on clustering related words together and providing compact visual representations of the word collections. Barth et al. [15] introduced three additional semantic word cloud algorithms and argued for their improvement over existing algorithms along visual and performance qualities: realized adjacencies, compactness, uniform area utilization, distortion, and aspect ratio and running time.

More recently, Felix et al. argued [33] that existing studies have not been holistic in exploring the design spaced based on analytic tasks. They designed four studies to benchmark visual parameters of word summaries based on four common analytic tasks: magnitude judgment, keyword search, topic matching and topic discovery. They found the effectiveness of different visual parameters vary greatly by task, and there were no winning visual design for all tasks.

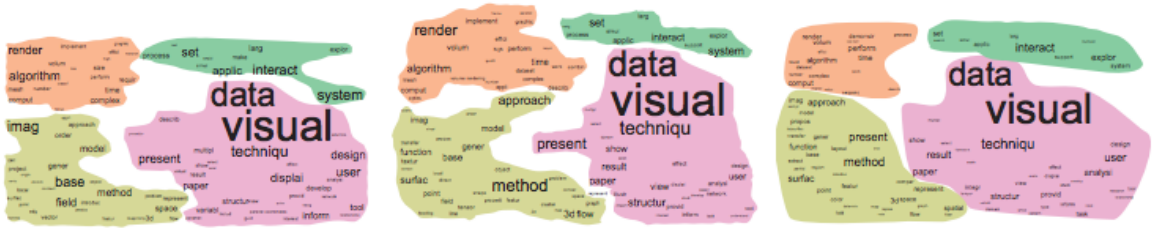


Figure 14: Semantic-preserving word clouds of the IEEE Vis/InfoVis paper abstracts at 1999, 2005, and 2010 [91].

2.6.1 Content visualization based on context and structures

Besides text cloud inspired visualizations, other techniques have been developed to preserve some of the original text structures. A notable example is the Word Tree [85], which lets the user select a word of interest, and displays the sentence segments next to that word by building and visualizing a prefix tree with the selected word at the root (Figure 15). Therefore, the user learns about the word in the context of sentences. Double Tree [29] extends this technique by building Word Trees on both sides of the word of interest. Wordgraph [70] allows query by wildcard and displays not only Word Trees on both sides of the query pattern but also branches in between query terms (Figure 16). The Word Tree and related visualizations are great tools to show context next to a point of interest in the text data, but they lack the ability to provide an overview of the dataset without a focal point. The user also needs have some knowledge of the data before hand to know what query to use, otherwise the visualizations cannot give meaningful results.

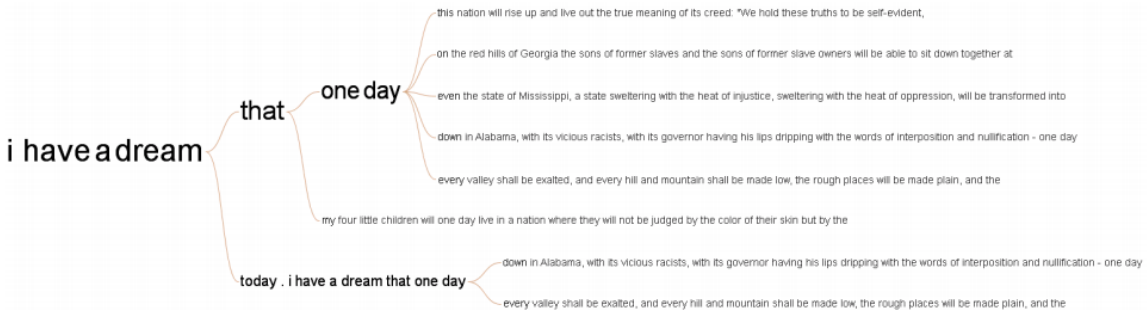


Figure 15: A Word Tree visualization of all occurrences of I have a dream in Martin Luther Kings historical speech [85].

Some visualizations try give an overview of keywords of the corpus by showing keywords, and also show context around the keywords by displaying patterns extracted from the corpus. Some visualizations we discussed in the Opinion Mining section display noun-adjective pairs extracted from the original reviews [94, 46] in a word cloud fashion.

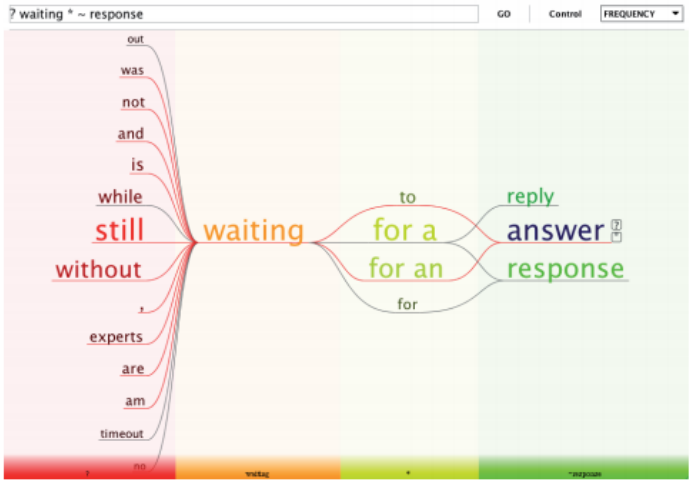


Figure 16: A WordGraph visualization of search results of the wildcard query '? waiting * response' [70]. Notice that branches not only appear on both sides of the pattern but also in-between the pattern.

Other visualization techniques show a graph of words to display structures from the data. The phrase net [80] visualization technique identifies word pair relationships (e.g., X and Y, X of Y, X's Y) in a document and displays those pairs in a node-link network (Figure 17). Although applied for a different purpose, namely showing uncertainty in statistically-derived lattice structures, the layout algorithm by Collins et al. [25] also attempts to position words to reflect sentence structures. It positions words from a sentence horizontally with potential replacement words drawn above others (Figure 10).

2.7 Style analysis

Text visualization are often used to reveal writing styles of different documents and authors.

The phrase net technique mentioned in the previous section can be used to surface patterns in that commonly appear in a document. These patterns often reveal the tendency of writing in a certain style [80] .

Heatmap-style visualization is used to display "literature footprint" of documents, with each text unit represented by a small square which is colored according to the

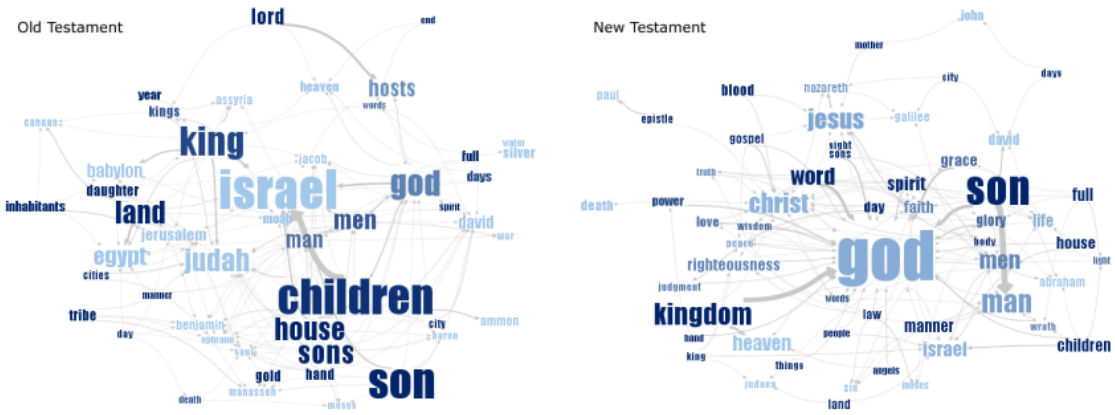


Figure 17: Phrase net visualizations comparing the old and new testaments for the pattern "X of Y" [80].

calculated value of a literature feature. The entire document is represented by a map of colored squares and patterns emerge when comparing maps [49]. VisRA [66] applies this technique to help writers quickly understand the readability of his/her writing on a sentence level (Figure 18).

Figure 18: "Literature footprint" of two novels by Jack London [49]. Color indicates vocabulary richness. It is clear that the two novels are very different, and the first and latter halves of the Iron Heel are also quite different.

CHAPTER III

VISUALIZATION OF CONSUMER REVIEWS TO SUPPORT DECISION-MAKING

In this chapter, I describe my work in the domain of consumer reviews. Among text documents produced by online communities, consumer reviews are relatively structured around a given topic (the product) and the associated analytic task is relatively clear (make a decision on whether to purchase). The primary challenge are that the review text is just free-form natural language and the potential number of reviews creates a scalability issue.

3.1 Background

Reviews are extremely valuable to consumers as they provide information that is difficult to observe in advance of the purchase [65]. For this reason, reviewers often have the power to influence future consumers' buying decisions [67]. Moreover, businesses also study reviews to adjust their product or service strategies [97]. Therefore a holistic and thorough understanding of reviews can benefit both consumers and businesses, while biased reviews may impact both parties negatively. In this project I focus on helping users understand reviews for informed purchasing decision. Research shows that gaining insights is becoming increasingly challenging for human readers as the number of reviews gets larger and larger [22, 44, 45, 95], so there are rooms for software assisted review reading.

Commercial websites often employ one of two approaches to help users explore large review sets. One approach, used by sites such as Amazon.com, lets readers vote

on the helpfulness of each review, and directs future readers to the most helpful reviews. The other approach, applied by sites such as Bing Shopping and Google Product Search, provides an overview of the most frequently mentioned product/service features, and the overall sentiment expressed in a collection of reviews. Users can then filter the reviews based on the identified features.

While a few high-quality reviews or the aggregated sentiment may provide useful information, previous research shows that users often desire finer-grained understanding of the reviews [22, 95]. In particular, people process information in an attribute-driven manner in the absence of actual products (e.g., online shopping) [56]. In such cases, people examine the attributes of a product to evaluate whether the product fits their purchase goal (e.g., buying a camera for underwater adventures). In addition, the positive or negative sentiment expressed by the reviewers toward each attribute helps justify the suitability of the product [95].

To facilitate attribute-driven evaluation of products, a number of systems produce an aspect-based summary, including the extraction of sentiment toward each of the aspects [22, 46, 60, 95]. Among these systems, several recent ones use noun-adjective pairs to summarize the aspects of a product/service (noun) and the sentiment (adjective) toward each aspect [46, 95]. However, this approach has several limitations. First, they cannot handle implicit opinions. For example, they cannot extract aspects "weight" or "size" implied by the expression "it is light and portable" [95]. Second, they do not deal with conflicting opinions expressed by different reviewers. For example, one reviewer raves "the screen is fantastic", while the other complains "positively claustrophobic in terms of screen usage". In such cases, the existing systems do not make it clear to the user that one noun (e.g., "screen") is associated with multiple adjectives (e.g., "fantastic" and "claustrophobic"). Third, the performance of these systems is limited by the imperfections in the underlying natural language processing (NLP) techniques. Because of the flaws in NLP (e.g., classifying "impeccable" as a

negative sentiment), users may find certain summaries mystifying [95].

Given the importance of opinion-mining and review summarization, researchers have developed sophisticated NLP techniques for aspect extraction and sentiment analysis. However, due to the challenging nature of the problems, the techniques still make many errors, and require large amount of labeled training data from matching domains. At the time of this research project, the state-of-the-art NLP techniques could only achieve 50% to 85% accuracy for either aspect extraction or sentiment analysis, depending on the domain [48, 61, 59, 63, 64, 78]. Presenting the NLP results directly to users can lead to them immediately noticing the errors, which is often followed by user frustrations and even distrust in the system as shown in users studies [95].

Based on previously described research into review reading behavior, I developed a novel interactive visualization system that assists reading of large quantities of reviews, and addresses the challenges in NLP. There are two high-level design goals: (1) automated creation of an aspect-based, effective visual summary to support users' real-world opinion analysis tasks, and (2) interactive user correction of system text analytic errors to improve the system quality over time. Meeting the first goal gives users a great tool to get the most out of consumer reviews and also motivates them to help correct system errors. User-assisted error correction is my solution to the NLP limitation. The outcome of the research is a prototype system named OpinionBlocks. I published and presented my work on OpinionBlocks at the International Conference on Human-Computer Interaction (INTERACT) in 2013 [43].

3.2 System design

In this section, I describe the design of the OpinionBlocks system. The user interface was designed to support two main goals: interacting with both the generated visual summary and the original reviews, and correcting system errors in text analytics.

To achieve the first goal, OpinionBlocks employs advanced NLP technologies to automatically create and present users with a fine-grained, aspect-based visual summary of opinions. As shown in Figure 19, the created visual summary allows a user to gain insights into a collection of reviews at multiple levels:

- Frequently mentioned aspects of a product/service, including those *explicitly* and *implicitly* expressed in the reviews (Figure 19a).
- The description of each aspect in a form of key phrases, a set of associated review snippets, and the inferred sentiment of each key phrase and snippet (Figure 19b).
- The full review containing extracted aspects (Figure 19c).

To achieve the second design goal, OpinionBlocks allows users to interact with the visual summary to amend analytic errors (Figure 22, Figure 23). It then aggregates user contributions to update and improve the visual summary for future users.

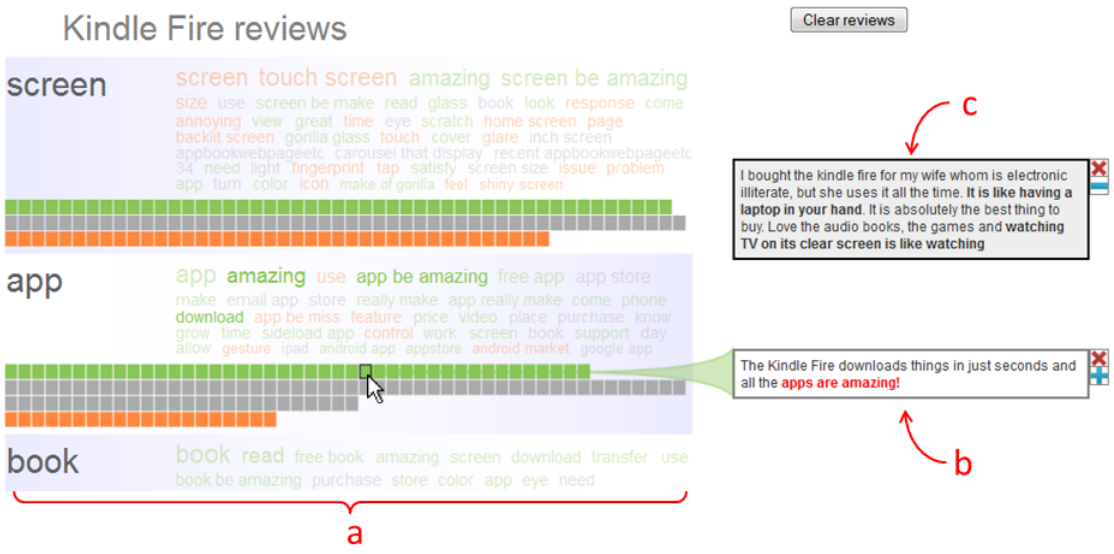


Figure 19: The interface of OpinionBlocks: (a) a system-generated aspect-based summary; (b) a system-extracted review snippet; (c) the full text of a review.

3.2.1 Interactive visualization to support decision-making

OpinionBlocks aims at aiding users in their information-driven decision-making processes. Based on previous research [46, 56, 94], we learned that a user's first step is to gain an overall impression of the important aspects of a product from available information. An informal discussion with 10 colleagues who recently made a major purchase also confirmed this habit. Our visual interface thus consists of two main parts. As shown in Figure 19, the left panel displays a visual summary of all the major aspects extracted from a set of reviews. The right panel is initially empty but shows relevant review snippets as a user interacts with the visual summary on the left.

A generated visual summary is made up of a set of aspect blocks (Figure 19a). From top to bottom, the aspects are ordered by their number of mentions in a review collection. Each aspect block further consists of three parts: (1) the aspect name, (2) a text cloud of keywords and phrases describing the aspect, and (3) a set of colored squares, each of which represents a review snippet describing the aspect. Automatically extracted from a review document (see below), a *review snippet* includes a sentence that expresses opinions toward the aspect. Three colors are used to encode the sentiment expressed in a snippet: green (positive), grey (neutral), and orange (negative). The words and phrases in the text cloud are extracted from the snippets, and are colored based on the aggregated sentiment orientation of the relevant snippets. The colored squares are placed in different rows by their sentiment orientation, facilitating the comparisons of contrasting sentiments in each aspect (e.g., how many positive versus negative comments for the "Screen" aspect?) and across all the aspects (e.g., which aspect received most conflicting reviews?).

Our design is motivated by previous research and our own study that review readers tend to form and adjust their impression of opinions by looking for most discussed and most debated aspects, and they tend to verbalize their impression

with short descriptive phrases [94]. Thus we designed the colored snippet boxes to support explicit comparison of comment frequency and polarity of sentiment under different aspects. And we help users highlight review snippets by keywords and phrases (Figure 20).

Furthermore, readers often wish to see the concrete evidence behind the extracted aspects and sentiment in a summary [60, 71]. OpinionBlocks enables users to "drill-down" through clicking or hovering on the visual elements, allowing them to see snippets associated with blocks, keywords associated with snippets, snippets associated with keywords, or even the full context of the original reviews (Figure 21).

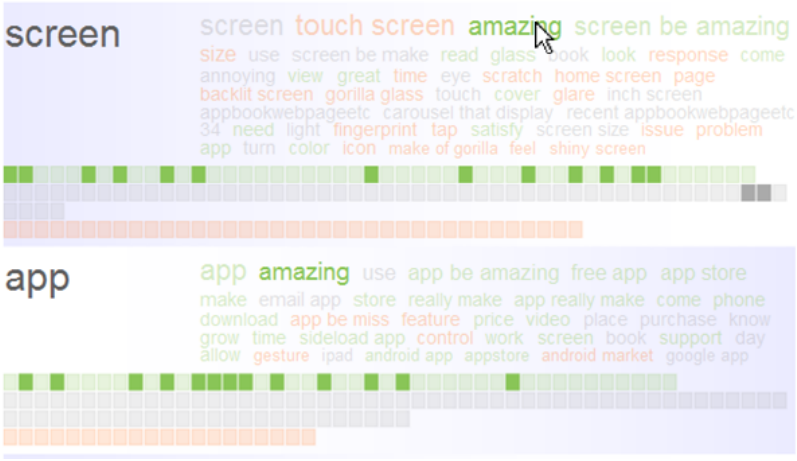


Figure 20: Hovering over the keyword "amazing" highlights all snippets containing the word.

3.2.2 Interactive Features to Support User Feedback

As discussed earlier, one of our main design goals is to leverage the power of the crowd to identify and correct system errors, in particular, NLP errors that occurred in review analysis and summarization. Yatani et al. [94] suggests that showing the contextual text behind the phrases and sentiment classification helps compensate for the imperfect analytic results. However, we wish to take a step further and encourage the users to identify and correct text analytic errors. By correcting the errors, users not only obtain a more accurate visual summary for themselves, but also help future

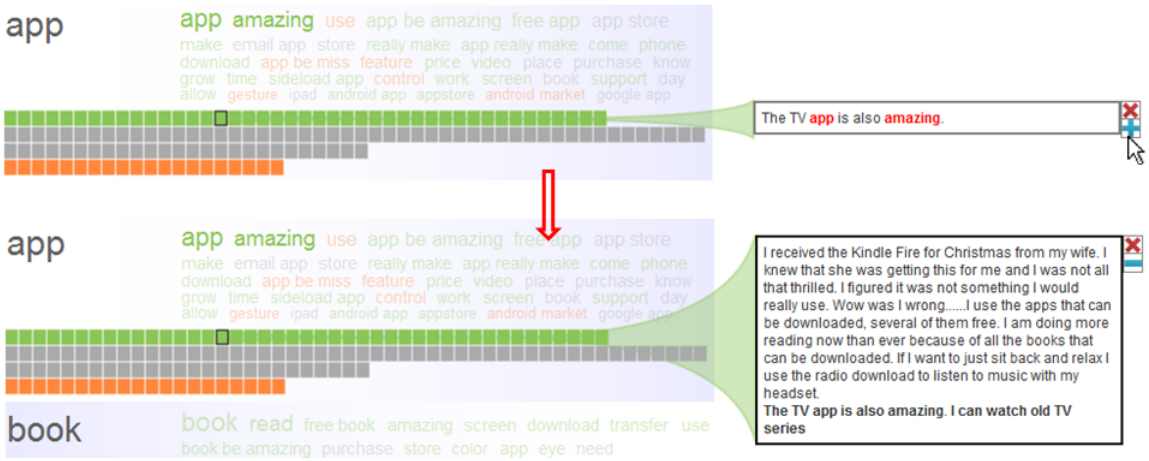


Figure 21: Clicking on the "+" button next to a review snippet brings up the full review.

users of the system. We have identified four major types of system errors:

- **Snippet omissions**: snippets that contain an opinion but were not extracted.
- **Erroneous snippet extraction**: snippets without meaningful opinions
- **Erroneous aspect**: snippets classified with the wrong aspect
- **Erroneous sentiment**: snippets associated with the wrong sentiment

OpinionBlocks focuses on leveraging users to fix the last three types of errors, since identifying the first type of errors would require the users to be familiar with the entire review corpus. To rectify the errors, users can drag the colored square representing a misclassified snippet to the correct aspect or sentiment row (Figure 22), or to somewhere out of the display area entirely if the snippet contains no meaningful opinion. Users can also click on an aspect name and change it to something more appropriate (Figure 23).

When a user makes a correction, OpinionBlocks does two things: (1) updates its interface for *the current user only* to reflect the user input, and (2) sends the user input to the back-end server and stores it in a database. Currently a system administrator

decides when to incorporate user feedback to update the system interface for all its users.

To incorporate user feedback, OpinionBlocks first selects qualified user changes among all the inputs, and then uses them to update the system. Since users may make mistakes, move things around randomly, or even try to game the system, not all user feedback can be trusted. Similar to adopting crowd-sourced results [16], OpinionBlocks incorporates user feedback only when multiple users report the same error and propose the same solution. It checks the number of identical user-corrections made against a threshold. The threshold is now set by the system administrator and may be different for different user groups (e.g., trustworthy user population versus the general public). For our user studies with the Kindle Fire reviews, we used three as the threshold. That is, if three or more users made the same change, the change is then adopted. For example, the review snippet "The touch screen has given me no problem so far" was misclassified as negative by OpinionBlocks. Six participants in our study moved this snippet to the positive row. Thus, OpinionBlocks later marked it as positive.

In practice, user-submitted corrections likely contain conflicts. A very common conflict happens when multiple users identify the same error, but recommend different solutions. For example, the review snippet "However, hardware volume control, bilateral speakers, and a more thoughtfully placed power button would have earned the Fire 5 stars from me" was classified as positive by OpinionBlocks. While four participants changed it to neutral, other three moved it to negative. In such cases, OpinionBlocks currently takes the solution by the largest number of "votes", assuming that the number of "votes" passes the threshold described above. Consequently, the sentiment of this review snippet was changed to neutral. Note that we do not require a "majority rule" here. Our rationale is that when an error is identified by many, it is better to correct it than to leave it in the system, even when there is

no consensus on the solution. Adopting the most suggested solution that passes the threshold seems sensible.

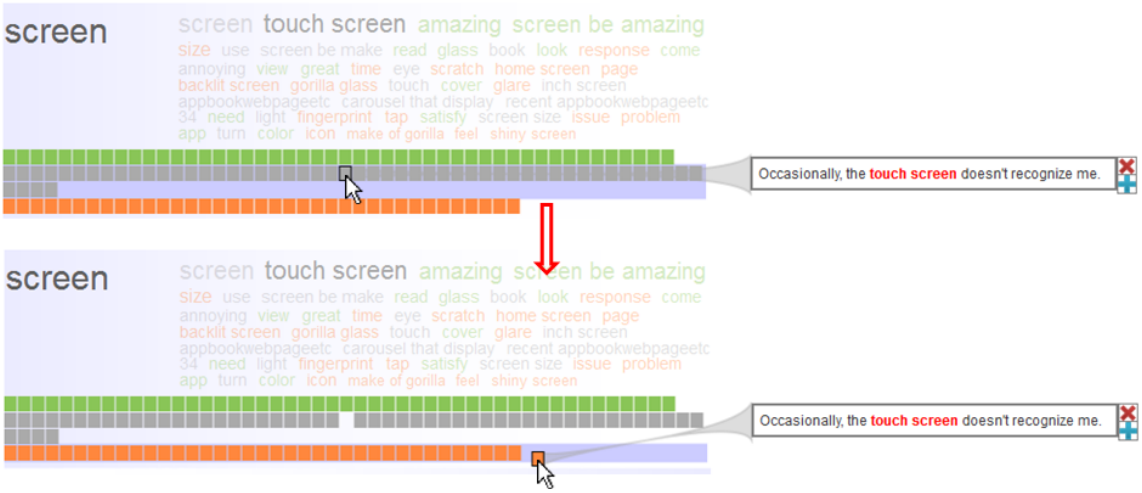


Figure 22: Moving a snippet misclassified as "neutral" to the "negative" row.

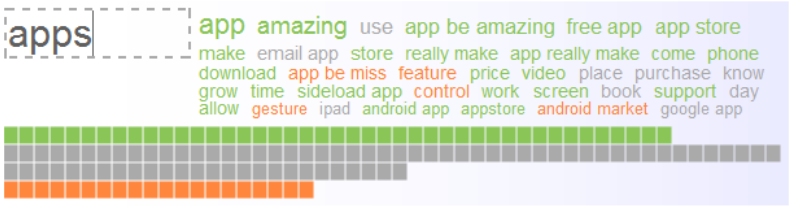


Figure 23: Changing the title of an aspect.

3.3 Text analysis component

In this section, I describe in detail the underlying text analysis techniques to OpinionBlocks. Note that I do not claim that the techniques used were innovative or state-of-the-art. Rather, these techniques were selected because they suited the prototyping needs; they can be viewed as independent components that may be upgraded or swapped without changing the design of OpinionBlocks.

To generate the information used in our visualization, OpinionBlocks performs a four-step process: 1) review snippet extraction, 2) aspect extraction, 3) keyword ex-traction, and 4) sentiment analysis.

3.3.1 Review Snippet Extraction

From a collection of reviews of a product, OpinionBlocks extracts a set of review snippets that describe various aspects of the product. To extract a review snippet, OpinionBlocks first uses the OpenNLP parser [7] to obtain a parse tree for each sentence in a review. It then builds subject-verb-object (SVO) triples based on the parse tree. For each SVO triple, it checks whether the lemma of the verb matches a selective list of verbs (e.g., be, look, appear, etc.) from VerbNet [72], which are often associated with various aspects mentioned in a review. If there is a match, OpinionBlocks then keeps the subject of a SVO triple as an aspect candidate and the sentence containing the SVO triple as a review snippet. For example, given a sentence "The display is made of Gorilla Glass, which is highly damage resistant", the extract-ed SVO triple is: [*the display, make, Gorilla Glass*]. The sentence itself is a review snippet, and the subject "the display" then becomes an aspect candidate.

Note that we generate aspect candidates by considering only noun phrases that are also subjects of a restrictive subset of sentences in the review texts (by requiring their verbs to match a limited list). This approach is inherently resistant to noise introduced by common contextual information, such as prepositional phrases and discussions irrelevant to product aspects (e.g., detailed life experience like "I tried out several different magazines").

3.3.2 Aspect Extraction

Aspect extraction is to identify frequent n-grams from aspect candidates. Specifically, we first tokenize each aspect candidate and lemmatize its tokens with the Stanford Natural Language Processing Package [9]. Next, we extract all possible n-grams of size 3 from each candidate (or the candidate itself, if its length is shorter than 3), remove any stop word at the beginning or end of the n-grams, and calculate the frequency for each unique n-gram. Our preference of longer n-grams (e.g. tri-gram vs. bi-gram)

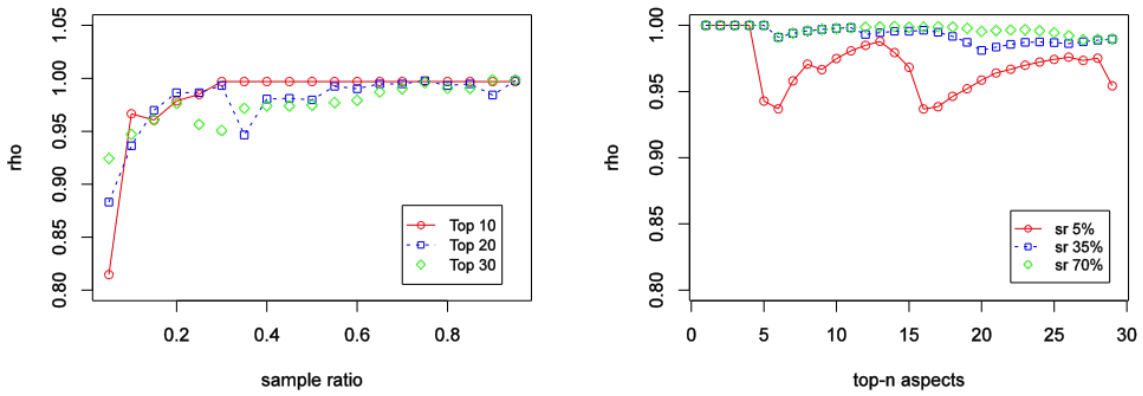
is intentional: we observed that longer n-grams are typically more informative than shorter ones and thus are better at conveying concrete information to users. We then select and use the top-K (K is adjustable in our system) most frequent n-grams as a set of extracted aspects to summarize a collection of reviews.

We conducted several experiments to investigate whether our approach of aspect extraction can generate a consistent set of aspects given different sizes of the review collections. Here, we used the top-K aspects with the full review collection as the base line to investigate the performance of our approach with different sample ratios. Two metrics are employed here: Spearman's rank correlation coefficient (rho) [34] and coverage rate, where rho measures the correlation of two ranks of top-K aspects, and coverage rate measures the fraction of the top-K aspects from the full collection that also occur in the top-K aspects from the subset of the collection. The two metrics were computed using twenty sample ratios. We performed ten test runs for each sample ratio and averaged the two metrics over the ten runs.

Figure 24 shows our experiment results. On the left, all reported rho values are over 0.8, which indicates that the top-K aspects identified with the samples are positively correlated to the aspects identified with all reviews (all values are significant). We also find that even with a small sample ratio of 0.35, the top-10 aspects have the exactly same rank as those identified using the full collection. The performance for top-20 and top-30 aspects with our approach is also very promising. For coverage rate, with a sample ratio of 0.35, our approach yields very good coverage (>0.95) for top-10 aspects and around 0.8 for top-20 and top-30 aspects. As a result, our aspect ex-traction generates consistent results over different sizes of review collections.

3.3.3 Keyword Extraction

To enrich an aspect-based summary, we also extract keywords from the relevant snippets for each aspect. We identify n-grams (unigrams, bigrams and trigrams) of



(a) rho

(b) Coverage

Figure 24: Left: Spearman's rank correlation coefficient (rho) of top-K aspects with different sample ratios. Right: Coverage rate of top-K aspects with different sample ratios.

all words and their frequencies from a collection of snippets related to an aspect. The n-grams with high frequency are used as keywords to describe each aspect. Because such a keyword relates to both an aspect and a snippet, we also use these keywords to index snippets. This way, keywords can be highlighted and easily spotted in a snippet when a user examines the snippet associated with an aspect (Figure 19b).

3.3.4 Sentiment Analysis

We use a simple lexicon-based approach [44] to infer the sentiment expressed in each snippet. This approach uses a public sentiment lexicon of around 6800 English words [8] to determine word sentiment orientation (positive or negative). We first tokenize and lemmatize a review snippet, and remove all stop words. The polarity of a snippet s is decided by its sentiment score $S(s)$, where $S(s) = |\text{positive words in } s| - |\text{negative words in } s|$. If $S(s) > 0$, then s is considered positive; if $S(s) = 0$, then s is neutral; otherwise s is negative. If the verb of a SVO-triple contained in a snippet is associated with negation (e.g. "is not"), this simple method may not work. In such a case, we set the sentiment score of the snippet to 0 (neutral).

3.4 User study

To validate the effectiveness of OpinionBlocks in meeting our two design goals mentioned in the introduction, we conducted user studies to answer two sets of questions:

1. How well does OpinionBlocks support real-world, opinion analysis tasks?
   (a) How well can users find important aspects mentioned in the reviews along with their associated sentiment?
   (b) How well can users find evidence behind reviewers' opinions?
   (c) How well can users get to the detailed facts and discussions as needed?
2. How practical is it for OpinionBlocks to leverage the crowd to improve its quality?
   (a) How accurately can users make amendments to correct system errors?
   (b) How willing are users to make such contributions?
   (c) How well do the amendments improve the system to benefit new users?

3.4.1 Study Design

To answer the questions mentioned above, we designed two identical studies and conducted them in sequence under two different experimental conditions. Both studies were used to answer the first set of questions and questions 2 (a-b) by steering the participants to identify and correct system analytic errors. In Study 2, however, the user corrections submitted in the first study were incorporated to answer question 2(c). We compared the user performance between the two studies to assess any improvements (e.g., task time) due to user corrections made in the first study. We used disjoint sets of subjects between the two studies, i.e., a between-subject experiment design, to avoid any learning effect.

Participants

Since OpinionBlocks is designed to help end users, we conducted both studies by recruiting participants from Amazon Mechanical Turk (called turkers from now on). After a pilot, we recruited 50 turkers for each study. Turker qualifications included being located in the United States, having done at least 50 approved Human Intelligence Tasks (HITS) on the site, and having over 98% approval rating for all HITS. Each approved task completion was paid $2.59 US dollars. Measures were taken to ensure that one turker could do the task only once.

Data Set

We used Kindle Fire reviews from Amazon.com as our primary data source. We selected this data set for two reasons: First, it is a large data set that can be used to assess user performance in real-world tasks. Second, it is in a domain that may appeal to a general audience. At the time when we conducted the studies, there were over 18,000 reviews on Kindle Fire, with more reviews added daily, indicating people's strong interest in the product. Overall, OpinionBlocks extracted 3034 aspects and 48,000 review snippets from the 18,000 reviews.

Tasks and Measures

Each turker was first directed to an online survey that contained a set of instructions and questions about the tasks. The survey started with a scenario: "Suppose you want to buy a tablet. You have just heard about Kindle Fire. You'd like to learn more about it so you can make an informed decision." The turker was then given a brief tutorial in a sequence of annotated screen shots of OpinionBlocks, explaining each interface element and function.

After the tutorial, the turker was given a link to launch the *live* OpinionBlocks tool in a separate browser window/tab. After OpinionBlocks was launched, the turker was then instructed to go to the next page of the survey to answer questions using the tool. There were a total of 27 questions in each survey, including fact-finding questions about the product (e.g., "Which aspect of the product received the most

conflicting reviews") and questions about the tool (e.g., "How would you rate your experience using our tool to explore the reviews"). A timer was started when the page of the survey containing the fact-finding tasks was loaded. The timer stopped if all the questions on the page were answered and the page was turned to the next one. The timed duration was used as a measure of completion time for fact-finding tasks.

3.4.2 Results

We received 50 completed surveys for our first study and 51 for the second one. After reviewing each response, we approved all of them. On average, each turker spent 35.5 minutes on our survey.

1(a) How Well Can Users Identify Important Aspects/Sentiments?

Suppose that users are potential customers in the market for a tablet. We designed two related questions to investigate this aspect. First, we asked them a yes/no question on whether they could make an informed decision on the tablet based on their use of OpinionBlocks. This question was to assess the users' overall confidence in their comprehension of important factors and their associated sentiment to influence their buying decisions. 81 out of the 101 turkers confirmed that the information is sufficient for them to make a decision on the product. One user also provided the rationale for his "Yes" answer: "there are more green bars than orange".

The second question asked the turkers to find the important aspects of the product. This question was to examine whether a user's understanding of the aspects was consistent with what the system provided. To do so, we counted the number of times that the users' responses contained at least one of the top-three aspects identified by the system: "screen", "app", and "book". 76 out of the 101 turkers' produced correct answers, indicating user-identified main aspects were consistent with that of the system.

In addition to these two questions, we also used a set of questions such as "Which aspect has received the most conflicting reviews?" to assess how well users can use OpinonBlocks to identify aspects with distinct characters (e.g., most positive, negative, and controversial). For these questions, for example, 66% of turkers in Study 1 successfully identified "screen" as the aspect that received most conflicting reviews, while 72% turkers did so in Study 2. Moreover, the turkers were able to cite both positive and negative sentiments to substantiate their findings (see more below). Considering that there were 3034 aspects extracted from 18,000 reviews, OpinionBlocks demonstrated its effectiveness in helping users identify salient aspects of the product.

1(b) How Well Can People Find Evidence to Substantiate an Opinion?

We designed three questions to ask the turkers about various review details (e.g., "What products are the main competitors of the Kindle Fire?"). For all of these fact-finding questions, users were required to excerpt one or two sentences from the reviews to support their answers. Two coders independently read all turkers' responses (3x101=303 responses from two studies) and marked the responses (Yes or No) based on whether the cited sentences correctly supported the answer. Krippendorff's alpha was computed to measure the inter-coder reliability, where alpha = 0.70, suggesting a good level of consistency between the two coders. We then computed the percentage of turkers that correctly found evidence to back up their answers (0.5 was used when the two coders diverged). Out of 303 responses, 274 were correct (90.4%). Clearly, OpinionBlocks was able to help users find specific evidence for opinions.

1(c) How Well Can People Get to Important Details?

As described above, we learned that users were able to cite relevant evidence to back their answers. However, we also wanted to measure how *accurate* their answers were. To do so, two coders independently read each answer to judge whether it was consistent with the answers suggested by the original data. The inter-coder reliability was measured at alpha = 0.82. The percentage of turkers that gave the correct answer

was 95.2%, indicating that the majority of users were able to use OpinionBlocks to find desired details of the product when needed.

2(a) How Accurately Can People Make Amendments?

During the studies, each turker was asked to identify and correct at least ten text analytic errors in OpinionBlocks. Each turker was randomly assigned five aspects displayed in the visual summary to perform this task. From Study 1, we collected a total of 659 user-made changes. Many of the changes were made by multiple participants. After removing the duplicates, we obtained 378 distinct amendments. Among them, 47 corrected misclassification of snippets by aspect; 347 corrected misclassification of snippets by sentiment; and 16 corrected both at the same time. After applying our rules for integrating user feedback, 49 unique amendments were incorporated into OpinionBlocks for Study 2.

Two coders examined the 378 unique changes and coded each of them to assess the correctness of the changes. Due to the inherent semantic ambiguities in interpreting the snippets, the initial independent codings had relatively low inter-coder reliability with alpha=0.36 for both aspect and sentiment placement. This low agreement in perception of aspect-based sentiment is also observed by Brody et al. [18]. Meetings were held between the coders to discuss a more consistent way of coding the results. They identified two common cases of ambiguity and built a set of coding rules: (a) the interpretation of sentiment orientation should be anchored around the aspect first then the product. For example, one snippet stated "after using the fire for a few weeks now, my ipad is gathering dust." If this snippet is under aspect "iPad", then it should be classified as negative, but if under "tablet", it then should be positive; (b) if a change makes sense or does not make it wrong, count it as correct. For example, the snippet "Software Controls - I can see why the lack of external buttons would annoy some but for me it is not a problem" can be interpreted as positive or neutral.

After applying these coding rules, we achieved good inter-coder reliability, 0.91

for aspect and 0.97 for sentiment respectively. The averages of the two coder's ratings were used in the accuracy calculation. For aspect placement, 22 of the 47 changes were coded as correct (46.8%); while 247 of the 347 sentiment changes were accurate (71.2%). These results suggest that people are more capable of fixing sentiment errors than aspect errors. Since the accuracy rates were not as high as we had hoped, we computed the accuracy rate for the 49 changes incorporated by OpinionBlocks, and found that these changes achieved an accuracy of 88.8%. This demonstrates the effectiveness of our user feedback integration rules (section 3.3), and suggests that OpinionBlocks can be improved by crowd-sourced input over the use of state-of-the-art machine learning techniques alone.

2(b) How Willing Are Users to Make Amendments?

We explicitly asked turkers about their willingness to make changes while using the system. From their answers, most users (95%) are willing to contribute.

We also asked the turkers to explain their main reasons for their answers. The reasons given by people who were willing to contribute fell into several categories:

About 50% of the turkers said that they would like to help improve the quality of the tool for its better use. For example, one said, "I'd be willing to spare a few seconds to improve a tool that I would gladly use." Another commented: "Those features are key to the tool's use" and "it can make the tool more useful and correct".

About 15% cited the community and social benefits. The reasons include "I thought this was a useful feature that made the tool more of a community-use tool rather than just an individual-use tool."; and "I think it will go a long way in making users of this app feel like they're contributing in some way. It may even become a draw of sorts for the app."

About another 15% felt simply that it was fun and cool to correct things. They mentioned "It's fun!", "It was interesting to correct the errors, because I found myself trying to figure out why each incorrect snippet had been improperly categorized.";

"It's cool that you can edit things."; "I like organizing things. Especially when mis-rated reviews stick out like a sore thumb."

The majority of people who expressed their unwillingness to contribute (5% of participants) voiced their main concerns about the potential abuse of the system: "If this was used by multiple people, it would end up being very abused."; "My only concern here is people messing with the system to improve reviews of their own products or make competitors look bad."; and "it's handy but should be checked by someone".

Other unwilling participants just did not want to bother, or wanted to get paid: "I'm not really interested in correcting mistakes."; and "I can't see doing it out of the kindness of my heart. If it were on Mechanical Turk I could see doing it for a small amount of money."

Overall our results suggest that it is feasible to leverage the power of the crowd to help improve the system.

2(c) How Much Have User-Amendments Made the System Better?

As discussed earlier, the turkers made many changes, of which 49 most common ones were integrated by OpinionBlocks. The incorporated amendments achieved an accuracy of 89%, thus improving the quality of the visual summary.

To measure the impact of integrating the user edits from Study 1 on user tasks, we compared user performance in both studies. To do so, we performed statistical tests using the sequence number of the studies as the independent variable, and all the performance measures taken in the studies as the dependent variables. We found that the turkers' time for completing fact-finding tasks in Study 2 (M=768.1, SD=338.5) was significantly lower than that of Study 1 (M 916.6 seconds, SD =370.2), t98=2.10, p=0.04. Turkers in two studies performed equally well in term of finding correct facts about the products and relevant evidences. In addition, turkers were equally satisfied with our system in both studies. On a 5-point Likert scale, both obtained a median

4 satisfaction ratings, with 5 being "very satisfied".

Overall, our results showed that it is practical to improve the system by leveraging the crowd to correct system errors, and the resulting improved system lets users perform tasks equally well, but significantly faster. One plausible reason for the improved task completion speed is that in the improved system, there is less misplaced unhelpful information, so users do not need to waste time reading.

3.5 Limitations and discussion

3.5.1 Limitations in Text Analytics

OpinionBlocks has adopted several text-mining approaches to analyze opinion text and glean useful insights. It also leverages the power of the crowd to help compensate for system mistakes and improve the overall analysis quality. Nonetheless, due to inherent difficulties in text mining, our current approach presents several limitations.

One difficulty is to decide which review snippets to include and how "big" each snippet should be. Currently, OpinionBlocks includes only text snippets following the sentence structure described in Section 3.3.1. This means it may miss out many useful sentences that do not conform to such a structure. Currently, each snippet contains only one sentence. This might be undesirable in situations where multiple adjacent sentences are used to express an opinion. The challenge is to balance the accuracy and recall when extracting review snippets, as well as balance the size of a snippet to provide sufficient information without overburdening the text analytic engine or the reader. To make the problem more difficult, striking such a balance may depend on factors particular to the data sets.

Another difficulty we have encountered is to determine which aspects to extract. Currently we extract aspects directly out of subject noun phrases, thus covering multiple categories. Besides aspects, such as "screen" and "app", which describe the Kindle Fire, we also extracted "iPad" which is a major competitor of the Kindle Fire.

Other extracted aspects, such as “wife”, “husband”, and “kid”, describe possible user groups of the Kindle Fire, and the aspect “problem” falls in a generic category applicable to any product. Depending on users and use cases, some might want to see only the aspects pertinent to the product, while others may want to learn more about the aspects of competing products (e.g., aspects of iPad in the context of Kindle). More work is needed to make aspect extraction more meaningful and extensible.

3.5.2 Common Ground Versus Personalization

In the User Studies section, we show that opinions are often ambiguous and that different people may interpret them very differently. Building a “ground-truth” of opinion summary is non-trivial and is unlikely to satisfy every user. Allowing a certain degree of personalization may be desirable in support of individual users? deci-sion making. Currently, OpinionBlocks allows each user to make amendments that affect only that user?s private session. These changes are propagated more widely when the system administrator decides to do so, and only high-quality changes suggested by many users are adopted. Thus, the standard version of OpinionBlocks that every user starts with is quality controlled, even though users may make amendments to their own private sessions. Complications may arise when merging divergent sets of amendments from many users. This will certainly make a good future research topic.

3.5.3 Potential System Abuse

A few participants of our user studies expressed their concerns over potential abuse of a system like OpinionBlocks, including trolling or businesses manipulating the information for their own commercial gains through user amendments of opinion summaries. Currently, OpinionBlocks gives the system administrator a great deal of control over which amendments can be integrated into the system. The system administrator can decide to tighten or loosen the threshold for integration or filter out changes from certain users. While further research is required to figure out how

to best monitor and moderate user behavior, one approach is to the leverage the crowd themselves. As shown in our user studies, the accuracy of aggregated user amendments is much higher than that of individual changes. This means aggregation of crowd input may help prevent or reduce malicious behavior. Currently our aggregation rules are very simple, future research is needed to develop more sophisticated rules, e.g., incorporating information such as the degree of difficulty of text analytic tasks and user reputation.

3.5.4 Fostering Healthy Online Review Communities

Our work bears a major implication on the research in online communities. Gilbert and Karahalios [37] pointed out two problems of current review sites: (1) large numbers of reviews are never read and in essence wasted; and (2) "pro" reviewers dominate the community and it's hard to hear the voice of "amateur" reviewers. They call on system designers to nudge community members toward community-wide goals. OpinionBlocks helps address both problems: It summarizes the reviews and helps users understand large collections of reviews. It also fosters a democratic environment for others to contribute. In short, we have taken the first step to create a platform to foster a healthier online community where users can potentially help the system and help one another.

3.5.5 Value to Text Analytics Research

It is also worth noting that our approach of marrying machine and human intelligence to text analytics produces invaluable assets for text analytics research. First, crowd feedback can be used as an indicator to identify "high-value" areas for users. As shown by our study results, users mostly made corrections to the sentiment classification but only a few on the aspect classification. This suggests that users may be more sensitive to certain types of errors than others. Moreover, user-submitted corrections can be used as a training corpus to help tune analytic algorithms. This is related to research

efforts in interactive machine learning, where a machine learning process is augmented by human intelligence to improve the results. For example, Patel et al. presented a development environment that helps developers find and fix bugs in machine learning systems [68]. Amershi et al. developed systems that can iteratively learn the desired results based on end-user interaction behavior [14, 13]. Similar to these efforts, our work also aims to improve machine intelligence (i.e., text analysis results) through user interaction. However, while prior work focuses on leveraging individual users to improve machine learning, we focus on leveraging the wisdom of the crowd to improve analysis of unstructured text, which is potentially more cost-efficient, though it also presents unique challenges such as the need to reconciling crowd inputs.

3.5.6 Contribution

In this chapter I presented my work in the consumer reviews domain which led to OpinionBlocks, a novel visual analytic system that aids users to analyze large sets of opinion text. OpinionBlocks is uniquely designed to combine interactive visualization, NLP technologies, and crowdsourcing to aid users in their real-world opinion analysis tasks. OpinionBlocks offers two unique contributions:

- It supports real-world, opinion analysis tasks beyond that of existing visual opinion analysis systems.
- It leverages the power of the crowd and the interactive nature of visualization interfaces to self-improve the quality of the text analytic results and compensate for the limitations in today's NLP technologies.

As demonstrated by the user studies involving 101 users on Amazon Mechanical Turk, the majority of participants not only were able to use OpinionBlocks to complete real-world decision-making and opinion analysis tasks, but they also exhibited a surprisingly high degree of altruism and concerns for the well-being of online

review communities. As users gain value from the system, they become willing contributors to help correct system analytic errors and improve the system. Moreover, the crowd-assisted system enhancement significantly improved task completion time. Based on these findings, combining visual analytics with crowd-sourced correction is thus shown both feasible and effective.

CHAPTER IV

VISUALIZATION OF SOCIAL MEDIA POSTS TO SUPPORT EXPLORATORY ANALYSIS

4.1 Introduction

The previous section summarizes my work in the consumer reviews domain. In this section, I discuss projects with a focus in the social media domain. Compared to consumer reviews, social media text is usually much more informal in form: Users typically write with a length limit in mind, therefore they often resort to abbreviations and other non-standard language. While topics sometimes emerge organically, social media conversations are not typically organized around a topic, unlike review sites or message boards which enforce topic threads by design. Social media sites also provide very little structural guidance towards expression form, unlike review sites which usually ask reviewers to provide a rating (thus setting the sentiment) and highlight pros and cons clearly.

The tasks in social media analysis sessions also tend not to be as well-defined: When examining consumer reviews, analysts usually have a clear high-level goal in mind, such as "decide on the best product to buy". With social media, the task is usually as vague as "understand all that is interesting to know about this topic" because it is near impossible to anticipate what the masses will say about a topic. Therefore, a social media analytic task is usually described as an open-ended, "exploratory analysis" process where the analyst starts without a clear goal or even a question in mind, and gradually explores the dataset for interesting insights.

Given the challenges with social media analysis, I first conducted a study with Twitter data myself. I used the limited tools available at the time and wrote a number

of computer programs for this project. In the following section I describe the analysis process and what I learned from it. This project helped me identify one of hardest problems with open exploration of social media text - gaining a quick, high-level understanding of the textual content to understand the main opinions and sentiment, and identify areas for further exploration. I designed a novel visualization technique, called SentenTree, for this task. I will describe the design and demonstrate its usage in a later section.

4.2 Breaking news on Twitter: a case study of exploratory analysis of social media text

4.2.1 Background

Microblogs play an increasingly important role in our social life and are gradually transforming the ways we communicate. One striking example is how the news of Osama Bin Laden's death leaked through Twitter. On the night of May 1st, 2011, US President Barack Obama addressed the nation at 11:35 pm EST and announced that Osama Bin Laden had been killed. However, as later noted by multiple sources [5, 6, 4], millions of people had already learnt of the news before the White House announcement thanks to Twitter. The person credited with breaking the news is Keith Urbahn, an aide for former US Defense Secretary Donald Rumsfeld, who tweeted at 10:24 pm that he had heard of the death of Osama Bin Laden from a reliable source. The tweet quickly went viral and produced numerous retweets and discussions. 21 minutes later major TV channels ABC, CBS, and NBC broadcast the news, which prompted even more Tweets. Twitter later reported that it saw a historic high in Tweet volume between 10:45 to 12:30 pm that night, with an average of 3000 tweets per second [11].

In 2011, the use of social media in news reporting was still emerging, and mainstream media's reaction to the significance of the Osama Bin Laden story was quite

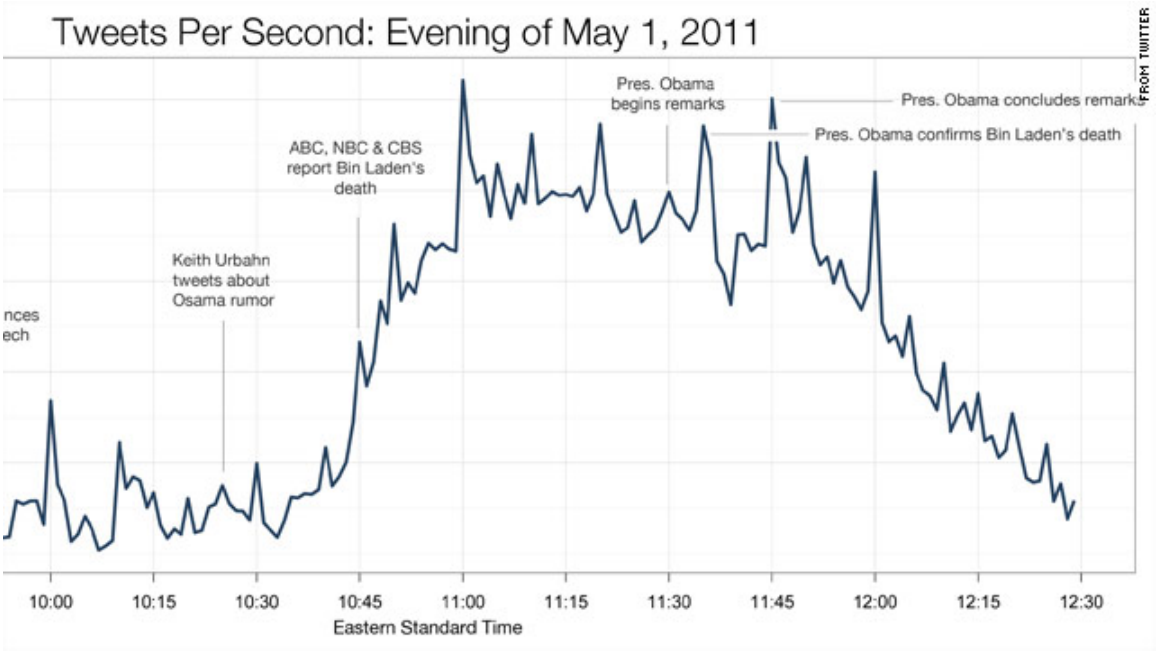


Figure 25: Tweet volume on the night of Osama Bin Laden's raid announcement [11].

mixed. While some predicted that this was a turning point and Twitter would completely reshape the landscape of journalism [6, 10], others dismissed Twitter as merely spreading rumors which no one believes until confirmed by the mainstream media [12]. A startup company named SocialFlow published a blog post in which they charted a follower network graph of Keith Urbahn to examine how his Tweet spread [4] and identified @brianstelter as a key person in helping spread the news. Inspired by SocialFlow's study I decided to perform a comprehensive analysis of Twitter's role in spreading the news, with a focus on understanding Twitter's relationship with traditional media.

It is worth noting that at the time of this study social media was still young. There were very few published case studies of social media's role in news events, so our analysis was completely open-ended without previous work for guidance. We referenced published work on meta-analysis of microblogging behaviors and information diffusion on Twitter: Java et al. [47] surveyed people's intentions in using Twitter and found most people either use Twitter to talk about daily life with friends or to

seek and share information. Kwak et al. [54] examined the topology of Twitter and concluded that Twitter is closer to an information network than a social network. Wu et al. [88] studied how news flow from mass media to the public on Twitter in the context of media communication theories and found that instead of acquiring information directly from mass media, most people rely on an intermediate layer of opinion leaders. In our study we also looked for opinion leaders and confirmed Wu's discovery. Also of relevance are studies that tried to identify major topics or utterance from news media and link social media posts that cite them [58, 79]. Our results are consistent with the high-level findings of these works that social media draws from news events and impacts mass media in turn. For this case study we did not work on identifying phrase variations or linking similar posts due to the complexity of text analysis, but the SentenTree project is inspired by this idea.

The case study was conducted in collaboration with colleagues at Microsoft Research Asia and published at the SIGCHI Conference on Human Factors in Computing Systems in 2012 [40].

4.2.2 Data

To perform a comprehensive study of Twitter's role in this breaking news event, we extracted Tweets about Osama Bin Laden posted in a two-hour time window during and after the breaking of the news.

Our data comes from a Twitter database maintained by our collaborators at Microsoft. Due to the huge volumes of data on Twitter, they employ a sampling method to obtain only a portion of the data. This method periodically collects Tweets for approximately 40 consecutive seconds in every two minutes, and randomly samples roughly 30% of all Tweets during those 40 seconds. Although the data that we worked on is incomplete, we argue that because of its large size and the systematic sampling strategy, we were able to infer general trends and significant disruptions on Twitter

through the sampled data.

Since we are interested in the "breaking" of the news, we decided to focus on May 1st 10:20pm to May 2nd 0:20am EST, which covers both the first rumors of the news and President Obama's speech. We were able to collect 614,976 Tweets containing the string "laden" posted during those two hours. We are aware that we miss many Tweets related to our subject matter that do not contain the term "laden". We argue that the vast majority of the missed Tweets are non-English, and since we are primarily interested in analyzing English Tweets they do not pose a big problem. We also miss some English Tweets that address the event but do not include the string "laden", but we believe the much larger "laden" set can generalize for the missed English Tweets. There also exist Tweets that contain "laden" but are unrelated to the event, but we believe the ratio is extremely low, since 98.52% of the Tweets in our dataset contain the string "bin" and 54.66% contain the string "osama" or "usama".

4.2.3 Did Twitter break the news?

The first step of our exploratory analysis is to confirm an existing hypothesis, namely the claim that @keithurbahn was the person who broke the news on Twitter. We examined all Tweets between 10:20 pm and 10:45 pm (when three major TV channels announced the news) looking for the most frequently mentioned Twitter users. We found the top mentioned users during those 25 minutes were @jacksonjk, @keithurbahn, and @brianstelter who were mentioned 3370, 1177, and 593 times respectively. @keithurbahn tweeted at 10:24 pm: "So I'm told by a reputable person they have killed Osama Bin Laden. Hot damn." @jacksonjk, a CBS News producer, tweeted 8 minutes later: "House Intelligence committee aide confirms that Osama Bin Laden is dead. U.S. has the body." @brianstelter, a New York Times reporter, retweeted both their Tweets and helped spread the news. These findings support the claim that Keith Urbahn wrote the first Tweet about Osama Bin Laden's death that made an

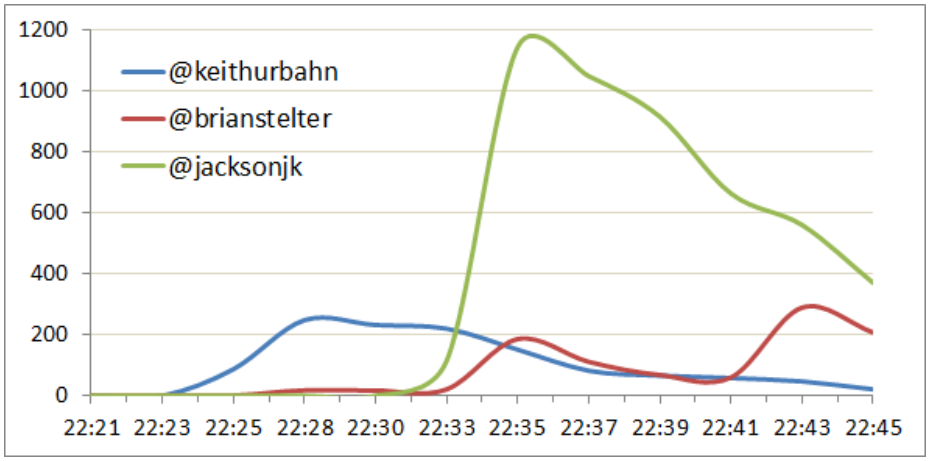


Figure 26: Trend of mentions for three key players during the breaking news moment.

impact on Twitter.

4.2.4 Did Twitter make a real impact?

Twitter may have broken the news first, but the question remains whether its users believed these early Tweets or if they viewed them as mere rumors. This was a key matter of contention when debating Twitter's significance in news reporting over traditional media. We wanted to find the percentage of Tweets convinced by the story and see how it changes before and after announcement from the three TV stations.

Given the size of the data, it's impossible for human analysts to read through all Tweets, so we trained a classifier to classify all English Tweets in our dataset as "certain", "uncertain", or "irrelevant". We assumed that if people expressed doubts or reservation then they were uncertain about this event. On the other hand if they made a statement about this event as if it was a fact then they had high confidence in it being true. So "Rumor, Bin Laden dead. Don't know for sure" indicates the author was uncertain about the event, while "They caught Osama Bin Laden!" infers the author was fairly sure the event was true. After labeling each Tweet "certain" or "uncertain", we calculate the percentage of "certain" Tweets as a confidence score. We recognize that the confidence score might be slightly inflated since people sometimes make statements about things even though they consider them rumors. Instead of claiming

that the percentage of certain Tweets directly translates to the percentage of Twitter users convinced by the news, we look for changes in percentage of certain Tweets which indicates shifts of confidence among Twitter users.

To train the classifier, we selected 300 Tweets posted before the "breaking news" Tweet, immediately after it and at the end of the dataset. Two researchers familiar with the dataset individually labeled each Tweet and they agreed on 235 Tweets (78.3%). Of the 235 Tweets, 54.9% were certain, 42.1% were uncertain, and 3.0% were irrelevant. We used the 235 Tweets to train two binary classifiers. The first classifier determines whether a Tweet is relevant and the second classifier classifies the relevant Tweets as either certain or uncertain. The Support Vector Machine (SVM) technique was used with bag-of-words as features. We estimated the performance of the classifiers with a 5-fold cross-validation scheme [35] and reported 75.8% overall confidence.

Figure 27 shows the percentage of "certain" Tweets among all English Tweets collected. We found certainty started very low and drastically rose to over 50% following @keithurbahn's Tweet. Then it gradually increased to nearly 80% by 10:45 pm. After 10:45 pm when ABC, CBS, and NBC "officially" announced the death of Osama Bin Laden on TV, certainty rose to over 80% and remained steady.

The certainty analysis suggests that a large percentage of Twitter users expressed confidence in the early Tweets. While it is difficult to explain why users were so confident without interviewing a large number of them, we have some speculation that their trust is partially due to the professional identities of the authors of the early Tweets. Examining their profile we noticed that @brianstelter was listed as a New York Times reporter with tens of thousands of followers. And while @keithurbahn and @jacksonjk did not have as many followers, their public profiles described their jobs. It is unlikely that an aide for former Defense Secretary Donald Rumsfeld or a CBS News producer would spread groundless rumors of something so important and risk

jeopardizing their reputation. To test our hypothesis we searched Tweets mentioning @keithurbahn and found 29.91% of them also included "Rumsfeld" and 18.61% of Tweets mentioning @jacksonjk contain the word "CBS", which are good indicators that people were using their professional credentials to validate their Tweets.

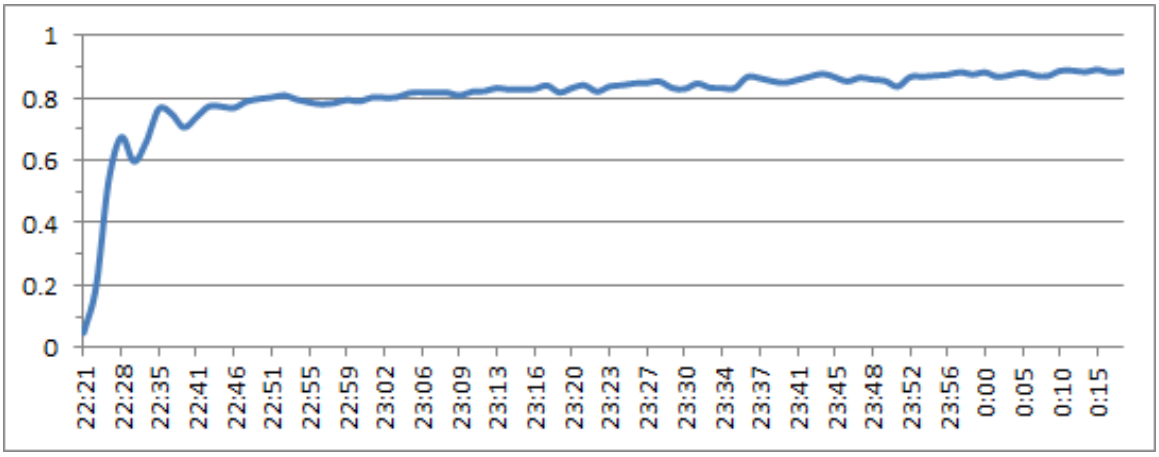


Figure 27: Percentage of Tweets sounding certain.

4.2.5 Who led the conversation?

After the news was "broken", Twitter users eagerly engaged in discussions. We wanted to understand the dynamics of the these discussions by finding out if there were leaders who got mentioned more than others. Since we had no idea if there were leaders or how many they were, we started by identifying the 100 users most mentioned in the dataset. Unsurprisingly, @CNN, @CNNEE (CNN in Spanish), and @nytimes (the New York Times) topped the list. @jacksonjk, who was among the first to report the news, ranked 4th. The 5th place went to @BarackObama. Together the 100 users were mentioned in 111,325 Tweets, which accounts for 18.10% of the total Tweets. This finding echoes the observation of Wu et al. [89] that on Twitter a great portion of all information consumed is generated by a small number of "elite users".

We examined the top 100 users one by one and manually grouped them into categories (see Table 2). We found 47 media-related accounts among the 100, with 26

being official accounts for mass media, 3 automatic news aggregators, and 18 personal accounts for individuals employed by major news organizations, such as reporters and news anchors. The 26 mass media accounts were mentioned in 5.84% of all Tweets and the 18 personal accounts were mentioned in 3.36% of all Tweets. This finding suggests that the journalists had a strong voice of their own independent from their employers. We also found that 31 of the top 100 accounts belonged to celebrities, whether in real life or only on Twitter (we treated 100,000 followers as a cutoff line for Twitter celebrities). Together these people were mentioned in 4.53% of all Tweets. Given the political nature of the subject matter, and the fact that few of the celebrities were known to be heavily involved in politics, this figure is quite high. The strong presence of celebrities is in line with the literature on social influence on Twitter [89].

To further understand the difference between categories, we charted the aggregated mention trend for the three major account groups: mass media, media people, and celebrities. As shown in Figure 28 we found that the mention patterns for the three groups were very distinct. Media people were mentioned first, which is what we would expect knowing some of them were among the first to spread the news. Around 22:45 mass media exploded with reports and instantly caught the attention of Twitter users. Celebrities tagged behind, but because of their large amount of loyal followers they gradually overtook mass media in the number of mentions. These findings suggest that the three groups influence Twitter users in different ways. While media people and the mass media compete to be the first to report the news, celebrities use their social influence to help spread the news and stimulate discussions.

4.2.6 Who were the creators of content?

As mentioned previously, at the time of this study there were heated discussions on whether social media can CREATE original content of value. In this previous section we found that people focused their attention on a few top Twitter users, leading us

Table 2: Number of Twitter accounts among the top 100 under each category.

mainstream media	26
media people	18
news aggregators	3
political figures and organizations	4
real-life celebrities	15
twitter celebrities	16
popular blogs	6
"Osama Bin Ladens", "Jesus Christs"	4
odinary users	5

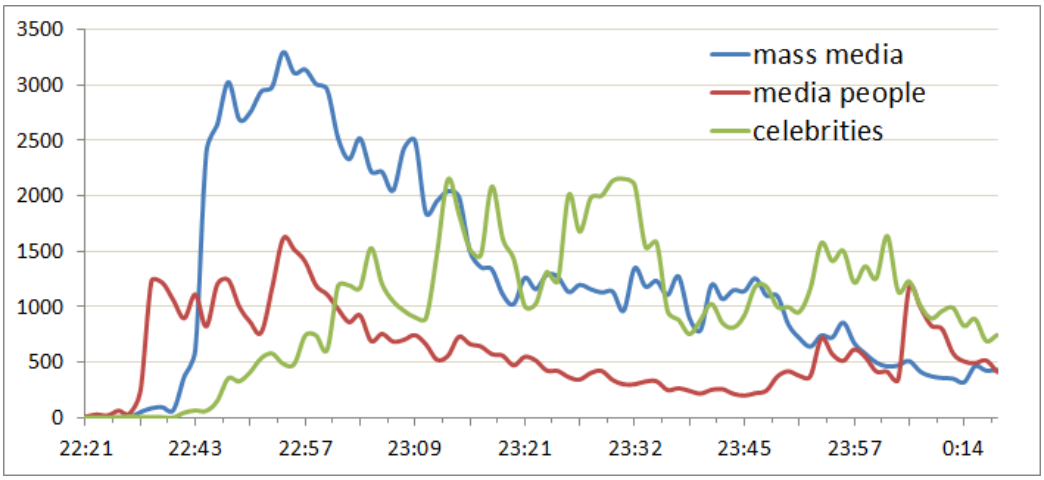


Figure 28: Number of Tweets per minute mentioning a Twitter account from one of the categories.

to believe content creators may be a small set of users. We also looked at the content shared on Twitter but keyword analysis didn't give us much information since most Tweet mention a similar set of keywords.

Knowing that each Tweet is limited to 140 characters and user often include a link in Tweets to show additional information, we also examined the sources of these shared links. Running a regular expression to matching URLs, we found that 9.69% of all Tweets in our dataset contain links. Most of the links were shortened with a Tweet-shortening service. We traced every link back to its destination web page. Then we used regular expression to retrieve the domain names and ran a script to count how many times each domain was linked. We found 26 websites accounted for 58.83% of all valid links. Table 3 lists the these sites. We manually classified the websites into mass media sites (the M group) and sites hosting user-generated content (the U group). The M group contains mainstream media sites such as cnn.com as well as sites like huffingtonpost.com which hosts curated news and blogs and enjoys high readership and reputation. Whitehouse.gov, which was not really a media site, was placed in this bin because of its authority and popularity. The U group include Tweet-shortening services (which allow people to write longer posts and include a link to the post on Twitter), picture and video sharing sites (at the time Twitter did not allow multi-media posts, so many users put images or videos on external sites and linked them in Tweets), other social network sites, and blog services. All of the sites in the U group host uploads by everyday users. We argue that in general content from the M group comes from a small number of "elite" sources, while content from the U group is produced by the common people.

We found that among the links that point to one of the top 26 sites, 64.07% of links point to a site of the M group, and 35.93% of links point to a site from the U group. This finding suggests that mass media still provides the majority of content that people share, especially in the context of news event and political discussions.

Table 3: The most linked sites and the number of times they were mentioned. The sites are labelled as either a Mass Media site (M) or a site for sharing user-created content(U).

cnn.com	4105	M	go.com	875	M
msn.com	2596	M	bbc.co.uk	826	M
nytimes.com	2569	M	wsj.com	788	M
twitpic.com	2560	U	facebook.com	750	U
yfrog.com	1689	U	whitehouse.gov	729	M
globo.com	1545	M	huffingtonpost.com	619	M
twitlonger.com	1257	U	aljazeera.net	597	M
youtube.com	1257	U	yahoo.com	587	M
tumblr.com	1148	U	google.com	568	M
reuters.com	1083	M	foxnews.com	560	M
tmi.me	1068	U	uol.com.br	551	M
lockerz.com	984	U	blogspot.com	541	U
mashable.com	930	M	globovision.com	540	M

On the other hand a significant number of users are also eager to share content created by themselves or other users. We also examined the web pages shared over 615 times (which means over 0.1% of Tweets in our dataset linked to that web page) and found six of them. All six web pages originated from mass media sites. The most popular one, linked by Twitter users for more than 2000 times, points to a video clip of Obama's speech on MSNBC.MSN.com. It seems that while ordinary users created plenty of content, their work had a limited reach. The content consumed by the most people is still created by mass media.

After publication of the original research paper, readers asked us multiple times what kind of images or videos were linked in Tweets. At the time it was very hard for us to examine linked images or videos at scale. We looked at some random samples and found that most of them were photographs of TV screens broadcasting the news. Since publication of our paper, Twitter has introduced the feature to upload pictures and videos in Tweets and created a page for exploring popular pictures related to a topic on Twitter.

4.2.7 Conclusion

Our certainty analysis on Tweets suggests that Twitter convinced many of its audience of Osama Bin Laden's death before confirmation came from mass media. We speculate this is because the people who posted the news were politicians and journalists, who were authorities on the subject matter. Our findings suggest that individuals in media related professions can play critical roles in breaking and spreading news on Twitter, since they enjoy high reputation and access to the news sources, and they could take advantage of the speed and reach of Twitter. It seems that news organizations have also noted this tendency. In November 2011, the Associated Press issued a warning to its staff members that they should file any breaking news to the wire before putting it out on social media [2].

Through our examination of the 100 most mentioned users on Twitter, we discovered a high concentration of attention on a very small subset of users and found three key user groups who influenced their audience in different stages of the news cycle. This could be interpreted through the lens of the two-step flow of communication theory, which Wu et al. discussed in length in [88]. The theory suggests most people acquire information not directly through mass media, but through an intermediate layer of "elite users", also known as "opinion leaders", who filter and interpret the information from mass media based on their own values. Wu et al. found significant evidence of the two-step flow of information on Twitter, and they identified celebrities as the most important group of opinion leaders. Our findings agree with the claim that information flow from mass media to celebrities, who voice their reactions on Twitter and pass their opinions to their followers.

Our analysis on user attention and links suggests that mass media is still at the core of reporting. Even though mass media may not be the fastest in breaking news, people still trust the content it produces more than other sources.

Our study confirms Twitter's rising potential in news reporting and identifies key

players in the breaking, spreading, and consuming of information on Twitter. We recognize that our study is limited to a single case. The Osama Bin Laden story is significant because it is among the earliest reports of Twitter breaking news before mass media. At the moment reports of social media breaking news are still rare and anecdotal [1], so it is hard to reach a general conclusion from a single case study. However, the set of methods we presented in this paper could be applied to study other situations, and we believe as social media plays an increasingly important role in our news generation and consumption, studies of this kind will become more and more valuable. The findings of the case study could also be applied to several research areas such as news event detection and tracking. For those interested in reporting early development of news stories, our result suggests that it should be important to monitor Twitter accounts of journalists. Furthermore, those trying to spread information on Twitter or to influence public opinion should target one of the "opinion leader" groups such as celebrities.

4.2.8 Discussion

This project shows a typical open-ended exploratory analysis process. When we started the process, we had little idea about what we expected to find, and we gradually formed questions or hypotheses as we progressed. For example, we started the research by trying to confirm an existing theory (that Keith Urbahn's Tweet broken the news) which led us to a related question (did the early Tweets convince the Twitter crowd). A social media analysis session includes a gradual building of insights by answering different questions about the data. And answering different questions often requires different tools to analyze and visualize the data.

We demonstrated this in our analysis process where we used different tools for different tasks. Most of these tools had to be built by hand, making the analysis process last multiple weeks. Both the task of finding the most mentioned users and the

task of finding the top linked websites were accomplished by writing custom scripts. These scripts are reusable for future projects and the logic behind these scripts can be easily incorporated into any system. When we presented the results, we used line charts to show the trend of mentioning for top users and tables to illustrate top domain names. We found we did not need more sophisticated visualizations in this particular study. However, a visualization of a re-tweet network might be useful in many social media analysis scenarios as demonstrated by SocialFlow's study [4].

Training the certainty classifier was a lot of work for us. We needed two experts manually label selected tweets for a training dataset and then train the classifier. We also acknowledge that the bag-of-words technique used in the paper was not the most advanced technique for the job. A professional analyst likely needs to have many NLP tools at hand for similar work.

When we were studying the content of the Tweets we found two major challenges: understanding the textual content and inspecting pictures/videos. For the textual content, we looked at the keywords and a few top retweeted posts, but it was challenging for us to get a holistic view of all Tweets in the dataset. In the study, we circumvented the problem by focusing on content creators instead of the content itself, but we would like to understand the content beyond a keyword level. Regarding pictures and videos we did not try to download them all for this study but instead opted to open a few random samples instead. For a more serious project focused on pictures and videos, the analyst might be willing to download all of the pictures and videos related to a topic and analyze them using image clustering software.

Finally, I would like to point out that the study was conducted a few years ago. In recent years, as social media platforms have become more sophisticated, they have started to offer some functionalities to see the "trending" items. For example, both Twitter and Facebook show trending topics and hashtags on a side panel, and Twitter has a tab for trending pictures. So some of the analysis work we previously

had to write scripts for is now available on the platform. However, understanding the content of a large body of social media text remains challenging. In the next section, I describe my proposed solution known as SentenTree.

4.3 SentenTree: visualizing textual content of social media

As we discovered from the Osama Bin Laden study, one of the remaining challenges to social media text analysis is to understand and summarize the content itself. Social media text collections are usually too large for the analyst to read through one by one. They typically contain many shared or repeated items and a lot of noise, so it makes sense to summarize them into a more condensed set before presenting to the reader.

There are two types of document summarization methods: extraction-based and abstraction-based [3]. Extraction-based summary methods extract key objects from the text collection without modification, while abstraction-based systems try to paraphrase sections of the source document. We feel extraction methods are more suitable for the social media text use case, since extracting condensed text segments instead of creating a full synopsis allows the system to cover diverse topics more efficiently, and extraction methods also present the original text which is critical for verification and reporting by human analysts.

There are typically two types of objects extracted by social media extraction techniques. The first type is representative sentences from the collection, where popularity usually reflects representativeness. For example, review sites such as Amazon.com often highlight the reviews that readers have voted as most "helpful", while social media sites such as Twitter recommend trending posts by the number of "likes" and shares among other signals. This popularity-based solution works well for some situations, but it runs a risk of lacking diversity. Research shows that the most popular messages on social media are usually produced by a small population of elite users or opinion leaders [89, 41]. Therefore, selecting the most popular messages usually means overlooking the voices of "ordinary" users of social media. In many scenarios, the opinions of ordinary people are precisely what the analyst wants to hear.

A second solution that considers diversity is to extract common information from

Figure 29: Part of a SentenTree visualization of a collection of 189,450 tweets (108,702 unique) posted in a 15 minute time window around the first goal of the opening game of the 2014 Soccer World Cup.

the entire document collection regardless of author. Numerous research efforts in the text mining community employ advanced rule-based and statistical methods (e.g., entity identification, topic modeling) to produce representative word lists or distributions and document clusters. The presentation of these outputs is almost inevitably some variation of a word cloud. While it makes sense to present topics using words, we argue that word clouds only give a sense of concepts, not more developed thoughts or opinions. Longer, connected phrases and sentences provide people with more complete ideas, thoughts, and sentiments of the document authors.

Many attempts to add more context or structure to word-based visualizations exist. One general approach is the use of semantic-preserving word clouds [28, 84, 91, 69]. Another is to connect words through visual structures such as lines [26, 55, 50]. Another approach, including systems such as Word Tree [85], Phrase Net [80], and Wordgraph [70], positions and links words spatially following their natural occurrences in sentences, thus better encoding thoughts and opinions. These projects inspire our design of a novel visualization technique for summarizing text documents.

We introduce the **SentenTree**, a novel technique for visualizing the content of unstructured social media text. SentenTree seeks a balance between showing the most frequent words and preserving sentence structure. SentenTree gives people a high-level overview of the most common expressions in a document collection and allows drilling down to details through interactions.

4.3.1 Design

Figure 29 shows an example of the SentenTree visualization for a Twitter dataset of 189,450 tweets commenting on the opening game of the 2014 Soccer World Cup, posted during a 15 minute time window around the first goal. A person looking at this visualization will immediately notice some prominent words like *first*, *goal*, *world cup*, *watching*, etc. Similar to a text cloud, the large font size of these words indicates their high frequency of occurrence in the dataset. These words are good indications that people were discussing the game between Brazil and Croatia. An edge between two words indicates their occurrence in the same tweet. By interacting with the visualization and following the linked words from left to right, the viewer can further identify patterns such as *first goal world cup own goal*. Hovering the cursor over the word *own* (Figure 30), highlights the expression *first goal world cup own goal*, fades other words, and displays example tweets containing these words in that order. Hovering over the word *score* (Figure 31), highlights another expression: *brazil marcelo score first goal world cup brazil*. These actions should help the viewer learn that the first goal of the World Cup was an accidental own goal by Brazilian player Marcelo against his own team.

Figure 30: Hovering over *own* in the World Cup visualization.

4.3.1.1 Design goals

In this section we discuss four goals that drove the design of SentenTree. The first design goal was to leverage the positive qualities of word clouds, namely their ability

Figure 31: Hovering over *score* in the World Cup visualization.

to facilitate fast impressions by utilizing size. Since the most frequent words are encoded in the largest font sizes, they jump out to the viewer. In the SentenTree visualization, we also wanted the frequent words and expressions to be displayed more prominently so they can be easily spotted by the viewer.

The second goal was to bring in more sentence structure from the items in the text collection. As researchers have repeatedly found, a bag of discrete words is limited in its expressiveness. Other approaches, such as the Word Tree [85], give context to a word through displaying sentence fragments next to that word. Review Spotlight [94] uses adjective-noun pairs to summarize opinions, which turned out to be much more informative than text clouds for the same data. For SentenTree, we use fragments extracted from sentences to represent those sentences. For example, a person should be able to reasonably guess the meaning of *first goal world cup own goal* without reading the full sentence *the first goal of the World Cup is an own goal*. These fragments are called frequent sequential patterns, and we will more formally define them in the next section.

We chose to use patterns instead of full sentences because of a third design goal: the visualization should be concise but yet cover as much of the dataset as possible. A concise pattern may summarize many sentences that are similar to each other but have slight differences. For example, in the previous example, the pattern *first goal world cup own goal* is shared by 14,935 tweets (Figure 30) and the pattern *score first goal world cup* is shared by 13,330 tweets (Figure 31). Frequent sequential patterns

are especially well suited to social media text because social media text collections on a given topic typically contain many sentences that share similar structures with small wording differences. The conciseness goal also suggests that we cannot show all frequent sequential patterns. Thus, the SentenTree technique collapses common parts of patterns to both save space and highlight their commonality. A reader familiar with the Word Tree might have noticed the similarity between Figure 29 and a Word Tree in that some large words have several branches extending from them on one or both sides. This is an indication that these big words are shared by several different sequential patterns.

Since our objective is to provide a high-level overview of the textual content, SentenTree follows the Shneiderman Mantra to provide an "overview first", then allow "zoom and filter" to get to "details on demand" [75]. By starting with the most common patterns, SentenTree addresses the entry point problem of the Word Tree [85]: instead of relying on people to identify their own entry point, the technique provides them with an overview of the most frequent patterns and allows them to select patterns of interest and drill down to see more details. This implies a fourth design goal on the technique: the pattern generation algorithm needs to be incremental.

4.3.2 Algorithm

4.3.2.1 Frequent Sequential Patterns

As discussed in the previous section, the central idea behind SentenTree is to take a large social media dataset, find the most frequent sequences of words, and build a visualization out from them. The sequences are called *frequent sequential patterns*. Below, we define the concept and then describe the sequence generation algorithm in detail. Because of the complexity of the algorithm, the following section provides an example to help explain its operations.

First, we formally define frequent sequential patterns for social media text based on a more general concept from data mining [74]. We consider a sentence a sequence

of words. Suppose we have two sequences $\alpha =< a_1 a_2 \ldots a_n >$, $\beta =< b_1 b_2 \ldots b_m >$ where $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m$ are words. We say that α is a subsequence of β, and β is a super-sequence of α, if there exist integers $1 \leq j_1 < j_2 < \cdots < j_n \leq m$ such that $a_1 = b_{j_1}, a_2 = b_{j_2}, \ldots, a_n = b_{j_n}$. This is denoted as $\alpha \subseteq \beta$.

The set of sentences containing the sequence α form the **support database** D_α. The **support** of α is the number of sentences in D. A sequence α is called a **frequent sequential pattern** when support of α exceeds some predefined lower threshold. In data mining, the threshold is called a **minimum support threshold** and is generally fixed for each mining task. In our case, the minimum support threshold is not fixed because we want to build frequent sequential patterns incrementally.

For example, suppose we have a database consisting of three sentences (sequences) s_1: *The first goal of the World Cup is an own goal*, s_2: *Someone somewhere has made a lot of money on first goal of World Cup being an own goal* and s_3: *Brazil's Marcelo scored the first goal of the World Cup.* The support of the sequence *first goal world cup* is 3 as it is a subsequence of all three sequences in the database, while the support of the sequence *first goal world cup own goal* is 2 because it is a subsequence of s_1 and s_2 but not s_3. If the minimum support threshold is 3, then only *first goal world cup* is considered a frequent sequential pattern. But if the minimum support threshold is 2, then both sequences are considered frequent sequential patterns.

Data: raw sentences
Result: graph
tokenized sentences = initialization(raw sentences);
create a pattern s without any word and make D_s = {all tokenized sentences};
list of leaf sequential patterns = patternGeneration(s, s, default word count on screen);
construct graph out of leaf sequential patterns;

Algorithm 1: graphCreation()

Data: root node of tree of sequential patterns, start pattern, number of visible words needed
Result: list of leaf sequential patterns
if *start pattern does not contain any word* **then**
  number of visible words needed -= number of words in start pattern;
end
leaf sequential patterns = empty list;
push start pattern to leaf sequential patterns;
while *leaf sequential patterns contains at least a word* ***and*** *visible word needed* > 0 **do**
  s = pop pattern with the largest support from leaf sequential patterns;
  if *s has no child sequences* **then**
    find the most frequent super-sequence s' of s that is exactly one word longer than s;
    split D_s into the support database for s' and a new $D_s = D_s - D_{s'}$;
    add s' as the left child of s, and the s with new D_s as right child of the old s;
  else
    s' = the left child of s;
    s = the right child of s;
  end
  number of visible words needed -= 1;
  push s' and s to leaf sequential patterns;
end
return leaf sequential patterns;

Algorithm 2: patternGeneration()

4.3.2.2 Sequential pattern generation and graph building algorithm

Algorithm 1 shows the process for constructing the graph data structure for the first time. The system first takes in raw sentences and goes through an initialization process (*initialization*()). This process normalizes sentences to lower case, performs tokenization to segment sentences into words (including numbers, hashtags, urls, etc.), and filters out stop words. Then an initial pattern is created without any words, and all tokenized sentences are put into its support database. This initial pattern serves as the root node of a tree of sequential patterns, which we use to store intermediate states of the sequential pattern generation process. This tree is important because we can reuse these intermediate states when zooming in and out on the visualization.

Then the algorithm calls the *patternGeneration*() function to grow new sequential patterns from the root one.

The task of *patternGeneration*() (Algorithm 2) is to start with a given sequential pattern and grow its super-sequences until the total number of words in the patterns reaches the given number of visible words needed. These patterns will appear in the visualization and the given number of visible words needed is determined by the screen size (typically 100-200). The algorithm grows patterns by maintaining a list of leaf sequential patterns. At first, the only item in the list is the given sequential pattern. In every run, the most frequent pattern is popped from the leaf pattern list, and the program finds its most frequent super-sequence which is one word longer than the old pattern. This means a new word will be added to the visualization and the number of visible words needed is reduced by one. The new sequential pattern becomes the left child of the old sequential pattern on the tree, and a pattern that looks exactly like the old one is added as the right child of the old sequential pattern. The support database of the old pattern is split between the new patterns, and the new patterns are added to the list of leaf patterns. Therefore, at anytime, the original database is split between the support databases of all the leaf patterns. The program continues growing new leaf patterns until the number of visible words needed is reduced to zero.

Function *patternGeneration*() returns the list of leaf patterns to function *graphCreation*(), which uses these leaf patterns to construct a graph data structure for the visualization. In the graph data structure, each node is a word in the sequences; a word shared by multiple leaf patterns appears as one node. A directed edge is added between every pair of adjacent words in a leaf pattern.

After the visualization is created, the user may wish to zoom in on a frequent sequential pattern and bring up more children patterns. In this case, the program calls *patternGeneration*() with the selected sequential pattern as the start pattern and starts growing new patterns from there. The algorithm is usually able to reuse

some patterns from the sequential patterns tree so it does not have to recalculate patterns that were discovered before. When the leaf patterns are returned, the program constructs a new graph from the leaf sequential patterns in the same way described in the previous paragraph.

Note the goal of this algorithm is different from a typical sequential pattern mining algorithm. Instead of trying to find **all** frequent sequential patterns based on a minimum support threshold, this algorithm only grows sequential patterns from existing patterns for building the graph visualization. Most algorithms in sequential pattern mining follow a depth-first approach to find sequential patterns because they want to enumerate all patterns satisfying a lower limit frequency. Our approach is breadth-first, and will take more time to perform than the depth-first approach if our goal is to exhaust all possible patterns. However, we only need a limited number of patterns each time due to display limits, therefore the breadth-first approach works well for its incremental quality.

Example - World Cup First Goal

We use a simplified World Cup example to illustrate our approach. Figure 32 illustrates the following steps.

We start with a dataset of 189,450 tweets. We first normalize the tweets to lower case, perform tokenization to segment each tweet into discrete words (including hashtags, urls, etc.) and remove stop words and punctuation.

1. We find the most frequent single word pattern (i.e. the most frequent word) in the dataset to be *goal*, and divide the dataset into tweets containing *goal* (74,554) and tweets without *goal* (114,896).

   leaf patterns: *empty pattern* (114,896), *goal* (74,554)

2. Next we pick the most frequent leaf pattern which is the empty pattern with 114,896 tweets. We find the most frequent single word pattern within these

tweets to be *watching* (11,248) and add it to the tree, we also end up with an empty pattern with a support of 103,648.

leaf patterns: *empty pattern* (103,648), *goal* (74,554), *watching* (11,248)

3. The most frequent leaf pattern is still the empty pattern. For illustration purpose we will ignore it from now on and pick the pattern *goal*. We will grow our pattern by one by finding the next most frequent pattern in the subset of tweets containing *goal*. The new sequential pattern is *first goal* (41,344). Note that the new pattern will always contain the previous pattern, as all tweets in this subset contain the previous pattern. We also end up with a pattern *goal* without a *first* before it and this subset has 33,210 tweets.

   leaf patterns: *first goal* (41,344), *goal* (33,210), *watching* (11,248)

4. The most frequent pattern is *first goal*. Create *first goal world*.

   leaf patterns: *first goal world* (36,136), *goal* (33,210), *watching* (11,248), *first goal* (5,208)

5. The most frequent pattern is *first goal world*. Create *first goal world cup*.

   leaf patterns: *first goal world cup* (36,081), *goal* (33,210), *watching* (11,248), *first goal* (5,208), *first goal world* (55)

6. The most frequent pattern is *first goal world cup*. Create *first goal world cup own*.

   leaf patterns: *goal* (33,210), *first goal world up* (21,220), *first goal world cup own* (14,861), *watching* (11,248), *first goal* (5,208), *first goal world* (55)

7. The most frequent pattern is *goal* (33,210). Create *own goal*. Note that in this new pattern *own goal* we have a word *own* and in a previously generated pattern *first goal world cup own* we also have a word *own*. But these two words

are not considered common branches in the SentenTree and will be represented by two distinct nodes in the final visualization.

leaf patterns: *first goal world cup* (21,220), *goal* (19,977), *first goal world cup own* (14,861), *own goal* (13,233), *watching* (11,248), *first goal* (5,208), *first goal world* (55)

8. The most frequent pattern is *first goal world cup*. Create *score first goal world cup*. Note that each time we grow a new pattern, a word is added to the existing pattern. The new word can appear before, behind or in-between the words of the parent pattern.

   leaf patterns: *goal* (19,977), *first goal world cup own* (14,861), *score first goal world cup* (13,291), *own goal* (13,233), *watching* (11,248), *first goal world cup* (7,929), *first goal* (5,208), *first goal world* (55)

The leaf patterns *first goal world cup own*, *score first goal world cup*, *own goal*, and *watching* are used to construct a graph structure for the final visualization (Figure 33). Note that *first goal world cup own* and *score first goal world cup* share a subsequence *first goal world cup*, and this subsequence also shares the word *goal* with *own goal*. While *watching* is not connected to the other words and forms its own graph. We will discuss the layout and visual encodings of the graph in the next section. The different sequential patterns can be highlighted by hovering the mouse over one of the words. We will discuss the interactions in detail in a later section.

4.3.2.3 Additional Considerations

"Big words"

As described in previous sections, SentenTree prioritizes the most frequent sequential patterns and grows new patterns out of the existing ones. This introduces a problem in some scenarios: a large pattern may dominate the view and prevent

Figure 32: An example pattern generation process. The words in boldface are new words added to the parent pattern to generate the current pattern. The numbers in parenthesis are the support of each sequential pattern.

Figure 33: A simple SentenTree of top sequential patterns in the World Cup dataset.

other interesting patterns from surfacing. This is especially likely to happen when the dataset is retrieved based on particular keywords, therefore those keywords exist in most entries in the dataset. Since the person using the system is already familiar with the keywords, we suspect they are not crucial to the view. Additionally, their large size and visual dominance may obscure useful new information. Figure 34 shows what happens with the World Cup dataset used in the teaser image (Figure 29) without deprioritizing *world cup*. Note how *world cup* is huge and makes many branches small and difficult to read. Only one graph is in the view, so it generates significant white space, compared to Figure 29, where multiple graphs fill up the screen.

We address this problem by imposing a rule that words that appear in a large of

number entries cannot be used to generate the first pattern of a graph. After testing a variety of values, we set the cut-off to be one third of the database size. This rule ensures that the view never ends up with a huge pattern that is significantly larger than the rest of its branches. These patterns will still show up in the visualization, but they will appear in multiple graphs so each one is smaller and does not skew the other words.

Figure 34: A SentenTree visualization of the same dataset as Figure 29 without deprioritizing *world cup*.

Tokenization and filtering

Tokenization segments sentences into words. Our implementation employs a regular expression pattern to match not only words but also numbers, hashtags, urls, @ handles, etc., because they are prevalent in social media text. The current implementation of SentenTree does not perform stemming; therefore, words like *watch* and *watching* are considered different tokens. We are considering including a stemmer for future versions of SentenTree.

The SentenTree algorithm does not include punctuation as tokens. Additionally, it discards stop words such as "the", "and", "of", etc [1]. These choices are opposite as done in the Word Tree, which keeps both stop words and punctuation to best preserve the context for each word. We decided to remove stop words and punctuation because

[1]SentenTree uses a modified list based on the English stop words corpus from the Natural Language Toolkit (NLTK).

SentenTree is a high-level summary of the text. Stop words and punctuation are not likely to be as informative as substantive words in the dataset. Furthermore, common stop words and punctuation may "wash out" the more important content words. Note that negation words like "not" are not considered stop words because taking them out would reverse the meaning of sentences.

Currently the algorithm does not filter out hashtags or urls, but we have noticed that in general they are less informative than regular words and may be a waste of screen real estate, so we plan to include options to filter them out in the future.

4.3.3 Visual design and interactions

4.3.3.1 Spatial layout and visual encodings

We produce a visualization from the graph structure generated in the previous section. The graph visualization is created through a force-directed approach. Each graph is its own SVG element so that the layout algorithm can run in parallel for multiple disconnected graphs. The SVG elements are ordered by the frequency of the largest word in the graph and packed as tightly as possible on the screen.

We developed alignment constraints for the force-directed layout and enforce them using the CoLa package [31]. The most basic constraint is the **word order** constraint: if two words appear in the same sequential pattern, the relative horizontal placement of words must follow their natural order in the pattern. This promotes that a person can read a pattern from left to right and understand its meaning. By experimenting with initial layouts, we further developed vertical and horizontal constraints that increase the legibility of the graph:

vertical: If two words always appear as a bigram, then we shorten the link between the two words and make sure they always appear on the same vertical level. An example is the words *world* and *cup* in Figure 29.

horizontal: If two patterns share some a common subpattern, then the words of the same distance from the common subpattern should center horizontally. An

example is Figure 35b where the words *way*, *buys*, *picks*, etc. are centered horizontally because they all appear next to *yelp*.

These constraints not only make the graph layout less cluttered, but they also impose structure on the layout so that words in similar syntactic positions align vertically and can be compared against each other. The power of these layout constraints is demonstrated in Figure 35. This dataset consists of tweets mentioning the food ordering service Eat24 shortly after it was acquired by Yelp. (Only part of the visualization is shown in Figure 35.) Figure 35a shows the result of running the force-directed layout without the horizontal and vertical constraints, and Figure 35b shows the result with the constraints. Note that alignment yields useful information: the observer can see that when people discuss the acquisition, they use many sentences that are similar in form and meaning but vary slightly in wording. For example, they use *picks* and *gobbles* in place of *buys*, and describe Eat24 as *food ordering service* or *delivery network*.

(a) Force-directed layout using only the left-to-right constraint.

(b) Force-directed layout with horizontal and vertical constraints added.

Figure 35: Part of a SentenTree visualizations of tweets discussing Yelp's acquisition of Eat24.

We use font size and color shading to double-encode the frequency of occurrence.

We make the font size of a word proportional to the square root of the number of text documents containing the sequential pattern where the word first arises. Using the simplified World Cup example from the previous section, the size of *goal* is proportional to the square root of the number of tweets containing the pattern *goal*, while the size of *cup* is determined by the square root of the number of tweets containing the pattern *first goal world cup*. More frequent words are in a darker shade of blue than less frequent words. We also made it optional to turn the first and last word in a pattern to purple in order to distinguish the beginning and ending of patterns. Some of the use cases in this paper have that option turned on. We are considering other options for word color, such as encoding the sentiment of words, but the benefit provided by such a change must be weighed carefully against the visual variation and inconsistency it introduces. We also experimented with varying the width and shade of the edges but found that this introduces more visual clutter than useful information, so we decided to render the edges as thin light-gray curves.

4.3.3.2 Interactions

Since we are placing sequential patterns in a graph, a problem arises in that people viewing the visualization often cannot tell where a sequence starts and where it ends. As shown in Figure 33, someone viewing this visualization might assume a pattern *score first goal world cup own* exists, but in reality the dataset only contains *score first goal world cup* and *first goal world cup own* which are connected by the common part in-between. This problem is also present in Double Tree (Word Tree on both sides) [29] and Wordgraph [70].

We address this problem by using interaction. A person can hover the mouse over a word and all other words besides those that appear in its sequential pattern will become semi-transparent. The highlighted sequential pattern is the most frequent "leaf pattern" containing the selected word. (A "leaf" pattern is a pattern without a

longer super-sequence on the screen.) Most words in the leaf pattern are colored in light blue but the words that appear as many or more times than the selected word are colored in dark blue. These words form the most frequent pattern containing the selected word, and a tooltip pops up showing the frequency of this pattern.

When a person hovers the mouse over a word, the system also displays the most common example sentences (e.g., Tweets) containing it in the lower left of the window, as shown in Figure 30 and Figure 31. This helps the viewer learn more about precise thoughts and opinions in the text.

We also enable drilling down to an existing sequential pattern to see more details. When a person clicks on a word, SentenTree zooms in to the most frequent pattern containing the selected work (the pattern is colored in dark blue) and filters out all other sequential patterns. SentenTree also grows the current sequential pattern to include new words. An example is Figure 37a. The viewer clicks on *penalty* and all other branches disappear. The branch with *penalty* in the center grows out to fill the screen. The viewer can click on a RESET button in the interface (not shown in the figure) to go back to the full view.

4.3.4 Implementation and performance

We have implemented the SentenTree algorithm in Javascript for the web. In this implementation, a person provides raw text (e.g., tweets) to the application. All following computations are performed in the browser on the client side. We use d3.js[2] for the visualization and cola.js[3] for the constraint-based force-directed layout.

The pipeline of SentenTree can be broken into three steps: 1) load and preprocess (e.g. tokenize) the input data, 2) extract frequent sequential patterns and construct the graph data structure, 3) visualize the graphs on the screen. When a user interacts with the view by drilling down to show more details, we repeat steps 2 and 3, though

[2]http://d3js.org/
[3]http://marvl.infotech.monash.edu/webcola/

the patterns generated from a prior step 2 can be reused.

The runtime of step 1 is linear to the total number of words in the dataset. Because the number of words in a sentence (or other social media text unit such as a Tweet) is limited, the runtime of step 1 is roughly linear to the number of sentences in the dataset. The runtime of step 2 is linear to the product of the number of sentences and the number of words in the final visualization. Because we limit the number of words shown due to available screen space, the runtime of step 2 is also roughly linear to the number of sentences in the dataset. The runtime of step 3 is more difficult to estimate, because the constraint-based force-directed layout is influenced both by the number of words and edges in the graph as well as the complexity of the graph. The layout algorithm runs in parallel for multiple disconnected graphs, so the time consumed is dependent on the largest, most complicated graph.

We tested the technique on datasets with 10,000 (10K), 100,000 (100K), and 1,000,000 (1M) unique sentences in a Google Chrome Browser on a MacBook Air laptop. The number of words shown in the visualization was fixed at 150. For the 10k datasets, step 1 took 0.25 to 0.45 seconds, and step 2 took 0.8 to 2 seconds. For the 100k datasets, step 1 took under 3.1 seconds, and step 2 took 11 to 17.5 seconds. For the 1M datasets, step 1 took under 30 seconds, and step 2 took under 1 minute. The runtime for step 3 is not related to the input data size, but is dependent on the most complicated graph in the visualization. For all of the datasets we tested, the most time-consuming graph layout took under 6 seconds, though we observed that layout started to stabilize after the first half second and only made minor movements afterwards. Thus, the effective total duration to display each of the three sizes of data was about 2 seconds, 20 seconds, and 2 minutes.

Figure 36: A SentenTree visualization of tweets commenting on the second goal of the opening game of the World Cup. This dataset contains 135,841 tweets (81,253 unique tweets).

(a) The full view of the dataset.

(b) A zoomed-in view focused on *penalty* after the viewer has clicked on that word.

Figure 37: A SentenTree visualization of tweets commenting on the third goal of the opening game of the World Cup. This dataset contains 132,599 tweets (75,930 unique tweets).

4.3.5 Example use cases

4.3.5.1 A typical exploration scenario - World Cup

In this section, we describe a use case of exploring comments on an event through SentenTree visualizations. Suppose a person wishes to learn about Twitter users' reactions to the opening game of the 2014 Soccer World Cup between Brazil and Croatia. The person already knows that multiple goals were scored during the game, each creating a huge spike in tweet volume. For each goal, the person is able to obtain 15 minutes of relevant tweets using keywords and hashtags like "world cup", "#worldcup2014", "#bravscro", "#bra", "#cro", etc. We describe how she explores the tweets for the first three goals of the game.

First Goal

The SentenTree visualization for the first goal is presented in Figure 29. As previously discussed, the person immediately notices a large pattern *first goal world cup*. Hovering her mouse over *cup* tells her that this pattern appears 36,081 times. In other words, close to 40% of tweets about the World Cup posted in this 15 minutes time window include the pattern *first goal world cup*. As she hovers the mouse over a few other branches, she notices patterns like *brazil marcelo score first goal world cup* and *first own goal world cup history*. After exploring more branches and reading some example tweets, she concludes that most Tweets were either describing what happened on the soccer field or expressing excitement over the goal and surprise at Marcelo's big blunder.

Second Goal

The person moves on to the second goal of the game (Figure 36). She immediately notices that the biggest branch is centered around *neymar*. Hovering over the branches she notices people mentioning that Brazilian player Neymar had scored the second goal of the game, which brings the score to 1-1. She also notices branches like *yellow Neymar* and *Neymar card*. She clicks on them to bring up example tweets explaining that Neymar had drawn a yellow card minutes before he scored the goal. The person clicks on *Neymar* to bring up a detailed view. This view directs her to longer patterns about Neymar which she explores for a while, learning that at age 22 Neymar was considered a "boy wonder" and this was the 32nd goal he scored for Brazil, making him the third highest Brazilian goalscorer.

Zooming back to the full view, she also notices a branch centered around *goal*. Hover her mouse over the branches, she finds patterns such as *own goal*, *score own goal*, and *marcelo goal* which indicates that people were still discussing the first goal of the game being an own goal. She also saw a big branch under *game* which reads *first game world cup tonight wait par coma majooooorr*. Intrigued, she clicks on the branch

to bring up an example tweet and discovers that people were retweeting an earlier tweet by Niall Horan which says "First game of the World Cup tonight! Can't wait! PRA CIMA MAJOOOOORR! CMON BRAZIIIILLLLL!". Niall Horan is a member of the popular boy band One Direction and one of the most followed celebrities on Twitter. Therefore it is not surprising that his tweet was retweeted so many times.

Third Goal

The person moves on to the third goal of the game (Figure 37a). She sees that a large branch is centered around the word *penalty* and another branch is centered around *neymar*. Hovering the mouse over a few branches she learns that Neymar scored a penalty kick for Brazil making the score 2-1. She is interested in learning more about people's reaction to the penalty goal, so she zooms in on *penalty*. Figure 37b shows a detailed view with *penalty* at the root. The person notices some new words connected to *penalty*, such as *bad*, *soft*, *never* to its left and *unbelievable* and *bad* to its right. She hovers the mouse over these words to see the branches and also brings up a few example tweets. She learns that the penalty decision was controversial, as Twitter users call it "wasn't a penalty" ,"never a penalty", "soft penalty", "bad penalty", "worst penalty decision", and even "unbelievable".

Zooming out and looking at the full view, the person notices that Niall Horan's tweet was still being retweeted by many followers. She also notices a tweet by @gary-lineker with many retweets. Clicking on the branch she brings up the original tweet. It appears to be a rather unflattering joke towards FIFA president Sepp Blatter which was echoed by over 3,000 Twitter users. (@GaryLineker: "I like this vanishing spray FIFA are using for the World Cup. Would it work on Sepp Blatter?")

By going through the three goal visualizations, the person was able to form a coherent narrative of not only what happened on the field, but also how people reacted to each goal. She discovered that people were shocked by the first own goal, applauded the second goal by Neymar, and were unhappy with the penalty goal which

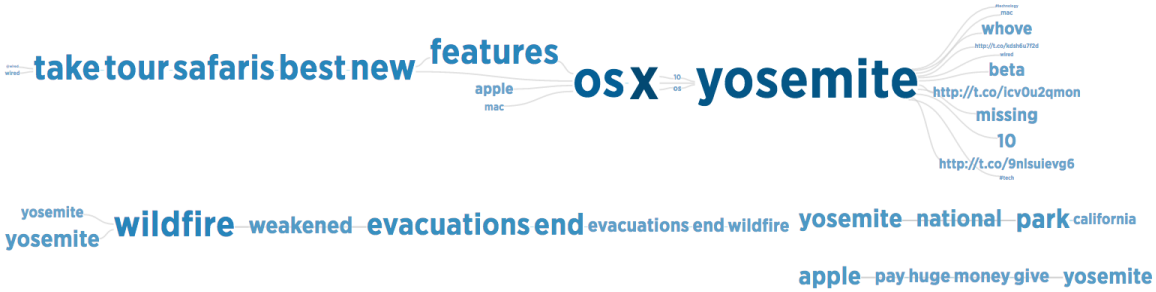


Figure 38: A partial SentenTree visualization of an entire day's (August 1st, 2014) tweets on the subject "yosemite". The dataset contains 6,712 tweets (4,963 unique tweets).

Figure 39: A SentenTree visualization of Amazon.com Reviews on a Samsung TV model. The dataset contains 109 reviews with 941 sentences in total. The unit of analysis is a sentence.

gave Brazil a lead in the game. They were excited by the dramatic opening game and expectant of the rest of the World Cup.

This is a typical use case of the SentenTree visualization for exploring a large dataset (or datasets). In the initial stage, SentenTree supports impression-forming just as a word cloud. In addition, it provides context for each word by placing the words within sequential patterns and allows easy highlighting of patterns through hovering. When interested in a sequential pattern the viewer can click on it to drill down to see more details. The viewer also can click on sequential patterns to bring up example sentences. Because Tweet-reading typically happens after the viewer has formed initial expressions of the dataset, the person only needs to bring up sentences that look relevant instead of drowning in a sea of text.

4.3.5.2 Natural content clustering

In the previous section, we illustrated the use of the SentenTree visualization to explore social media reactions to a high-profile event. Based on our experiences applying SentenTree to different datasets, we have observed that for datasets with

somewhat diverse content, SentenTree often results in natural clustering of texts. In this section, we give three examples of natural clustering and discuss how SentenTree can be used to explore these datasets.

4.3.5.3 Communicating multiple word meanings

We obtained a dataset of tweets posted on Aug 1, 2014 with the keyword "yosemite". As *yosemite* is present in every tweet in the dataset, it is not allowed in the first step of sequential pattern generation. The resulting visualization is shown in Figure 38. Yosemite is a word with ambiguous usage, and this is reflected in the visualization. We observe that the biggest branch is about OS X Yosemite (a version of the Mac operation system). Another smaller branch is also about the Mac OS with *Apple* as the most prominent word in the branch. Another usage of the word "yosemite" is the name of a national park in California. We find one branch with *yosemite national park california* and another branch about a wildfire in the park. SentenTree is able to separate tweets on the Mac OS and the national park without any semantic analysis.

4.3.5.4 Communicating multiple concepts

A SentenTree visualization of tweets about the startup Eat24 retrieved shortly after Yelp's announcement to acquire the company resulted in three clusters. (Please refer to the supplement material for the image.) The first is a branch focused around the sequential pattern *yelp eat24* about the news of the acquisition worded in slightly different ways. The second is a branch with *@eat24* at the center that appears to be Eat24's official Twitter account which interacts with its customers frequently. From the visualization we learn that customers communicate with @eat24 to talk about the mobile app, coupons, their order, etc. A final small branch has *eat24* without *yelp*. Expanding the branch, we discover that tweets in this branch are focused on Eat24's Super Bowl commercial with rapper Snoop Dogg.

4.3.5.5 Communicating multiple facets

So far, all examples we have shown are tweets. We include a final dataset of consumer reviews to illustrate another application of the technique. We include this case to demonstrate SentenTree's power to highlight different facets in a dataset. The dataset includes reviews crawled from Amazon.com about a Samsung TV model. As the unit of analysis is a single sentence, we segment the reviews into sentences before feeding them to SentenTree. We also specify a rule that *samsung* and *tv* should not be used in the first round when generating sequential patterns as we do not want these words to dominate the view.

Figure 39 is the SentenTree visualization of the dataset. The visualization consists of several branches each having a bigger word in the center and a set of smaller words on both sides. These branches are reminiscent of a double-sided Word Tree (e.g. the Double Tree). This indicates that the data set does not have longer frequent sequential patterns. The likely reasons are 1) the dataset is considerably smaller than the previous tweet datasets and there simply are not enough sentences to form similar long sentences, and 2) in consumer reviews people tend to write longer paragraphs with more variations in their expressions than what people tend to write on Twitter.

Although there are no long patterns, the branches give indications to different facets people frequently mention. There are *picture*, *sound*, *amazon*, *lcd* which are all product/services facets of the TV. By browsing the words used together with each facet word, the viewer is able to learn what people like and/or dislike about each facet. Major branches with *great* and with *good* at the center are evident. By looking at these branches the viewer is able to figure out that reviewers report positive feelings about the picture, the price, Amazon, and the TV in general.

Consumer reviews research suggests representing facets as nouns and using adjectives or phrases close to the nouns to describe each facet [94, 46, 43]. Although SentenTree does not perform parts-of-speech analysis, facets (such as "picture", "sound",

etc.) naturally emerge because the words are used frequently by reviewers. Additionally the words that frequently co-occur with facet words are shown on either side of the facet words. We argue that SentenTree serves well as a generic tool for someone casually exploring a review dataset, especially considering it does not perform complex natural language processing.

4.3.6 Domain expert feedback

4.3.6.1 Study process

We showed SentenTree to three domain experts to gain initial feedback about the system. All three were research scientists or data scientists at technology companies who regularly work with social media text. They used statistical analysis tools and simple visualization tools in their work. We gave each expert a short demo of SentenTree and asked them to use the system to analyze a few different Twitter datasets. We observed how they interacted with the visualization and asked for feedback at the end of the session.

All three people immediately understood the correspondence between font size/color and frequency, and all of them spent significant time hovering the cursor over different words in the view to read example tweets. The participants had to be reminded of the less discoverable functionalities (such as clicking to drill into a pattern, searching, and zooming). It also initially surprised them when they saw the same word multiple times in the visualization, but once they were given an explanation and used hover to check that the word appeared in different patterns, they became comfortable with the concept. The experts agreed that SentenTree has a learning curve, and it will take some practice for a new user to master the tool completely.

4.3.6.2 Findings

The aspects of the technique receiving most positive comments include:

- The most frequent words and patterns "pop out" in the visualization because

they are larger and darker, and the frequency count can be read in a tooltip.

- The context to a word or pattern is immediately available by cursor hover.
- Similar content is grouped into a graph or branch that makes it easy to form an impression of the major topics.
- The visualization can serve as a filtering tool and greatly saves time in finding interesting content to read.
- Less frequent content is not eliminated; instead it is put into small but discoverable branches.
- One can keep drilling down into interesting branches and search for words on the screen.

The participants also pointed out a few items to improve on or to add to SentenTree:

- A common request was to perform stemming and spell checks, and to merge different forms of the same word into a single item.
- Improvements to the UI such as packing the graphs tightly to reduce white space and sorting vertically parallel branches by frequency or alphabetic order.
- Perform sentiment analysis and color-code words by sentiment.
- Render the example text in its original form. Some of the tweets they saw contained images that provide important context and it may be helpful to render them inline.
- Reduce the noise in the visualization, e.g., provide an option to filter out certain words such as hashtags.
- Provide the ability to filter on meta-information. For example, one of the people works with geographic information regularly and asked for the ability to filter text based on author location.

4.3.6.3 Discussions

All three experts found SentenTree enjoyable to use and commented that it added fun to reading text. They also found SentenTree effective for helping to explore text datasets. As discussed previously, they pointed out it introduces some learning curve but agreed they will become comfortable with the tool after some practice.

One interesting observation we made from discussing with the experts is they spent the biggest chunk of their time reading individual raw tweets. When we described SentenTree they considered it not a tool to give them answers, but a tool to help them find interesting tweets to read. Here are quotes from two of the experts we interviewed:

- *SentenTree can be a good organization tool for my documents.*
- *It helps me find which tweets to read.*

This finding might be slightly surprising to people who are trying to build tools that tell analysts the *result* based on a text document. But it is consistent with previous studies conducted by Yatani et al. [94] who pointed out users do not trust computational results, especially if the results contain errors as is inevitable with NLP. For data scientists who care deeply about the accuracy of their analysis, it is very important they can confirm the computational results by reading the raw text and forming their own opinions. They also need to include some sample text when reporting their work. Therefore, SentenTree's model of exploring the data set to find sample tweets fit very well with their work flow.

4.3.7 User study

To gain a better understanding of SentenTree's performance on different analytical tasks, I conducted a study with 50 participants comparing SentenTree against a benchmark text summary method. The benchmark method is a set of the most

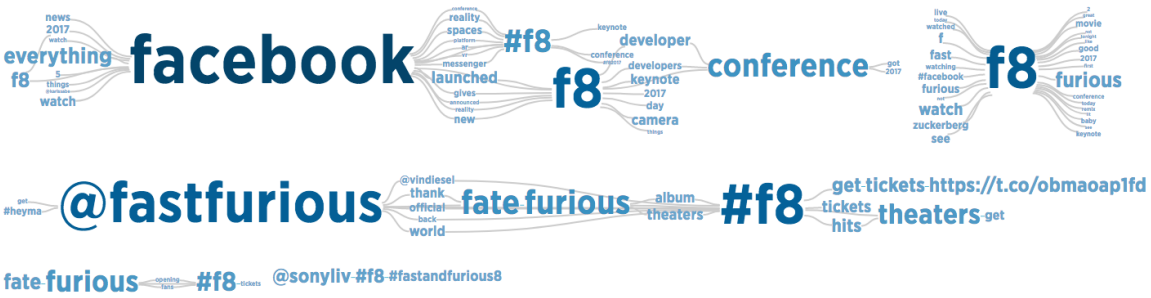


Figure 40: SentenTree visualization of the "f8" dataset used in the user study.

popular documents extracted from the dataset. I examined how subjects performed different analytical tasks using the two methods.

4.3.7.1 Experiment Design

The dataset for this study are tweets mentioning the tag "f8" posted between April 12 and 20 in 2017, gathered through the public Twitter API. The dataset contains 56,260 unique tweets from a total of 113,289 tweets including re-tweets. This dataset was selected because it includes many diverse topics and opinions since #f8 was used to denote two completely different events that coincided: the 8th Fast and Furious movie and the Facebook Developer Conference F8.

I performed the study with Mechanical Turk workers. Workers were required to be living in the United States and had more than 1000 HITs approved with greater than 95% approval rate. They were compensated $5 each for the study.

25 Mechanical Turk workers were randomly assigned to the SentenTree group and 25 workers to the List group. Workers in the SentenTree group were given instructions of SentnTree and asked to load SentenTree in their browser (Figure 40 shows the SentenTree visualization with the dataset loaded). Workers in the List group were instructed to open a Google Spreadsheet of the top 242 tweets with most re-tweets from the dataset, ordered by the number of re-tweets (Figure 41). Workers in both groups could interact with the tool given to them.

The hypothesis of the study is that workers would perform better on analytical

1	rt @pitbull: #tbt crafting 'hey ma' with @camila_cabello and @jbalvin for @fastfurious . the fate of the furious is in th	1446
2	rt @fastfurious: the fate of the furious the album is out now. #f8 https://t.co/xa6tu1ckg9 https://t.co/vgebhovbir	1249
3	rt @fastfurious: thank you for making the fate of the furious the biggest global opening of all time! #f8 https://t.co/by	1153
4	rt @thereallukevans: who said he was dead? not me... #f8 #owenshaw https://t.co/xoadc87pra	1109
5	rt @fastfurious: critics are calling the fate of the furious an “action masterpiece.” #f8 opens friday. get tickets: https://	1029
6	rt @fastfurious: saving the world in style. #f8 is now playing. https://t.co/cifmzyfwxk	1009
7	rt @jonnysun: later fast and furiouse sequels:- the f8 of the furious- the f9al fantasy of the furious- why helo there fas	968
8	rt @fastfurious: no one’s ready for this. watch the new trailer for the fate of the furious now! #f8 https://t.co/3xetv3y	921
9	rt @fastfurious: it’s official. #f8 is the number one movie in the world. get tickets now: https://t.co/obmaoap1fd https	829
10	rt @fastfurious: #f8 opens tomorrow! get tickets: https://t.co/obmaoap1fd https://t.co/cvft8x2be9	730

Figure 41: A list view of the top tweets from the“f8” dataset used in the user study.

tasks with SentenTree than with the List. To test this hypothesis, I asked workers in both groups to answer the same survey with the help of SentenTree or the List. The survey questions were designed to represent primary analytical tasks gathered from our previous research with real-world tasks and professional analysts: impression forming of high-level topics, identifying interesting facets, and finding substantiating evidence. The survey was organized in the following way:

1. Find high-level topics
    - Question: Identify two meanings of ”f8” (8th Fast and Furious movie and Facebook Developer Conference)
    - Rate how easy it was to answer the question
    - Rate how confident you feel about your answer
2. Identify interesting facets related to the movie
    (a) Find facts:
    - Sanity check question: Find name of the movie
    - Rate how easy it was to answer the question
    - Question: Find two people related to movie and explain why they are mentioned
    - Rate how easy it was to answer the question

(b) Understand opinions:

- Question: Find one positive and one negative opinion and provide two tweets to support each opinion
- Rate how easy it was to answer the question

(c) Identify interesting details:

- Question: Find three interesting points and provide a tweet to support each
- Rate how easy it was to answer the question

3. Identify interesting facets related to the conference

(a) Find facts:

- Sanity check question: Finding name of the leader of the company
- Rate how easy it was to answer the question
- Question: Finding three technologies related to company and explain why they are mentioned
- Rate how confident you feel about your answer

(b) Understand opinions: (same as for the movie topic)

(c) Identify interesting details: (same as for the movie topic)

Questions in the survey cover tasks including topic summary, fact-finding, opinion-finding, and open-ended identification of interesting points. For opinion-finding and open-ended questions, workers were asked to paste one or two tweets to support their answers. After answering each question, workers were asked to rate on a Likert scale of 5 how easy it was to answer that question with the help of the visualization or the list. The questions were presented in the following order: first, workers were asked to identify two high-level topics from the dataset (which we expect to be the movie

and the conference). Then for the movie, they were asked to answer two fact-finding questions, one opinion-finding question, and one open-ended question. Next, they were asked to respond to the same set of questions for the conference. We designed the question order based on findings from a pilot study which showed that tweets about the conference were harder to find than tweets about the movie. By first asking workers to identify high-level topics first, then focus on the movie, and finally on the conference, we made sure early questions did not give away answers to later questions.

4.3.7.2 Results

For each question in the survey, we examine at 1) the open-ended question results, 2) the "easiness" rating and 3) the confidence rating (if available). Some of the open-ended questions have a clear ground truth (e.g., the name of the movie). For these questions, we check the accuracy of the answers. Other questions are more open-ended and can have many different answers that make sense. For these, we used two external graders familiar with both the topics and dataset to check all answers. First, they made sure the responses provided by workers reasonably answer the question. For example, if the question asks for a positive opinion related to the movie, the answer should be both positive and related to the movie. The graders also check the provided tweets to make sure the answer is supported by evidence. Another metric they check is how *diverse* the answers are. We chose this metric based on the assumption that analysts are interested in diverse and especially unexpected facets of the dataset, so a tool that lets them discover more unique ideas from a dataset with a similar amount of work is preferred. To quantify the number of unique ideas, the two graders independently examined the response to each question and developed a list of unique ideas from all responses. Then, they consolidated their lists. Finally, they used the consolidated list to count the unique number of ideas presented by each

group.

Once all answers had been normalized to a numeric value (e.g., how many unique ideas were identified by workers in the SentenTree group), I tested whether there is a statistically significant difference between the SentenTree group and the List group using a Two Sample t-Test (two-tailed with 95% significant level). For all results that are significantly different between the two groups, I report the t-value and p-value.

Table 4 shows the quantitative results of the study. Starred items are statistically significantly different between the SentenTree group and the List group.

Identifying high-level topics

The first question in the survey asks workers to identify two meanings of the tag "f8". We expect workers to identify both the 8th Fast and Furious movie and the Facebook Developer Conference.

We found all 50 workers answered correctly about the movie. Out of the 25 workers who used the list of top tweets, 5 missed the Facebook meaning altogether, and another 3 mentioned something about Facebook but did not get the conference part. Out of the 25 workers who used the visualization, 2 of them missed Facebook entirely, and one answered Facebook but missed the conference.

We also asked workers to rate how easy it was to find the answer using the visualization or list on a Lickert scale of 5 (with 5 being *very easy* and 1 being *very hard*) and how confident they felt about their answers (with 5 being *very confident* and 1 being *not confident at all*). We found that workers rated the visualization as more easy to find the answer and workers in the List condition are slightly more confident about their answers (Table 4), though the differences for neither scores are statistically significant. Interestingly, self-rated confidence does not correlate with accuracy at all. Out of the 7 people who missed the Facebook meaning completely, 5 said they were *very confident (score 5)* or *somewhat confident (score 4)* about their answers. This seems to suggest that self-reported confidence is not a good indicator

Table 4: Summary of results from the user study. (FF) indicates questions focused on the Fast and the Furious movie and (FB) indicates questions related to the Facebook conference. Starred items are significantly different between the SentenTree visualization and the List. Likert responses for ease ranged from 1 - very hard to 5 - very easy, and foro confidence ranged from 1 - not confident at all to 5 - very confident.

	SentenTree visualization	List of top tweets
Identify topics (FF)	25 correct	25 correct
Identify topics (FB)	22 correct, 1 partial, 2 miss	17 correct, 3 partial, 5 miss
Easy?	4.04	3.44
Confident?	4.44	4.48
Find facts sanity (FF)	25 correct	25 correct
Easy?	4.44	4.64
Find facts sanity (FB)	22 correct, 3 partial	23 correct, 1 partial, 1 miss
Easy?	4.76	4.48
Find facts (FF)	10/11	8/11
Easy?	2.8	2.88
Find facts (FB)	18/24	14/24
Easy?	3.16*	2.36*
Find opinions easy? (FF)	1.68	1.36
Find opinions easy? (FB)	1.6*	2.24*
Find details (FF)	28/37	26/37
Easy?	2.64	2.76
Find details (FB)	20/24	16/24
Easy?	2.72	2.92

of the quality of answers.

Finding facts

We compared SentenTree against the list of top tweets on fact-finding tasks. For both the movie and the conference, we asked a sanity check question and a more in-depth question.

The sanity check question for the movie asks for the name of the movie ("Fate of the Furious" or the series name "Fast and the Furious"). All 50 workers answered this question correctly. The sanity check question for the Facebook conference asks for the name of the leader of the technology company ("Mark Zuckerberg"). Three workers from the visualization condition and one from the list condition answered

"Facebook". This likely happened because they did not read the question carefully. Another worker from the list condition failed to identify the Facebook topic entirely and missed all questions related to Facebook. On average workers rated the sanity check questions between *very easy* and *somewhat easy* and the differences were not significantly different.

The more in-depth questions ask the workers to identify two people related to the movie and write a sentence explaining why they were mentioned, and to identify three technologies related to the company and write a sentence as well. These questions are open-ended and there can be many correct answers. The two graders examined at all answers and determined almost all of them to be high-quality. Then they counted the unique answers and found that workers in the visualization group gave more unique answers to both questions: for the movie people question, 11 people were identified among all 50 responses. 10 were mentioned by the SentenTree group and 8 were mentioned by the List group. For the Facebook technology question, 24 technologies (e.g., social VR, 360 camera) were identified among all 50 responses. The SentenTree group found 18 and the List group found 14 (Table 4). The differences were not statistically significant.

The workers rated finding movie-related people to be between *somewhat hard* to *neither easy nor hard* without significant difference between the conditions. However they rated finding Facebook technologies to be *significantly* easier with SentenTree than with the list of top tweets (visualization 3.16, list 2.36, $t = 2.26$, $p = 0.03$).

Understanding opinions

Workers were asked to identify one positive and one negative opinion towards the movie, and one positive and one negative opinion towards the company (Facebook). For each answer, they were asked to paste two supporting tweets. The two graders read through the responses and determined almost all of them to be high-quality (i.e., the opinions are correctly labeled as positive or negative, and the tweets support

what they claim). Because the Fate of the Furious movie was well-received, five workers using the list and three workers using the visualization were unable to find any negative opinion towards the movie. They decided against counting the unique responses for the opinion question since the responses were either vague (e.g., "It's a good movie") or repeated in the next question where users were asked to identify three interesting points.

Overall, workers found opinion-finding questions to be the hardest questions in the survey. They rated the movie opinion question to be between *very hard* and *somewhat hard* (visualization 1.68 and list 1.36). The score for the Facebook opinion question is also low at 1.6 for SentenTree and 2.24 for the list, but the SentenTree score is significantly lower ($t = -2.08$ and $p = 0.04$). This result suggests that SentenTree may not be as helpful for sentiment identification tasks.

Open-ended detail exploration

Workers were also asked to identify three interesting points about the movie, and three interesting points about the company, and to paste a supporting tweet for each point they identify. This is an open-ended question, and we were interested in whether SentenTree allows workers to get diverse answers easily.

The two graders read through all answers and put together a list of unique points for each question, and counted unique points made by workers in the two conditions. As shown in Table 4, directionally, workers using SentenTree identified more unique aspects compared to workers using the list of topic tweets for both questions.

Workers also rated the open-ended questions between *somewhat hard* to *either easy nor hard* with no significant difference between SentenTree and the list of top tweets (movie question 2.64 for the visualization and 2.76 for the list, and Facebook question 2.72 for the visualization and 2.92 for the list).

4.3.7.3 Discussion

The study evaluated SentenTree against a benchmark on four tasks: finding a high-level summary, finding facts, understanding opinions, and open-ended exploration. Workers were asked to provide evidence (tweets) to substantiate their answers to most of the questions.

Identifying high-level topics

On the task to identify major topics from the dataset, SentenTree shows a directional advantage over the list of top tweets. This advantage is likely because SentenTree presents clusters of tweets based on similar topics, while with the list users have to read each tweet one-by-one. In fact, we were surprised that two workers from the SentenTree group missed the Facebook meaning since "Facebook" is the largest word in the landing view of the visualization (Figure 40). We also noted that workers reporting this task as easy and feeling confident about their answers might have missed a significant topic completely. This suggests that a tool that summarizes major topics can be beneficial even when human analysts consider this task easy.

Identifying facts and interesting details

Finding important facts or interesting facets are common tasks performed by analysts when exploring a social media text collection. For both fact-finding and open-ended exploration, SentenTree directionally out-performed the list of top tweets on the *diversity* of answers found by users on all four questions, even though the differences were not statistically significant. Workers also rated it significantly easier to find Facebook technologies using SentenTree than the list of top tweets. Our suspicion is the visualization provides a good aggregation of the keywords related to Facebook, many of them being technologies mentioned at the conference (Figure 42), while with the list, workers have to first look up Facebook related tweets and then read them one-by-one.

Finding opinions

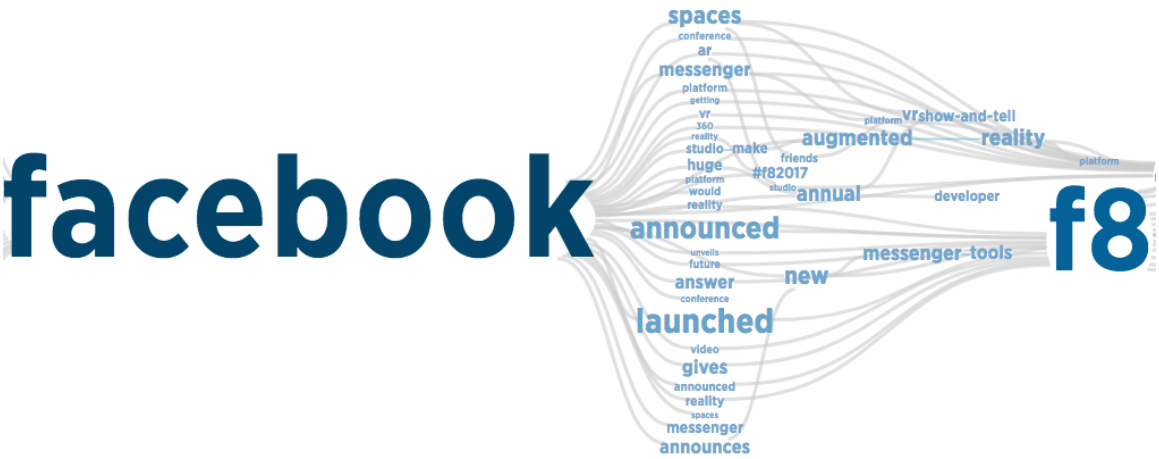


Figure 42: Partial view of the SentenTree visualization zoomed on to "Facebook". There are multiple Facebook technologies mentioned, such as *Spaces, AR, Messenger, augmented reality.*

SentenTree did not show any advantage on opinion understanding against the benchmark, and even was rated as statistically significantly harder for the Facebook question. This is probably unsurprising considering SentenTree does not have built-in sentiment analysis. The easiness ratings for this task suggests opinion understanding is one of the hardest tasks with the list of top tweets as well, so there is needs for more tool support. In our future work, we plan to introduce sentiment analysis to the system.

Finally, as demonstrated through the domain expert study, SentenTree has a higher learning curve compared to the list view. The workers from the Mechanical Turk study were unfamiliar with SentenTree and were working under a time constraint. We are confident that users would find it more efficient once they had more practice using the tool.

4.3.8 Conclusion

4.3.8.1 Contributions

We have presented SentenTree, a novel visualization technique for social media text. We designed the SentenTree technique to provide a number of the critical benefits

evident in both word clouds and the Word Tree, while also overcoming some of the limitations of each. SentenTree highlights the important (most frequent) words as do word clouds, but it fills in the connections between words to communicate sentence structure and underlying concepts, themes, and ideas more fully. A Word Tree shows the most frequent prior and following words and sentence fragments clearly, but a person must select a word to serve as the focus for the visualization. SentenTree, conversely, constructs a network of sentence fragments and automatically extracts and shows different words that effectively serve as foci within the visualization.

Additionally, the interactive capabilities of SentenTree allow a person to see specific connected sets of words just as they would appear, in order, in social media posts. A person can use interaction to effectively "drill-down" and display the individual postings from which words are taken.

Through both an interview with domain experts and a quantitative study, we have demonstrated that SentenTree can help people gain a rapid understanding of key concepts and opinions in a large text collection, and it organizes raw text into a format that is easy to locate. The SentenTree model fits well into the typical workflow of data scientists who work regularly with social media text.

SentenTree can serve as a cheap alternative to complex NLP programs that take hours to compute for tasks that do not require high accuracy. As tested in Section 4.3.3.2 SentenTree can create a reasonable visual summary of the major topics of a 100K sentence dataset in 20 seconds and supports iterative exploration of topic details. We envision it either as a stand-alone visualization tool or as a helpful component in many other text and document visualization systems. SentenTree is designed in a modular way so it may be customized or integrated into a multi-view system. Multiple people to whom we have demoed the system have commented that they would like to incorporate SentenTree as a view in their systems.

We have open-source SentenTree for the public to use [4]. The original academic paper was published at IEEE InfoVis conference [42].

4.3.8.2 Limitations

Like many research techniques, SentenTree has limitations. It is arguably less intuitive to interpret than word clouds and the Word Tree in its raw static form. Viewers may infer patterns or sequences of words that do not occur within the collection unless they use the interactive capabilities of the system. That reliance on interaction, which in general provides power to a visualization, is also a potential shortcoming.

The technique also does not pack visual elements (i.e., words) in a densely and efficiently manner into a rectangular region. Often, visualizations produced with the system exhibit quite a bit of whitespace. For tools where screen real estate is at a premium, this can be a problem.

Another limitation of the current technique is that it does not explicitly communicate any information about the temporal ordering of the social media messages it is portraying. As we demonstrated with the World Cup dataset, each SentenTree visualization is a snapshot of text between a period of time, and the viewer has to create multiple visualizations to see how content changes through time.

Finally, as discussed in previous sections, the algorithm for generating a SentenTree does not run in interactive real time. For collections around 100,000 social media messages, the system takes about 20 seconds to produce a final visualization that is ready for viewing and interaction. Ideally, SentenTree could be integrated into a surrounding system in a way that would lessen the impact of waiting. However, the time to run the algorithm means the visualization is not dynamic and cannot be changed on the fly. For example, SentenTree does not support merging or removing of words in the view because removing a frequent word would cause all patterns containing

[4] SentenTree sourcecode repository: `https://github.com/twitter/SentenTree`

that world to be re-created. In that case, the final graphs might look completely different.

CHAPTER V

CONCLUSION

5.1 Conclusion

In this thesis, I explore design principles for interactive visualizations that facilitate analysis of large quantities of text documents from social media and online communities. I focused on two domains of text: consumer reviews and social media posts. Consumer reviews are relatively similar to traditional text documents, and are usually analyzed with a clear goal. Conversely, posts from social media such as Twitter are notably different from traditional media in both topic and form, and associated analytical tasks are usually open-ended. Both domains share challenges unique to social media and online communities text, including huge scale, repeated information, and high noise-to-information ratio.

For each domain, I addressed the following research questions:

R1: What are the data characteristics and typical analytic tasks associated with text documents from social media and online communities?

R2: In order to best leverage the strength of both computational analysis and human judgment, can we identify tasks or aspects of tasks in social media analysis that are better suited to the human analyst than algorithms? What are limitations to algorithmic solutions and can they be addressed by the human analyst?

R3: How do we use interactive data visualization to address tasks or aspects of tasks identified in R2? What are efficient ways to visually communicate computational analysis results to the human viewer and for the human to interact with the views in order to gain insights about the underlying text data?

Based on insights gained through answering these research questions, I designed and implemented two interactive visualization systems, OpinionBlocks, and SentenTree. Most of the research described in this thesis has been published [40, 43, 42].

5.1.1 Contributions

Here I organize contributions of my thesis based on the three research questions.

Understanding social media data

In the thesis, I summarized some of the most important characteristics of text data from social media and online communities. The shared characteristics include their huge volume, their informal form, their high rate of repeated information and noise, and the fact that opinions expressed in different documents are often in conflict. All of these characteristics pose challenges for summarizing and analyzing the data, in addition to the difficulties of processing natural language. Of the two domains I studied, social media posts also possess other unique characteristics: the posts are much shorter than traditional text documents; therefore social media datasets tend to have a higher concentration of the most frequent words. They also have a higher density of repeated phrase patterns due to sharing and rephrasing. Social media datasets extracted based on keywords or hashtags likely contain many different topics because people can use the same keywords or hashtags in different ways.

Understanding analytic process with social media and online communities text

I also described analytic tasks associated with text data from social media and online communities. For reading consumer reviewers, I show that the most typical tasks include identifying the frequently mentioned and most controversial aspects of the product/service, inferring sentiment for each aspect, and finding supporting evidence for opinions from the original review documents.

To understand the exploratory analysis process for social media posts, I conducted

a study with the Osama Bin Laden story on Twitter. I show that in an open-ended exploratory process the analyst usually starts with confirming hypotheses, and through the analytic process comes up with more questions and answers each by looking at different aspects of the data. I highlight some pain points using existing tools and show that gaining an understanding of the textual content itself remains quite challenging, which I seek to solve with the introduction of SentenTree. The domain expert study with SentenTree highlights that organizing social media text into a format that allows analysts easy exploration and discovery will significantly improve their work process.

Leveraging the strength of both human and machine for text analysis

Text analysis systems make use of natural language processing algorithms to extract insights from text documents and save people the effort of reading through the entire dataset. From my research in both domains, I show that reading the original text document is critically important to analytic tasks for the following reasons:

- Forming opinions: original sentences can communicate more complicated concepts than aggregated words and patterns.
- Confirming opinions: analysts do not always trust computed results and need to confirm their hypothesis against the original document.
- Providing evidence: domain experts often need to provide sample documents to substantiate their claims.

Instead of trying to eliminate reading of the original text via algorithms, I propose that we should build systems that help people organize the documents and identify the most useful ones to read.

Based on my research in both domains, I have summarized three steps that take an analyst to the raw text that will potentially give them insights. These steps are

outlined in Table 5. First, we help the analyst gain an overall impression of the dataset. Second, we help them find interesting facets of the dataset that are worth exploration. And finally, we take them to the original documents so that they can make their own decision. The two visualization systems introduced in this thesis tailor these steps based on the characteristics of their domain.

Table 5: This table shows the similarity in workflow between analyzing consumer reviews using OpinionBlocks and analyzing social media text using SentenTree. Not included in the table: OpinionBlocks also allows users to correct computational mistakes at each step.

		OpinionBlocks	SentenTree
1	Form impressions	Show top aspects and sentiment	Show most frequent syntactic patterns
2	Identify interesting facets	Highlight conflicting opinions	Highlight word patterns with rich branches
3	Substantiate with original documents	Show review snippets & full review	Show raw text samples

My work with consumer reviews also shows that quality of the algorithms affects how likely people trust or enjoy using text visualization systems. Given that NLP algorithms are error-prone and it is easy for the human user to spot issues with the computed results, we can leverage human user's feedback to improve the results. I show in the OpinionBlocks user study that users are willing to provide feedback if the interaction is designed right, and improving the accuracy of the presented results enhances future users' experience with the visualization system.

Introducing novel text visualization metaphors and interactions

I designed SentenTree, a novel visual metaphor tailored for social media text documents. As discussed before, efficient visualization of text content remains lacking. Existing text visualization techniques are primarily either word-based (a variation of word cloud) or full sentence-based (as represented by Word Tree). SentenTree is a novel technique that provides the key benefits of both sides, while also overcoming

some of the limitations of each. It is uniquely suited for social media data because of the data's characteristics including a high concentration of keywords and patterns, large volume, and diverse topics. Like the famous word cloud, SentenTree can be used in conjunction with other visualization techniques. And I believe in many scenarios it can replace a word cloud and provide additional value.

I also designed novel visualization and interactions for the crowd-sourcing aspect of text correction in OpinionBlocks. We show that by making interactions very lightweight and fun, the majority of participants in our user study reported interest in helping correct text analysis errors in a visualization system for reviews.

5.1.2 Future Research Directions

Field study

For both OpinionBlocks and SentenTree I only conducted lab-based user studies and expert interviews. I believe that valuable feedback can also be gained once people have used these tools for a while. Recently, we open-sourced SentenTree, and have already heard from people expressing interest in trying it out with their own data and even integrating SentenTree into their visualization systems, so we are very optimistic about getting adoption and additional long-term feedback.

Integrating SentenTree into multi-view systems

SentenTree is intentionally built in a modular format so it can be extended or integrated into visualization systems easily. From our user studies, many participants suggested exciting ways that SentenTree can be integrated into other systems. For example, an analyst can run topic modeling on a large dataset before putting documents from each topic group into SentenTree. SentenTree can help the analyst explore each topic and drill down to the raw text easily.

Adding sentiment analysis to SentenTree

As discussed in the user study section of SentenTree, one of the big missing pieces is sentiment analysis. Since SentenTree uses a modular design, visualization systems adopting SentenTree can add a sentiment analysis component using some existing algorithm. Sentiment polarity can be encoded by the color of words, and it will not make a big difference to the layout of the visualization since we are currently double-coding size and color to denote the frequency of words.

Using user feedback to train NLP algorithms

I designed OpinionBlocks to take human feedback to improve the results of natural language processing algorithms. When enough people try to correct the same mistakes made by an algorithm, OpinionBlocks adopts the correction into the visualization, and new users no longer see the errors.

Future researchers might consider closing the loop by using the user feedback to train the algorithm. This work will create a system that can learn from users and can keep improving itself. Users will see fewer issues not only in visualizations of old datasets but also in visualizations for new datasets because the algorithm has learned from mistakes made in the past.

Over time, OpinionBlocks will also collect valuable training data that can be shared with other algorithms. Training data has always been essential but expensive for the NLP community as it usually requires manual labeling by experts. If OpinionBlocks can work as a platform to collect labeled data verified by the crowd, it can provide value for the entire NLP community.

Visualizing temporal shift of content snapshot

There are other useful dimensions for social media text analysis that I have not explored in this thesis. One critical dimension is time. Both OpinionBlocks and SentenTree treat all documents in the dataset as the same regardless of the time. But in many scenarios time is important. For example with consumer reviews, the viewer

likely wants to know if the opinion has shifted over time. Similarly, OpinionBlocks provides a snapshot into the social media text posted within a time window, but what if the analyst wants to know how social media content for the same topic changed over time? In the Related Work section, we review a large body of work focused on highlighting topical changes over time. Many of those visualizations techniques are based on a ThemeRiver-inspired stacked graph and add details on top of the stacked graph [86, 27, 93]. They often overlay a word cloud to show the content at a given time window, and another word cloud at a later time which highlights the differences. Since SentenTree is designed to replace word clouds, it will be interesting to explore if it can be used in combination with ThemeRiver-like visualizations. The challenge here is to make SentenTree pack tightly so it can show useful information even in a small space.

Visualizing Dialogues

Another aspect of social media text I have not addressed is its *conversation* aspect. Many posts are part of a conversation thread instead of stand-alone. To get insights from the data, we need to understand the context of these posts. This task becomes very challenging when we try to summarize a dataset since each post has a different context. Solving this challenge will no doubt bring a lot of value as many social media platforms and forums are designed to encourage conversations, so conversational data are in abundance.

REFERENCES

[1] "10 news stories that broke on Twitter first." http://www.techradar.com/news/internet/10-news-stories-that-broke-on-twitter-first-719532.

[2] "Associated Press reporters told off for Tweeting." http://www.bbc.co.uk/news/technology-15772243.

[3] "Automatic summarization - wikipedia." `https://en.wikipedia.org/wiki/Automatic_summarization`. Accessed: 2017-12-20.

[4] "Breaking Bin Laden." http://blog.socialflow.com/post/5246404319/breaking-bin-laden-visualizing-the-power-of-a-single.

[5] "How the Bin Laden Announcement Leaked Out." http://mediadecoder.blogs.nytimes.com/2011/05/01/how-the-osama-announcement-leaked-out/.

[6] "Lessons from the Osama bin Laden coverage." http://www.guardian.co.uk/technology/2011/may/09/lessons-from-bin-laden-coverage.

[7] "OpenNLP." http://opennlp.apache.org.

[8] "Sentiment lexicon." http://www.cs.uic.edu/ liub/FBS/opinion-lexicon-English.rar.

[9] "Stanford log-linear part-of-speech tagger." http://nlp.stanford.edu/software/tagger.shtml.

[10] "Twitter just had its CNN moment." http://www.businessinsider.com/twitter-just-had-its-cnn-moment-2011-5.

[11] "Twitter: Last night saw the highest sustained rate of Tweets ever. From 10:45 - 2:20am ET, there was an average of 3,000 Tweets per second." http://twitter.com/#!/twitterglobalpr/status/65125115272249344.

[12] "Why All the Hyperventilating About Twitter 'Breaking' Bin Laden's Death Is Total Nonsense." http://adage.com/article/mediaworks/twitter-broke-news-bin-laden-s-death-nonsense/227327/.

[13] Amershi, S., Fogarty, J., and Weld, D., "Regroup: Interactive machine learning for on-demand group creation in social networks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 21–30, ACM, 2012.

[14] Amershi, S., Lee, B., Kapoor, A., Mahajan, R., and Christian, B., "CueT: human-guided fast and accurate network alarm triage," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 157–166, ACM, 2011.

[15] Barth, L., Kobourov, S. G., and Pupyrev, S., *Experimental Comparison of Semantic Word Clouds*, pp. 247–258. Springer International Publishing, 2014.

[16] Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K., "Soylent: a word processor with a crowd inside," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 313–322, ACM, 2010.

[17] Bosch, H., Thom, D., Worner, M., Koch, S., Puttmann, E., Jackle, D., and Ertl, T., "Scatterblogs: Geo-spatial document analysis," in *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pp. 309–310, IEEE, 2011.

[18] Brody, S. and Elhadad, N., "An unsupervised aspect-sentiment model for online reviews," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 804–812, Association for Computational Linguistics, 2010.

[19] Cao, N. and Cui, W., *Introduction to text visualization*. Springer, 2016.

[20] Cao, N., Lin, Y.-R., Sun, X., Lazer, D., Liu, S., and Qu, H., "Whisper: Tracing the spatiotemporal process of information diffusion in real time," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2649–2658, 2012.

[21] Cao, N., Sun, J., Lin, Y.-R., Gotz, D., Liu, S., and Qu, H., "Facetatlas: Multifaceted visualization for rich text corpora," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1172–1181, 2010.

[22] Carenini, G. and Rizoli, L., "A multimedia interface for facilitating comparisons of opinions," in *Proceedings of the 14th international conference on Intelligent user interfaces*, pp. 325–334, ACM, 2009.

[23] Chuang, J., Manning, C. D., and Heer, J., "Without the clutter of unimportant words: Descriptive keyphrases for text visualization," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 19, no. 3, p. 19, 2012.

[24] Chuang, J., Ramage, D., Manning, C., and Heer, J., "Interpretation and Trust: Designing Model-driven Visualizations for Text Analysis," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 443–452, ACM, 2012.

[25] Collins, C., Carpendale, S., and Penn, G., "Visualization of uncertainty in lattices to support decision-making," in *Proceedings of the 2007 Joint Eurographics / IEEE VGTC Conference on Visualization (EuroVis)*, pp. 51–58, Jun 2007.

[26] Collins, C., Viegas, F. B., and Wattenberg, M., "Parallel tag clouds to explore and analyze faceted text corpora," in *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pp. 91–98, IEEE, 2009.

[27] Cui, W., Liu, S., Tan, L., Shi, C., Song, Y., Gao, Z., Qu, H., and Tong, X., "Textflow: Towards better understanding of evolving topics in text," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2412–2421, 2011.

[28] Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M. X., and Qu, H., "Context-preserving, dynamic word cloud visualization," *IEEE Computer Graphics & Applications*, vol. 30, no. 6, pp. 42–53, 2010.

[29] Culy, C. and Lyding, V., "Double tree: an advanced kwic visualization for expert users," in *Information Visualisation (IV), 2010 14th International Conference*, pp. 98–103, IEEE, 2010.

[30] Dou, W., Wang, X., Skau, D., Ribarsky, W., and Zhou, M. X., "Leadline: Interactive visual analysis of text data through event identification and exploration," in *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pp. 93–102, IEEE, 2012.

[31] Dwyer, T., Koren, Y., and Marriott, K., "IPSep-CoLa: An Incremental Procedure for Separation Constraint Layout of Graphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 821–828, Sept. 2006.

[32] Faridani, S., Bitton, E., Ryokai, K., and Goldberg, K., "Opinion space: a scalable tool for browsing online comments," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1175–1184, ACM, 2010.

[33] Felix, C., Franconeri, S., and Bertini, E., "Taking word clouds apart: An empirical investigation of the design space for keyword summaries," *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[34] Fieller, E. C., Hartley, H. O., and Pearson, E. S., "Tests for rank correlation coefficients. I," *Biometrika*, pp. 470–481, 1957.

[35] Geisser, S., *Predictive Inference: An Introduction*. CRC Press, 1993.

[36] Gensler, S., Völckner, F., Liu-Thompkins, Y., and Wiertz, C., "Managing brands in the social media environment," *Journal of Interactive Marketing*, vol. 27, no. 4, pp. 242–256, 2013.

[37] Gilbert, E. and Karahalios, K., "Understanding deja reviewers," in *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pp. 225–228, ACM, 2010.

[38] Havre, S., Hetzler, E., Whitney, P., and Nowell, L., "Themeriver: Visualizing thematic changes in large document collections," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 8, no. 1, pp. 9–20, 2002.

[39] Hearst, M. A., "Whats Up with Tag Clouds?," *Visual Business Intelligence Newsletter*, 2008.

[40] Hu, M., Liu, S., Wei, F., Wu, Y., Stasko, J., and Ma, K.-L., "Breaking news on twitter," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2751–2754, ACM, 2012.

[41] Hu, M., Liu, S., Wei, F., Wu, Y., Stasko, J., and Ma, K.-L., "Breaking news on twitter," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2751–2754, ACM, 2012.

[42] Hu, M., Wongsuphasawat, K., and Stasko, J., "Visualizing social media content with sententree," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 621–630, 2017.

[43] Hu, M., Yang, H., Zhou, M. X., Gou, L., Li, Y., and Haber, E., "OpinionBlocks: a crowd-powered, self-improving interactive visual analytic system for understanding opinion text," in *Human-Computer InteractionINTERACT 2013*, pp. 116–134, Springer, 2013.

[44] Hu, M. and Liu, B., "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, ACM, 2004.

[45] Huang, J., Etzioni, O., Zettlemoyer, L., Clark, K., and Lee, C., "Revminer: An extractive interface for navigating reviews on a smartphone," in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 3–12, ACM, 2012.

[46] Huang, J., Etzioni, O., Zettlemoyer, L., Clark, K., and Lee, C., "Revminer: An extractive interface for navigating reviews on a smartphone," in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 3–12, ACM, 2012.

[47] Java, A., Song, X., Finin, T., and Tseng, B., "Why we twitter: understanding microblogging usage and communities," in *Proc. WebKDD/SNA-KDD '07*, pp. 56–65, ACM, 2007.

[48] Jo, Y. and Oh, A. H., "Aspect and sentiment unification model for online review analysis," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 815–824, ACM, 2011.

[49] Keim, D. A. and Oelke, D., "Literature fingerprinting: A new method for visual literary analysis," in *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pp. 115–122, IEEE, 2007.

[50] Kim, K., Ko, S., Elmqvist, N., and Ebert, D. S., "Wordbridge: Using composite tag clouds in node-link diagrams for visualizing content and relations in text corpora," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pp. 1–8, IEEE, 2011.

[51] Krstajic, M., Bertini, E., and Keim, D. A., "Cloudlines: Compact display of event episodes in multiple time-series," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2432–2439, 2011.

[52] Krstaji, M., Najm-Araghi, M., Mansmann, F., and Keim, D. A., "Story Tracker: Incremental visual text analytics of news story development," *Information Visualization*, vol. 12, no. 3-4, pp. 308–323, 2013.

[53] Kucher, K. and Kerren, A., "Text visualization browser: A visual survey of text visualization techniques," *Poster Abstracts of IEEE VIS*, vol. 2014, 2014.

[54] Kwak, H., Lee, C., Park, H., and Moon, S., "What is Twitter, a social network or a news media?," in *Proc. WWW '10*, pp. 591–600, ACM, 2010.

[55] Lee, B., Riche, N. H., Karlson, A. K., and Carpendale, S., "Sparkclouds: Visualizing trends in tag clouds," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1182–1189, 2010.

[56] Lee, Y. E. and Benbasat, I., "Interaction design for mobile product recommendation agents: Supporting users' decisions in retail stores," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 17, no. 4, p. 17, 2010.

[57] Leskovec, J., Backstrom, L., and Kleinberg, J., "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 497–506, ACM, 2009.

[58] Leskovec, J., Backstrom, L., and Kleinberg, J., "Meme-tracking and the dynamics of the news cycle," in *Proc. KDD '09*, pp. 497–506, ACM, 2009.

[59] Liu, B., "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[60] Liu, B., Hu, M., and Cheng, J., "Opinion observer: analyzing and comparing opinions on the web," in *Proceedings of the 14th international conference on World Wide Web*, pp. 342–351, ACM, 2005.

[61] Liu, B. and Zhang, L., "A survey of opinion mining and sentiment analysis," in *Mining text data*, pp. 415–463, Springer, 2012.

[62] Luo, D., Yang, J., Krstajic, M., Fan, J., Ribarsky, W., and Keim, D., "EventRiver: interactive visual exploration of constantly evolving text collections," *IEEE Transactions on Visualization and Computer Graphics*, 2010.

[63] Moghaddam, S. and Ester, M., "ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 665–674, ACM, 2011.

[64] Mukherjee, A. and Liu, B., "Aspect extraction through semi-supervised modeling," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 339–348, Association for Computational Linguistics, 2012.

[65] Nelson, P., "Information and consumer behavior," *The Journal of Political Economy*, pp. 311–329, 1970.

[66] Oelke, D., Spretke, D., Stoffel, A., and Keim, D. A., "Visual readability analysis: How to make your writings easier to read," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 5, pp. 662–674, 2012.

[67] Park, D.-H., Lee, J., and Han, I., "The effect of on-line consumer reviews on consumer purchasing intention: The moderating role of involvement," *International Journal of Electronic Commerce*, vol. 11, no. 4, pp. 125–148, 2007.

[68] Patel, K., "Lowering the barrier to applying machine learning," in *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 355–358, ACM, 2010.

[69] Paulovich, F. V., Toledo, F. M. B., Telles, G. P., Minghim, R., and Nonato, L. G., "Semantic wordification of document collections," *Computer Graphics Forum*, vol. 31, no. 3, pp. 1145–1153, 2012.

[70] Riehmann, P., Gruendl, H., Potthast, M., Trenkmann, M., Stein, B., and Froehlich, B., "Wordgraph: Keyword-in-context visualization for netspeak's wildcard search," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1411–1423, 2012.

[71] Rohrdantz, C., Hao, M. C., Dayal, U., Haug, L.-E., and Keim, D. A., "Feature-based visual sentiment analysis of text document streams," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 2, p. 26, 2012.

[72] Schuler, K. K., "VerbNet: A broad-coverage, comprehensive verb lexicon," 2005.

[73] Serrano-Guerrero, J., Olivas, J. A., Romero, F. P., and Herrera-Viedma, E., "Sentiment analysis: A review and comparative analysis of web services," *Information Sciences*, vol. 311, pp. 18–38, 2015.

[74] Shen, W., Wang, J., and Han, J., "Sequential Pattern Mining," in *Frequent Pattern Mining*, pp. 261–282, Springer, 2014.

[75] Shneiderman, B., "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pp. 336–343, IEEE, 1996.

[76] Stasko, J., Grg, C., and Liu, Z., "Jigsaw: supporting investigative analysis through interactive visualization," *Information visualization*, vol. 7, no. 2, pp. 118–132, 2008.

[77] Taboada, M., "Sentiment analysis: An overview from linguistics," *Annual Review of Linguistics*, vol. 2, pp. 325–347, 2016.

[78] Thet, T. T., Na, J.-C., and Khoo, C. S., "Aspect-based sentiment analysis of movie reviews on discussion boards," *Journal of Information Science*, p. 0165551510388123, 2010.

[79] Tsagkias, M., de Rijke, M., and Weerkamp, W., "Linking online news and social media," in *Proc. WSDM '11*, pp. 565–574, ACM, 2011.

[80] Van Ham, F., Wattenberg, M., and Vigas, F. B., "Mapping text with phrase nets," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1169–1176, 2009.

[81] Viégas, F., Wattenberg, M., Hebert, J., Borggaard, G., Cichowlas, A., Feinberg, J., Orwant, J., and Wren, C., "Google+ ripples: A native visualization of information flow," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 1389–1398, ACM, 2013.

[82] Viegas, F. B., Wattenberg, M., and Feinberg, J., "Participatory visualization with Wordle," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1137–1144, 2009.

[83] Vigas, F. B. and Wattenberg, M., "Timelines tag clouds and the case for vernacular visualization," *interactions*, vol. 15, no. 4, pp. 49–52, 2008.

[84] Wang, J., Zhao, J., Guo, S., North, C., and Ramakrishnan, N., "ReCloud: Semantics-based word cloud visualization of user reviews," in *Proceedings of Graphics Interface 2014*, pp. 151–158, May 2014.

[85] Wattenberg, M. and Vigas, F. B., "The word tree, an interactive visual concordance," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 6, pp. 1221–1228, 2008.

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

[87] Wise, J. A., Thomas, J. J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., and Crow, V., "Visualizing the non-visual: spatial analysis and interaction with information from text documents," in *Information Visualization, 1995. Proceedings.*, pp. 51–58, IEEE, 1995.

[88] Wu, S., Hofman, J. M., Mason, W. A., and Watts, D. J., "Who says what to whom on twitter," in *Proc. WWW '11*, pp. 705–714, ACM, 2011.

[89] Wu, S., Hofman, J. M., Mason, W. A., and Watts, D. J., "Who says what to whom on twitter," in *Proceedings of the 20th international conference on World wide web*, pp. 705–714, ACM, 2011.

[90] Wu, Y., Liu, S., Yan, K., Liu, M., and Wu, F., "OpinionFlow: Visual analysis of opinion diffusion on social media," 2014.

[91] Wu, Y., Provan, T., Wei, F., Liu, S., and Ma, K.-L., "Semantic-preservingword clouds by seam carving," *Computer Graphics Forum*, vol. 30, no. 3, pp. 741–750, 2011.

[92] Xu, J., Tao, Y., Lin, H., Zhu, R., and Yan, Y., "Exploring controversy via sentiment divergences of aspects in reviews," in *Pacific Visualization Symposium (PacificVis), 2017 IEEE*, pp. 240–249, IEEE, 2017.

[93] Xu, P., Wu, Y., Wei, E., Peng, T.-Q., Liu, S., Zhu, J. J., and Qu, H., "Visual analysis of topic competition on social media," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 12, pp. 2012–2021, 2013.

[94] Yatani, K., Novati, M., Trusty, A., and Truong, K. N., "Review spotlight: a user interface for summarizing user-generated reviews using adjective-noun word pairs," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1541–1550, ACM, 2011.

[95] Yatani, K., Novati, M., Trusty, A., and Truong, K. N., "Review spotlight: a user interface for summarizing user-generated reviews using adjective-noun word pairs," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1541–1550, ACM, 2011.

[96] Zhao, J., Cao, N., Wen, Z., Song, Y., Lin, Y.-R., and Collins, C., "# fluxflow: Visual analysis of anomalous information spreading on social media," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1773–1782, 2014.

[97] Zhu, F. and Zhang, X., "Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics," *Journal of marketing*, vol. 74, no. 2, pp. 133–148, 2010.

Visualization of textual content from social media and online communities

Mengdie Hu

120 Pages

Directed by Dr. John T. Stasko

In this thesis, I explore design principles for interactive visualizations that facilitate analysis of large quantities of text documents from social media and online communities. I summarize characteristics of such text documents, including their huge volume, short and informal expressions, high density of repeated language patterns, high noise-to-information ratio, and the prevalence of conflicting opinions. All of these characteristics pose challenges for analyzing the data, in addition to the difficulties of processing natural language. I focus on two domains of text, consumer reviews and social media posts, and show that analytical tasks in both domains share three common steps: 1) gaining an overall impression of the dataset by learning the major topics, 2) finding interesting facets of the dataset that are worth exploration, 3) reading the original documents to gain insights. I introduce two visualization systems that address these tasks for the two domains I study. OpinionBlocks presents a novel visualization interface for reading consumer reviews and enables crowd-correction of text analysis errors. SentenTree is a new visualization technique uniquely suited for social media text analysis by providing the key benefits of both word-based (a variation of word cloud) and sentence-based (as represented by Word Tree) visual metaphors while overcoming some of the limitations of each.