Visualizing Topical Quotations Over Time to Understand News Discourse

Nathan Schneider, Rebecca Hwa, Philip Gianfortoni, Dipanjan Das,
Michael Heilman, Alan W. Black, Frederick L. Crabbe, and Noah A. Smith

CMU-LTI-01-013

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Visualizing Topical Quotations Over Time to Understand News Discourse

Nathan Schneider*
Carnegie Mellon University

Rebecca Hwa
University of Pittsburgh

Philip Gianfortoni
Dipanjan Das
Michael Heilman
Alan W. Black
Carnegie Mellon University

Frederick L. Crabbe
U.S. Naval Academy

Noah A. Smith
Carnegie Mellon University

July 2010

Abstract

We present the PICTOR browser, a visualization designed to facilitate the analysis of quotations about user-specified topics in large collections of news text. PICTOR focuses on quotations because they are a major vehicle of communication in the news genre. It extracts quotes from articles that match a user's text query, and groups these quotes into "threads" that illustrate the development of subtopics over time. It allows users to rapidly explore the space of relevant quotes by viewing their content and speakers, to examine the contexts in which quotes appear, and to tune how threads are constructed. We offer two case studies demonstrating how PICTOR can support a richer understanding of news events.

1 Introduction

Quotations are ubiquitous in journalism. They are used to support claims and perspectives identified by the journalist. Through inclusion of quotations from experts, witnesses, persons involved, or observers, the news is made more concrete and more personal. A catchy sound bite may be quoted under multiple contexts over time as the news story evolves. Quotes that relate to each other can be seen as a *thread* that runs through an event or topic as it unfolds in news discourse.

While applications such as Google News allow users to scan the major news headlines of the moment, there is less support for users to examine and explore the news collection at a deeper semantic level. For instance, a social scientist may want to understand how a hotly contested public policy was covered in the media over time. While search engines can help filter out the irrelevant articles, for a complex topic that contains many facets, there would still be a lot of text for the user to process. We conjecture that relevant quotes can serve as an economical unit of analysis to aid understanding of a topic. Our approach combines methods of natural language processing (NLP), information retrieval (IR), and text visualization.

We present a system that queries a large news corpus for topical quotations and then visualizes these quotations over time. Major features of our approach include:

- an NLP technique for identifying quotes and speakers in an article;
- a methodology for selecting quotes relevant to a user query;
- a simple, tunable metric for scoring quote similarity in order to filter and cluster related quotes into threads;
- a graph-based visualization for plotting relevant quotes over time, emphasizing links between highly similar quotes; and
- an interactive interface that permits tuning of the similarity metric and exploration of a quote's surrounding context.

Following a discussion of the motivation for this work (§2) and a system overview (§3), we describe our visualization in terms of its basic units: quotes (§4) and threads (§5). Two case studies (§6) are presented to highlight the benefits of this visualization. We believe that a tool similar to our prototype would offer substantial support to analysts, journalists, students, and social scientists by providing a novel window into the news.

*e-mail: nathan@cmu.edu

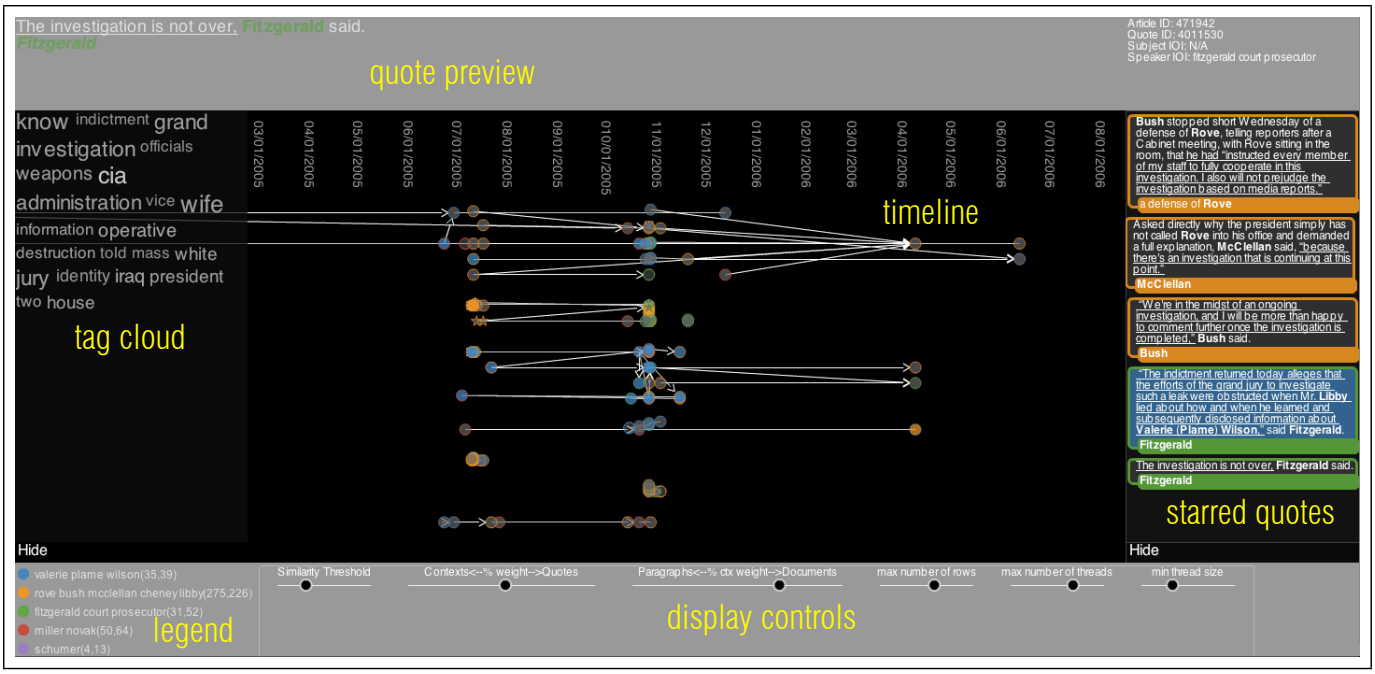


Figure 1. The PICTOR quote browser, with labels marking regions of the interface. Nodes represent quotations; they are laid out horizontally over time, and related quotes are connected to form threads.

2 Motivation

There exist a number of systems that perform text analysis over news data. Many support an interactive user interface (Hearst, 1995; Hearst and Karadi, 1997; Leuski et al., 2003; Havre et al., 2002). Typically, these systems emphasize *fast retrieval* and *at-a-glance summary*. Their target users are people who want to find a piece of information quickly or to receive an overview of hot topics.

In contrast, we want to support certain kinds of *exploratory data analysis* (Tukey, 1977)—in particular, the analysis of statements and opinions of those directly involved in news events.

As a concrete example, consider the following hypothetical essay topic in a social science class:

> In 2005, Hurricane Katrina, one of the deadliest natural disasters in the history of United States, had devastated the city of New Orleans. Both the Head of the Department of Homeland Security, Michael Chertoff, and the then-Director of the Federal Emergency Management Agency, Michael Brown, were under scrutiny for their handling of the disaster relief. Analyze the relationship between the two men over the course of the year following the disaster.

Implicitly, the essay topic requires the writer to make inferences about the relationship between the two men by analyzing what they publicly said about each other over a period of time. Researching the topic with existing NLP and IR tools is challenging for several reasons. To begin with, while it is straightforward to retrieve articles about Hurricane Katrina that mention Chertoff and Brown, the result set would be large since they were prominent figures in the news of that period, but many retrieved articles would not be relevant for the essay topic.

Moreover, automatically finding quotations embedded in news articles and attributing them to the right sources is a non-trivial NLP task. An utterance does not always reside within a pair of quote marks:

(1) Chertoff insisted he had urged Brown to get help for victims stranded in New Orleans.

and the presence of quote marks does not always indicate a complete utterance:

(2) Brown accused Chertoff of instigating a “cultural clash” between FEMA and DHS.

The source of a quotation is not always explicitly mentioned. It may not be located near the quotation, so *syntactic parsing* and *named entity recognition* may be necessary. The use of pronouns is also common, such that *anaphora resolution* is needed to determine the name of the source. Our belief about the identity of the speaker for the sentence,

"If I had known about Michael's agenda, I would have done something differently," he said.

would change depending on whether it was preceded by Sentence (1) or Sentence (2). Due to ambiguities at different levels of text processing, automatic quote extraction does not guarantee perfect results.

A further challenge is that the essay topic requires the writer to organize the quotations by formulating the semantic and temporal relationships between them. These are challenging NLP questions under active research (Das and Smith, 2009; MacCartney and Manning, 2007; de Marneffe et al., 2008; Chambers et al., 2007), but current analyzers have not reached the same level of maturity as some other NLP applications to be broadly applied to arbitrary text.

This example highlights the limitations of current NLP and IR technologies. To meet these challenges and develop an analytical tool that ties together topics, events, and named entities over time, we develop a system that supports human-computer collaboration mediated through text visualization and interactivity. Text annotation processes, including named entity tagging, parsing, co-reference resolution, and quotation extraction, are performed automatically (albeit imperfectly) over large amounts of data as a backend support. To help the user to make inferences about the semantic and temporal relationships between relevant quotes, the interface must organize the complex, multi-dimensional (and potentially incorrectly annotated) dataset in a systematic and easy-to-understand way. Because the quotes are the primary objects, and the user's goal is to establish relationships among them, we chose a graph-based representation in which the quotes are the nodes. Affiliated information, such as the speaker and the subject, are encoded by colors. Finally, we order nodes with strong semantic similarities temporally to form threads within a broader topic. Each thread can be viewed as simulating an ongoing public conversation within the fabric of news discourse.

3 System overview

The design of our system is driven by three objectives. First, we want to develop an application that encourages information exploration and discovery beyond retrieval and access. Second, we want to focus on quotations because they are interesting and complex units of text. Thanks to repetition of speakers and quoted content, quotes allow us to tie together information across documents and over time. Third, we want to use intuitive visualization and interactive user interfaces to compensate for the lossy nature of automatic text analysis.

To support these goals, we extend a standard text retrieval model (Manning et al., 2008; Strohman et al., 2005) to facilitate search for quotations attributed to (or mentioning) *relevant individuals* in news articles. To do this, we develop and implement a method to automatically harvest quotes from all articles in our database. We also extend the search interface to construct quote-centered queries. Finally, we use an interactive visualization to organize the system's presentation of retrieval results by relating quotes over time.

3.1 Components

Our system consists of the following components:

1. The **quote search engine** retrieves quotes deemed relevant to a textual query specified by the user (§4.1).
2. The **quote similarity scorer** is used to compute terms of the pairwise relatedness scores necessary for thread discovery (§5.1).
3. The **front end** uses components of the similarity scores to compute and display threads in an interactive visualization (§4.3, §5.2). There is also a simple textual interface to display the source articles for relevant quotes (Figure 2).

3.2 Implementation platform

As the size of our data is large—673,885 articles (drawn from the *New York Times* and Associated Press for 2004–2006), our system is set up as a client-server application. We elected to build our system as a Web application for platform independence (the only requirements are Firefox 3.5+ and the Adobe Flash plugin) and so we could take advantage of familiar interface elements in browsers. The server uses Ruby on Rails (Bächle and Kirchberg, 2007) to mediate between the user interfaces and the data sources. The visualization was implemented with the Prefuse Flare visualization library for Flash (Heer et al., 2005), obtained from `http://flare.prefuse.org/`.

For preprocessing text, we use the MXPOST part-of-speech tagger (Ratnaparkhi, 1996); Mark Greenwood's reimplementation of the noun phrase chunker of (Ramshaw and Marcus, 1995); and the Stanford named entity recognizer (Finkel et al., 2005). Our system does not have access to full syntactic parse information.

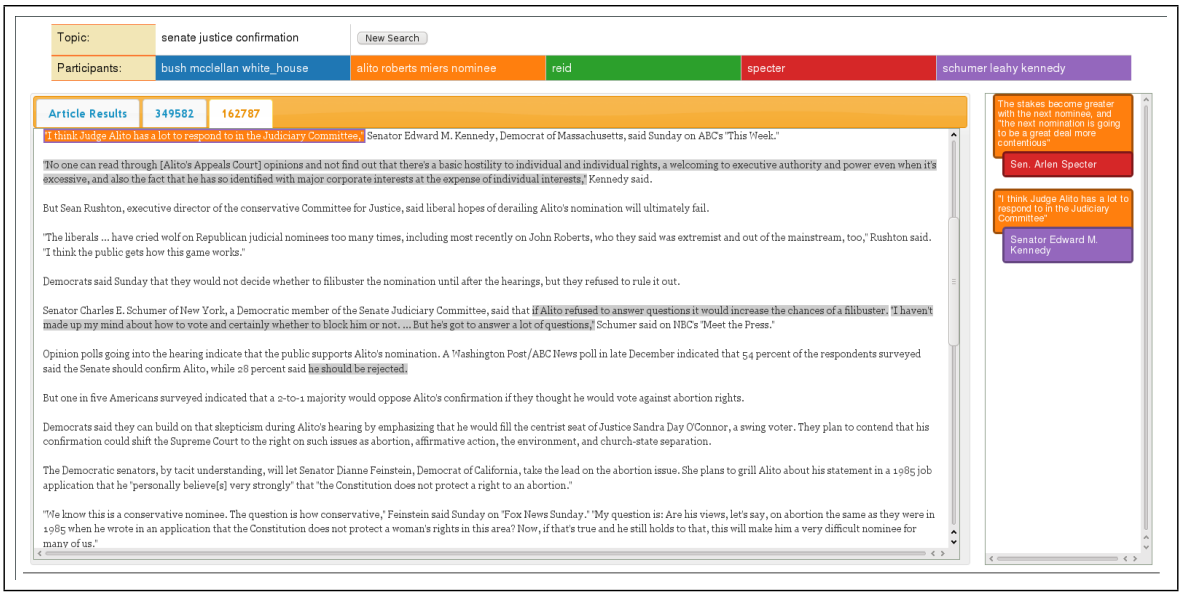


Figure 2. Textual interface for source articles. Tabs allow for easy switching among relevant articles. Relevant quotes are highlighted in place and starred quotes (also listed in the panel at right) are color-coded. Relevant quotes can be starred or unstarred in either this view or the visualization, and the respective starred quotes panels are synchronized.

4 Quotes

4.1 Extracting quotes

Automatically extracting quotes in free news text is a problem that has not been sufficiently addressed by NLP research. It resembles problems in **information extraction**, such as identifying mentions of named entities (people, places, and organizations), which has been widely studied (Grishman, 1997), especially in specific text domains like biomedical articles (Krauthammer and Nenadic, 2004). As illustrated in §2, a reporter presents quotes in many different ways such that—like most problems in natural language processing—listing all of the patterns to match is a near-impossible task.

While the development of NLP methods is not the focus of this paper, here we give a sketch of our quote extraction technique. We asked human volunteers to manually annotate the beginning and ending of quotes as well as their attributed sources (as spans of the text) in a small number of articles. Then, a computational linguist, in consultation with the data, developed a context-free grammar (CFG) with designated nonterminals for quotes and speakers, respectively. The grammar has 273 nonterminal rules. For example, the rule

$$\texttt{SpVP} \rightarrow \texttt{Per vs}$$

indicates that a speaker-plus–verb phrase pattern might consist of a phrase referring to a person (`Per`) followed by a speaking verb (`vs`), where `Per` and `vs` can further rewrite as more complex structures, eventually producing a series of word tokens; this is designed to match a phrase such as *the senior White House official said*, thereby pinpointing *the senior White House official* as the speaker. A derivation of the grammar yields one paragraph.

The linguist manually assigned numerical weights to the rules, implying a preference system; the highest-weighted parse of a paragraph, discoverable by dynamic programming, gives a derivation in which the quote and speaker nonterminals are coupled and indicate the presence of quotes. The model is called a weighted context-free grammar, and is equivalent in expressive power to well-known probabilistic context-free grammars (Smith and Johnson, 2007).

Hand-written grammars do not represent the latest in NLP technology for most structured language analysis problems, but they offer several advantages—notably speed (e.g., Microsoft Word uses a fast, hand-written grammar for grammar checking). Automatically learning a grammar would require either annotation of complete grammatical structures—something only a CFG expert would be able to do, and two experts would be unlikely to agree on the encoding—or sophisticated structure discovery. We did conduct experiments with automatic improvement of the hand-selected grammar weights using the Expectation-Maximization algorithm

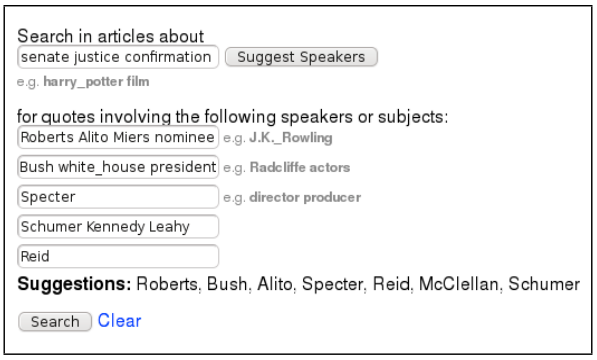


Figure 3. Search page. The topical query is specified in the first textbox: `senate justice confirmation`. The remaining are devoted to individuals. Here the first individual textbox lists a group of people as an equivalence class; the second lists multiple monikers of the same person.

(Dempster et al., 1977; Lari and Young, 1991; Pereira and Schabes, 1992), but found it harmed performance compared to the expert-selected weights. This result is consistent with some other findings in NLP (Merialdo, 1994).

Performance of the quote extraction grammar was estimated on the annotated data at 86% precision and 75% recall when measured by words correctly ascribed to a quote or speaker, and 56% precision and 52% recall when measured in terms of completely correct speaker-quote pairs.

We used this system in preprocessing to extract quote-speaker pairs from all articles in our database. Along with the article text, they are indexed for rapid retrieval at query time with the Indri search engine platform (Strohman et al., 2005).

4.2 Selecting quotes

The entry point to our system is the search page. The user is prompted for two query types: terms describing the **topic** of desired articles and terms naming **individuals** quoted or mentioned in quotes in those articles. To help users tracking multiple individuals without overwhelming them, the system allows up to five individuals to be specified. The word "individual" can be construed broadly to include any entity that might be referred to in a quote or its attribution—this includes personal names (*Barack Obama*), roles and titles (*vice president*; *legislators*), and names, roles, or metonymic references for groups and institutions (*Microsoft Corp.*; *the White House*; *France*; *football team*). We do not currently interpret pronominal references (e.g., *she* or *his wife*).

Figure 3 shows a screenshot of the search page. The system only considers articles that match all of the terms in the topic query. Of the highest-scoring of these articles, all quotes with at least one of the individual terms in the speaker attribution are returned as **relevant quotes**. This retrieval procedure is fairly restrictive in order to obtain high-precision results involving select individuals.

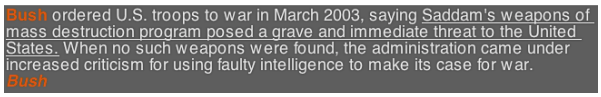
In general, a user may not be sufficiently familiar with the topic at hand to identify the important individuals. To address this scenario, he or she may click on a "Suggest Speakers" button to see the fifteen most frequent speakers for quotes in the articles matching the topic query.

In order to alter the query, the user is required to return to the search page, where he or she can issue a modified query to restart the visualization.

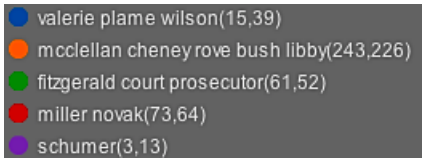
4.3 Visualizing quotes

Upon submitting a search query, the user is presented with the PICTOR visualization interface. Relevant quotes are represented in the visualization with circular nodes. These **quote nodes** are color-coded to match the individuals named in the query: the border of the node is colored according to the matched speaker, and if an individual is mentioned in the content of the quote the circle is filled with the corresponding color (otherwise, the fill color is gray).

Hovering the mouse over a quote node displays the text of its source paragraph in a **preview** pane at the top of the visualization (Figure 4a). Names of queried individuals are color-coded in the preview (with a legend at the bottom left-hand corner of the screen; Figure 4b); the quote itself is underlined, and the speaker attribution is repeated below the quote text. Clicking on the preview text opens the entire **source article** in another browser



(a)



(b)

Figure 4. Above: preview of a paragraph containing an indirect quote. The quote itself is underlined; the attributed speaker is repeated below the paragraph and color-coded according to the legend of individuals for the search, below. See §6.2 for the search query. Counts in the legend correspond to the number of relevant quotes for which one of the individuals in the group is mentioned in the quote or its attribution, respectively.

tab (Figure 2), which enables the user to see the full context of the quote (this is particuarly important given the uncertainty of the quote extraction process and the ambiguity inherent in natural language).

Clicking on a quote node of interest will star the quote, replacing the circle with a star icon and displaying the quote text in a panel to the right of the visualization. This ability to mark particular quotes is useful because it allows users to compare quotes side-by-side as well as enabling easy interaction with the source article.

A **tag cloud** (Viégas and Wattenberg, 2008) provides a lexical summary of the displayed quotes: the most frequent words (omitting stopwords) from these quotes are displayed, and their relative sizes are in proportion to frequency. An example is shown in Figure 5. Hovering over a keyword in the cloud causes the quote nodes using that keyword to flicker.

The user may adjust the time range of quotes in the current view by panning (clicking and dragging the canvas) and zooming (with the mouse wheel). As nodes leave from or enter into view, the tag cloud updates to reflect the shift, thereby serving as a qualitative indicator of the semantics localized in a particular part of the space.

5 Threads

5.1 Discovering threads

Quote threads are clusters of related quotes. Specifically, a **similarity score** is computed between all pairs of relevant quotes, and nodes representing the quotes are linked if their similarity score surpasses a (tunable) threshold value τ. The connected components of the resulting graph are taken as threads.

Our prototype uses a variant of **cosine similarity** for this purpose. A standard and simple technique for computing relatedness of two pieces of text, cosine similarity (Manning et al., 2008) is the inner product of two vectors of word counts:

$$c(x, x') = \frac{\mathbf{x}^\top \mathbf{x}'}{\|\mathbf{x}\|\|\mathbf{x}'\|}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are the respective word count vectors for strings x and x'. In other words, the larger the proportion of words common to the two pieces of text, the higher the score.

However, in estimating quote-to-quote similarity, we can easily leverage information about quotes beyond their words—in particular, we consider the words appearing in context (i.e. in the same paragraph or document). We adopt the following function as a simple measure of pairwise quote similarity:

$$quote\text{-}sim(q, q') = \alpha \cdot c(q_c, q'_c) + \beta \cdot c(q_p, q'_p) + (1 - \alpha - \beta) \cdot c(q_d, q'_d)$$

where a quote q is represented with three strings: q_c for its content (not including the speaker attribution), q_p for its entire containing paragraph, and q_d for its entire containing document. α and β are tunable parameters[1]

[1] α is constrained to fall between 0 and 1; β is constrained to fall between 0 and $(1 - \alpha)$.

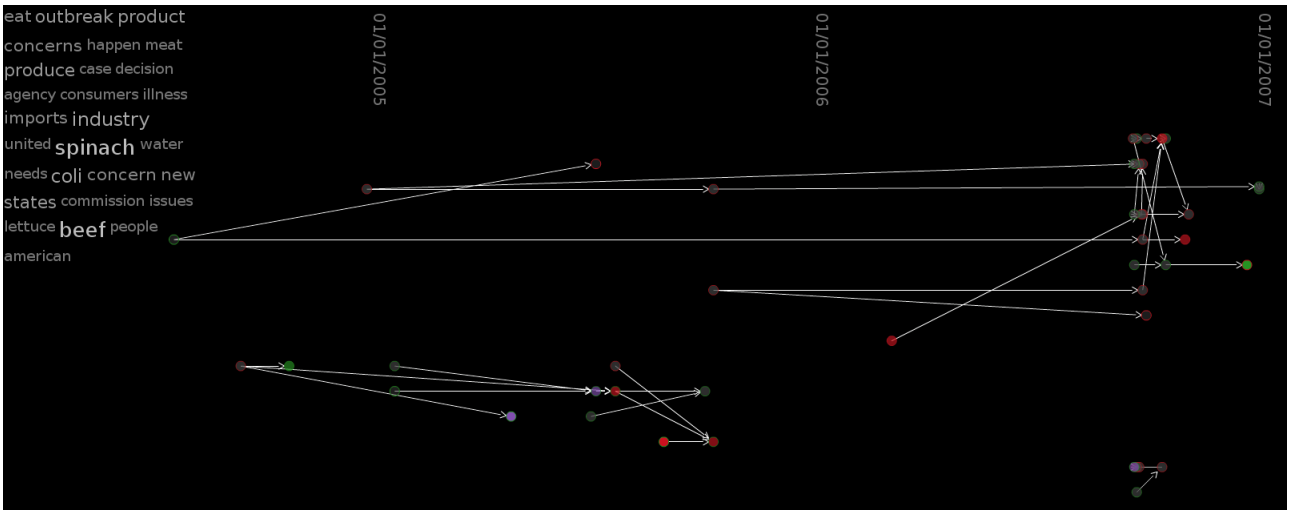


Figure 5. Three threads of quotes related to food safety. Arrows within each thread indicate progression over time. The tag cloud at left summarizes the frequent words within all quotes currently in view.

that trade off the relative impacts of these three types of information: $\alpha = 1, \beta = 0$ uses only local information, whereas $\alpha = 0$ uses almost exclusively nonlocal information.

The most useful setting of the three parameters governing thread formation (α, β, and τ) likely depends on the user's query and goals. Rather than fix their values *a priori*, we leave this up to the user: slider controls below the visualization allow the user to customize the threading behavior interactively via these parameters, enabling the user to trade off between thread *size* and *coherence*.

We found that in practice, most threads consist of a very small number of quotes, and are thus not very interesting. To focus the user's attention on interesting threads, our system only displays threads containing a minimum number of quotes. This minimum value is tunable in the interface; it is 3 by default.

Finally, it is worth noting that there may be better methods of clustering quotes into threads. In particular, similarity metrics based solely on words counts are shallow; determining a more sophisticated distance function which yields better threads is a natural language processing problem. Indeed, an active area of NLP research is the recognition of semantic relationships that may or may not hold between a pair of sentences, such as paraphrase (Das and Smith, 2009), entailment (MacCartney and Manning, 2007), and contradiction (de Marneffe et al., 2008). Threads might also benefit from automatic extraction of narrative event schemas (Chambers and Jurafsky, 2009) or data reflecting human similarity judgments (Klippel and Weaver, 2008). In our view, text visualization technologies could be not only a useful application of existing semantic relationship detection algorithms, but could also provide insight into the types of relationships that ought to be studied in NLP. Moreover, our interactive framework would allow for rapid experimentation with different similarity metrics across many queries.

5.2 Visualizing threads

Next we consider the problem of visualizing the discovered threads. There are several desiderata for the visualization: the quotes in each thread should be visualized as interactive nodes (§4.3); the temporal structure of the displayed quote collection should be apparent; and highly similar pairs of quotes within each thread should be linked in a manner that potentially illustrates substructure within the thread, but without cluttering the space with too many links. Our solution is to display each thread as a polytree,[2] a form which is somewhat more expressive than tree-structured threads in a typical email or newsgroup conversation ((Kerr, 2003; Zhu and Chen, 2008)) because nodes may have multiple parents. Figure 5 shows three such threads.

Our thread discovery algorithm is as follows. First, edges are chosen from the undirected similarity cluster using Prim's greedy algorithm (Prim, 1957) to find a maximum spanning tree, with edges scored by the similarity of the two quotes they link. Then, the edges of the tree are directed according to the relative temporal ordering of the adjacent quotes, forming a polytree. The polytree is laid out horizontally by time, with date markers labeling the horizontal axis at regular intervals. To conserve vertical space, branches of the tree are merged (i.e.

[2] A polytree is a connected directed acyclic graph with no undirected cycles.

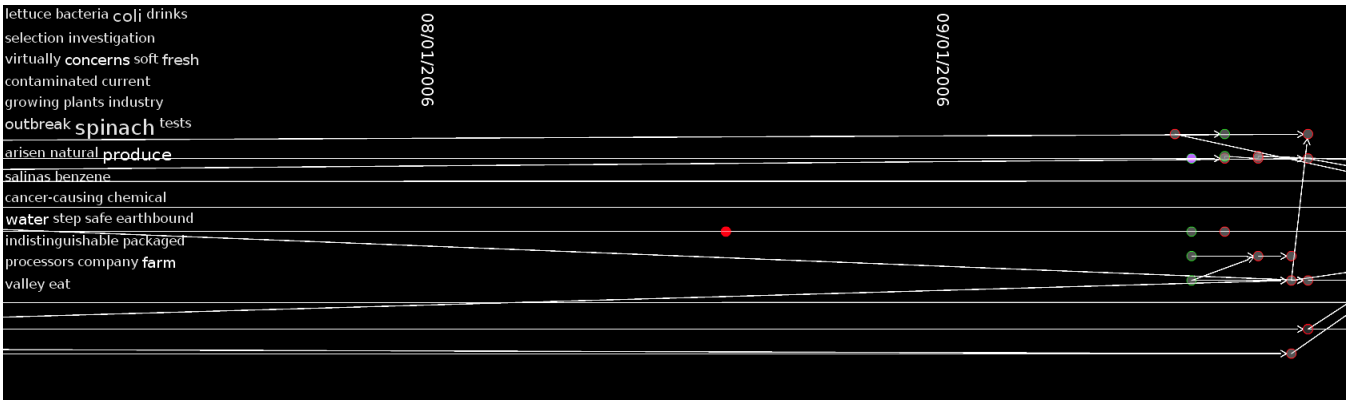


Figure 6. Zooming in to focus on part of a thread. Note the spike in activity in September 2006.

some similarity links are swapped for lower-scoring ones) until the nodes of all visible threads can be laid out in a given number of rows. Controls allow the user to adjust the maximum number of threads to display, as well as the limit on the total number of rows occupied by the visible threads.

6 Case studies and feedback

6.1 Case study 1: Food safety

Suppose that a user of our system wants to learn about food safety threats in the United States between 2004 and 2007: the outbreaks that occurred, including their evolution over time and any political implications. The user performs the following steps:

(1) Search: The user navigates to the search page (as depicted in Figure 3) and enters the following query terms: *topic:* `food safety`; *speaker/subject terms:* `regulators`, `bush`, `officials`, `food`, and `safety`. This will launch a search for quotations in documents relevant to the topic of food safety whose speaker attribution contains at least one of the specified speaker/subject terms.

(2) Visualization tuning: Upon initiating the search, the PICTOR visualization interface will open in a new browser tab. The user proceeds to adjust the thread settings (§5.2). The default similarity threshold (.500) is too high in this case, so the visualization canvas will initially be empty. Using the "Similarity Threshold" slider at the bottom of the screen, the user starts to reduce this threshold value; as the slider is dragged, a dynamic query causes threads to appear and grow in size, as more of the relevant quotes are deemed sufficiently similar to join the quotes in the thread. With about 40 quote nodes on the screen, the user decides to fix the new similarity threshold (at .325).

At this point the 40 visible quotes are grouped into just two threads. The user experiments by adjusting the other similarity parameter settings which control the degree to which context affects the similarity scores between quotes. By moving the "Paragraphs vs. Documents" slider to the right (changing the value of β to .215 from its default setting of .25, while leaving the default setting $\alpha = .5$), the user biases the similarity function so that words in the same paragraph are given less strong of a preference over words elsewhere in the document. This has the effect of splitting the visualization into three threads, as shown in Figure 5.

(3) Exploration: Next, the user explores the quotes in each thread to obtain an overview of the topic of food safety as reflected by news discourse, and to determine what aspects of the topic are salient at different points in time. The user focuses first on the bottom-left thread, dragging the canvas to move most of the quotes from other threads out of the current view. This causes the tag cloud to update so as to emphasize frequent keywords in this thread. These keywords (including **beef**, **standards**, **imports**, **tokyo**, **american**, and **united**) suggest concern over the standards of beef products that is affecting trade between the U.S. and Japan. Examining the quotes in this cluster reveals that they pertain to Japanese concerns over U.S. beef standards, and Japan's halting of imports of U.S. beef as a precaution against mad cow disease.

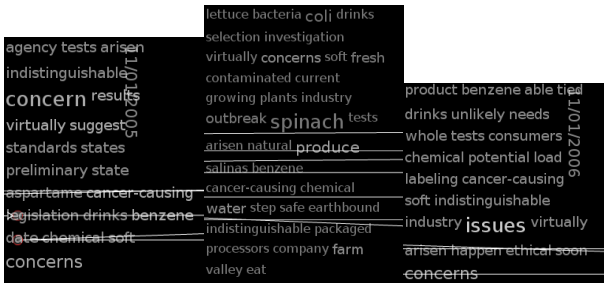


Figure 7. Tag clouds when focusing on a thread's quotes before, during, and after a spike in activity.

The tag cloud can also be used to ascertain how threads change over time. The top thread spans nearly two years, and exhibits a large spike in quotes during September 2006. If the user drags the timeline horizontally to focus on the spike, as shown in Figure 6, the top keywords become **spinach**, **concerns**, **outbreak**, **coli**, **fresh**, **water**, and **produce**. The user can compare these with the tag clouds when focusing to the left and right of the spike, as Figure 7 illustrates: **concern(s)** and **benzene** are prominent prior to the spike, whereas **issues** and **concerns** are prominent after the spike. Interestingly, **concerns** is common across all three portions of the timespan, whereas **spinach** is prominent only in the spike.

The third thread, though not as interesting or large as the first two, still contains related and relevant quotes. This thread in particular contains quotes pertaining to contaminated water.

This example demonstrates that the quote visualization can provide an effective at-a-glance summary of a topic as its appearance in the news changes over time. In particular, the clustering of quotes into threads separates different aspects of a topic, and the density of quotes can serve as a rough measure of a thread's prominence at a given time. Our interactive interface allows the user to zero in on salient parts of the graph, using the tag cloud, quote previews, and date markers to quickly obtain a rough understanding of these parts. Then, the user can click through to the source articles for additional details about the discovered quotes.

6.2 Case study 2: Plamegate

Our visualization is especially useful when viewing a large number of quotes from many different articles. The quotes serve as bite-sized units of content, and the threads can reveal important subtopics by clustering related quotes. We illustrate this by exploring the topic of "Plamegate," the scandal in which the Bush administration was accused of leaking to the press, for political reasons, Valerie Plame's status as a covert CIA operative. This scandal—and the ensuing criminal proceedings—occupied the news pages for over a year and involved several parties: Plame and her husband, Joe Wilson; journalists involved in the initial leak; Bush administration officials and their Republican allies; Democratic critics; and the special prosecutor investigating the case.

Understanding the many dimensions of the story—including the political, legal, and national security ramifications of the leak—and how they unfolded over time would probably require reading dozens of articles about the story. Summaries of the topic (e.g. encyclopedia articles), if available, might be lacking in breadth or depth, and are less likely to contain evidence from primary sources. The visualization is designed to help manage this topical and temporal complexity. While quotes alone may not always provide sufficient context and detail, it is hoped that a visualization organizing quotes into threads will simplify the search for a set of relevant articles satisfying the user's need for depth as well as topical/temporal coverage.

Searching with the query

topic: `plame scandal`
speaker/subject term groups:

- `valerie plame wilson`,
- `mcclellan cheney rove bush libby`,
- `fitzgerald court prosecutor`,
- `miller novak`,
- `schumer`

the user tunes the parameters so as to yield 7 threads containing a total of 108 quotes from 43 different articles,

as shown in Figure 1.[3] The fifth and sixth threads are not particularly interesting: they apparently correspond to one phrase that occurs in several articles published on the same day. Other threads, however do contain interesting and related quotes. The first is an amorphous thread about the leak itself and the administration's response to the scandal. Quotes in the second thread pertain to the criminal investigation, and quotes in the third thread discuss Valerie Plame and her relation to the CIA. The last thread is clearly about weapons of mass destruction.

Branches in the thread structure can be meaningful as well, in that they contain many of the same words and phrases. For example, the first thread has a branch wherein quotes from multiple articles over time used the phrase "two senior administration officials":

(3) Columnist Robert Novak has said only that "two senior administration officials" were his sources. *3/4/2004*

(4) The case has its roots in an article published two years ago by the syndicated columnist Robert Novak. He reported that "two senior administration officials" had told him that Plame was "an agency operative on weapons of mass destruction." *5/4/2005*

(5) In 2003, Novak exposed Plame's identity eight days after her husband, former U.S. Ambassador Joseph Wilson, accused the Bush administration of manipulating prewar intelligence to exaggerate the Iraqi threat. In the column disclosing Plame's CIA status, Novak said the sources for his column were two administration officials. *11/3/2005*

Quotes that appear multiple times over several months may indicate important sound bites and point to highly relevant articles.

The PICTOR visualization can thus serve as an entry point to a user trying to understand a complex and dynamic topic. Upon identifying some of the different aspects of and perspectives on that topic, the user can drill down to appropriate source articles for fuller accounts of these concerns.

6.3 Qualitative feedback

To obtain a preliminary assessment of the system for guidance in development, we demonstrated versions of our prototype to several users (computer science undergraduates) and asked them to perform an exploratory data analysis task similar to the task presented in §2—once using the full system, and once using a text-only baseline search interface. Feedback from users was very positive, with nearly all users preferring the full system:

> I wish I could have used [the full system—rather than the baseline—] all of the time.
>
> it would be great to have full functionality. I liked using this system more than google for a scholastic task.

Users differed on their perceptions of the difficulty of using the interface, and some felt more experience using the system would have been helpful. Some sample responses:

> The interfaces, although requiring some practice, were fairly easy to use after a little trial and error.
>
> It was difficult to get used to using the interface.
>
> The interface was very easy to use. The scroll and zoom controls were a little awkward, but it was not a major problem.

Almost all users found the quote extraction accuracy sufficient for the task.

> There were some errors, but they were not very significant and generally still had relevant information.
>
> Overall, they were fairly accurate. However, quotes were often skipped entirely, and were sometimes misattributed, especially when written about in passive voice. Also, sometimes the entire quote was not included.

A more rigorous, quantitative evaluation of the system is left for future work.

[3] In this case, the similarity threshold was set to .478, and the similarity parameters were set to $\alpha = .520, \beta = .144$

7 Related work

Our goal of thread discovery is similar to that of MemeTracker (Leskovec et al., 2009), which also extracts trends from quotations and visualizes them over time. A major difference is that MemeTracker discovers verbatim quote snippets *repeated* across many different sources (blog posts), whereas we are interested in bringing together related quotes from newswire/newspaper sources that may have substantially different text. In a similar vein, Rose et al. (2009) identify "themes" from news streams using a keyword extraction and clustering procedure, and display several of these over time in a static visualization. The approach taken here differs from both the MemeTracker and Rose et al. methods in that it allows the user to specify a focused query, discovers and clusters pertinent indirect (as well as direct) quotes, identifies the attributed speaker for each quote, and provides users with the option to see each quote in context in its source article. It should be noted that, due to the size of our corpus, our methods do not scale to the unfiltered scenario where no topic query is specified; therefore, our system can be thought of as complementing the MemeTracker and Rose et al. methods: these could be used to obtain a macro-level view of the corpus, and the exploration could then be refined with a search in our system.

There has also been previous work using visualization and interactive tools to assist researchers in evaluating and optimizing NLP algorithms. One example of such a system is iNeATS (Leuski et al., 2003), which extends the NeATS summarization tool (Lin and Hovy, 2002) by presenting a collection of document summaries in an interface, giving the user control over parameters for the summarization process and visualizing locations found in the data on a map. Like the quotes in our system, the summaries in iNeATS are linked back to their source documents. Similarly, visualization has been used to facilitate analysis of text processing techniques including document classification and clustering (Eick et al., 2006), opinion mining (Oelke et al., 2009, 2008), and latent semantic analysis (Crossno et al., 2009).

Text visualization for the humanities and social sciences has been explored previously (Zhu and Chen, 2008; Tat and Carpendale, 2002; Rohrer et al., 1998; Fekete and Dufournaud, 2000; Viégas et al., 2004; Plaisant et al., 2006; Don et al., 2007; Vuillemot et al., 2009; Collins et al., 2009). Most of this work is concentrated at the level of words or short phrases on the one hand, or documents on the other—whereas here we consider intermediate units of structure (quotes). Our use of an initial narrowing query is another difference with much of the work in this area.

8 Future work

While we believe our prototype as described above would be helpful to a user given adequate training in the system, before our system is made available to a general audience we envision a few usability polishes—e.g., simplifying the search interface and better integrating it with the visualization, intelligently choosing default parameters based on the data, and improving the thread layout algorithm. Another area of future work is to identify more clearly the types of situations in which our system is likely to outperform other text analytic tools (e.g., full text search), and to run further experiments with users to verify our system's potential.

As suggested above (§5.1), there is also a significant opportunity to further develop and apply natural language processing techniques in support of our text visualization application. Two of the most promising lines of investigation in NLP motivated by our application are quote-aware coreference resolution for speakers and subjects to improve quote recall, and more sophisticated similarity metrics for clustering quotes into threads.

9 Conclusion

We have described a visualization for text analytics on large news corpora. Our system automatically extracts quotations from articles that are relevant to a user-specified query. It clusters these quotations and displays them in threads that illustrate their topical and temporal relationships. Its interactive interface allows for tuning of the parameters that govern the clustering and display process. The interface also links quotes back to their source articles in order to make additional context available. Finally, we presented two case studies that demonstrate the potential utility of visualizing quotes for gaining a high-level understanding of the participants and issues reflected in the news over time.

References

Michael Bächle and Paul Kirchberg. Ruby on Rails. *IEEE Software*, 24(6):105–108, 2007.

Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. In *Proceedings of ACL-IJCNLP*, pages 602–610, 2009.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying temporal relations between events. In *Proceedings of ACL*, pages 173–176, 2007.

Christopher Collins, Fernanda B. Viégas, and Martin Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, 2009.

Patricia J. Crossno, Daniel M. Dunlavy, and Timothy M. Shead. LSAView: a tool for visual exploration of latent semantic modeling. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, 2009.

Dipanjan Das and Noah A. Smith. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of ACL-IJCNLP*, pages 468–476, 2009.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. Finding contradictions in text. In *Proceedings of ACL-HLT*, pages 1039–1047, 2008.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

Anthony Don, Elena Zheleva, Machon Gregory, Sureyya Tarkan, Loretta Auvil, Tanya Clement, Ben Shneiderman, and Catherine Plaisant. Discovering interesting usage patterns in text collections: integrating text mining with visualization. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, pages 213–222. ACM, 2007.

Stephen G. Eick, Justin Mauger, and Alan Ratner. Visualizing the performance of computational linguistics algorithms. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 151–157, 2006.

Jean-Daniel Fekete and Nicole Dufournaud. Compus: visualization and analysis of structured documents for understanding social life in the 16th century. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 47–55. ACM, 2000.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370, 2005.

Ralph Grishman. Information extraction: techniques and challenges. In *International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, pages 10–27. Springer-Verlag, 1997.

Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. ThemeRiver: visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

Marti A. Hearst. TileBars: visualization of term distribution information in full text information access. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 59–66. ACM Press/Addison-Wesley Publishing Co., 1995.

Marti A. Hearst and Chandu Karadi. Cat-a-Cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 246–255. ACM, 1997.

Jeffrey Heer, Stuart K. Card, and James A. Landay. Prefuse: a toolkit for interactive information visualization. In *Proceedings of SIGCHI '05*, pages 421–430. ACM, 2005.

Bernard Kerr. THREAD ARCS: an email thread visualization. In *Proceedings of the IEEE Symposium on Information Visualization*, 2003.

Alexander Klippel and Chris Weaver. Analyzing behavioral similarity measures in linguistic and non-linguistic conceptualization of spatial information and the question of individual differences. In *GIScience Workshop on Information Semantics and Its Implications for Geographical Analysis (ISGA)*, 2008.

Michael Krauthammer and Goran Nenadic. Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37(6):512–526, December 2004.

K. Lari and S. J. Young. Applications of stochastic context-free grammars using the Inside-Outside algorithm. *Computer Speech & Language*, 5(3):237–257, July 1991.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 497–506, 2009.

Anton Leuski, Chin-Yew Lin, Liang Zhous, Ulrich Germann, Franz Josef Och, and Eduard Hovy. iNeATS: interactive multi-document summarization. In *Proceedings of ACL: Interactive Posters and Demonstrations Session*, 2003.

Chin-Yew Lin and Eduard Hovy. From single to multi-document summarization. In *Proceedings of ACL*, 2002.

Bill MacCartney and Christopher D. Manning. Natural Logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, 2007.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

Bernard Merialdo. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2):155–171, 1994.

Daniela Oelke, Peter Bak, Daniel A. Keim, Mark Last, and Guy Danon. Visual evaluation of text features for document summarization and analysis. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 75–82, 2008.

Daniela Oelke, Ming Hao, Christian Rohrdantz, Daniel A. Keim, Umeshwar Dayal, Lars-Erik Haug, and Halldór Janetzko. Visual opinion analysis of customer feedback data. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, 2009.

Fernando Pereira and Yves Schabes. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, 1992.

Catherine Plaisant, James Rose, Bei Yu, Loretta Auvil, Matthew G. Kirschenbaum, Martha Nell Smith, Tanya Clement, and Greg Lord. Exploring erotics in Emily Dickinson's correspondence with text mining and visual interfaces. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 141–150, 2006.

Robert C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36(6): 1389–1401, 1957.

Lance A. Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, pages 82–94, 1995.

Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 1, pages 133–142, 1996.

Randall M. Rohrer, David S. Ebert, and John L. Sibert. The shape of Shakespeare: visualizing text using implicit surfaces. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 121–129, 1998.

Stuart Rose, Scott Butner, Wendy Cowley, Michelle Gregory, and Julia Walker. Describing story evolution from dynamic information streams. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 99–106, 2009.

Noah A. Smith and Mark Johnson. Weighted and probabilistic context-free grammars are equally expressive. *Computational Linguistics*, 33(4):477–491, 2007.

Trevor Strohman, Donald Metzler, Howard Turtle, and W. Bruce Croft. Indri: a language-model based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, 2005.

A. Tat and M. S. T. Carpendale. Visualising human dialog. In *Proceedings of the Sixth International Conference on Information Visualisation (IV'02)*, pages 16–21, 2002.

John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1977.

Fernanda B. Viégas and Martin Wattenberg. Tag clouds and the case for vernacular visualization. *interactions*, 15(4):49–52, 2008.

Fernanda B. Viégas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 575–582, 2004.

Romain Vuillemot, Tanya Clement, Catherine Plaisant, and Amit Kumar. What's being said near "Martha"? Exploring name entities in literary text collections. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, 2009.

Bin Zhu and Hsinchun Chen. Communication-Garden system: visualizing a computer-mediated communication process. *Decision Support Systems*, 45(4):778–794, 2008.