StatVis: Visualization of Statistical Analysis of News Data

Nicholas Kong
University of California, Berkeley
nkong@eecs.berkeley.edu

ABSTRACT
We describe a visualization of statistical analysis of news data, specifically the headlines of the New York Times. We detail its implementation and its integration with our existing collaborative visualization framework. We then discuss the advantages and disadvantages of the visualization and outline a number of further directions we wish to take this project.

Author Keywords
Visualization, collaborative work, large text corpora.

ACM Classification Keywords
H.5.2. Information Interfaces: User Interfaces.

INTRODUCTION
Text is the most common form of visualization, and one of the most idiosyncratic. Since the advent of the Internet our access to text, and concomitantly our difficulty to assimilate it all, has grown exponentially. Clearly, some form of summarization is needed.

However, as Hearst points out, text is difficult due to the inherent ambiguity of language [7]. Text is highly contextual and homographs only exacerbate the problem. But because text is also highly redundant, we can still create meaningful summaries through intelligent pre-processing of the text. In this paper we describe a visualization of a novel statistical analysis of a text corpora, with the specific challenge being revealing the message of these statistics. We first review related work in the area of text visualization.

RELATED WORK
There has been a substantial amount of work in the visualization of single documents. TextArc uses animation and a visual representation similar to tag clouds to show a story's progression [16]. Tag clouds (such as Wordle [18]) are prevalent on the web; they encode the frequency of a word by its size. Recently, Many Eyes [11] has incorporated an interactive document visualization called "Word Trees", in which the user can search for words or phrases and see a distribution of the frequency of phrases that follow. This helps to somewhat reveal the structure of a document.

More pertinent to this proposal is the work in visualization collections of documents. Rennison did early work in this area with his Galaxy of news work [14]. His goal was to allow users to adroitly explore and assimilate large amounts of news information through grouping of related articles. He used a news feed with an initial hierarchical structure, then analyzed co-occurrences to create a hierarchy of keywords to headlines to articles. The exploration interface was similar to the concurrently presented Pad++'s Document Explorer [3]. Both used a semantic zooming metaphor; in Rennison's work, the user could zoom in on "Media", which would then reveal keywords (such as "Advertising"), which on subsequent zooming would reveal relevant headlines, then articles. In some respects, this grouping of news data is the grandfather of services such as Google News and online news aggregator visualizations such as Newsmap [12].

Albrecht-Bueler et al. take a different approach. They use a live RSS feed to continuously update a visualization of "hot" topics [1]. Their analysis is solely based on co-occurrence of words, and the visualization is a graph. Each word is a node, and the length of each edge is inversely proportional to the number of co-occurrences between two words. The result is that terms that are often used together end up grouped together, so the viewer can extract the talked about topics and possibly sentiment.

Wise et al. outline two main approaches to document visualization [17]. Both approaches hinge on the representation of a document as a multidimensional vector. For example, one possible way to vectorize a document is simply to give a vector of the number of occurrences of each word in the document. In their Galaxies visualization, they project the multidimensional document vectors into 2D space, where documents cluster together based on a similarity measure. In their ThemeScape visualization, they create a three-dimensional "landscape" in which "themes", recovered from their document analysis, are plotted as a function of height. So, popular themes end up as the tallest peaks, and related themes cluster next to each other.

CONTRIBUTIONS
What we contribute is a visualization of a new sort of data analysis, where we visualize words that "predict" the existence of a keyword, in our case "Iraq". The main challenge we face is how to design a visualization that

brings meaning and comprehensibility to the underlying statistics.

APPROACH AND IMPLEMENTATION

Our objective was two-fold: to take a first step in interactive visualization of patterns in time for large news corpora, and to integrate this visualization into our current collaborative visual analysis tool. We are working closely with Professor Laurent El Ghaoui and his StatNews [15] project at UC Berkeley to recover trends in news data. Their statistical analyses provide the data that we seek to visualize. This problem is not trivial as an "intuitive" meaning of, say, a naïve Bayesian classifier coefficient does not necessarily exist. This paper describes a visualization of an early StatNews analysis of New York Times headlines spanning from January 1981 to December 2007. More sophisticated and larger-ranging analyses are planned as the project matures. We also plan significant extensions to the current visualization as well as new visualizations to accommodate other analyses; these are further detailed in the Future Directions section.

StatNews Naîve Bayesian Analysis

A brief discussion of the analysis of the underlying data is in order. As mentioned previously, the corpus in question contains the New York Times headlines spanning from January 1981 to December 2007. The statistics are computed in a rolling horizon fashion; that is, each word is analyzed in the context of the preceding year of headlines. A word will have a large positive coefficient if it increases the likelihood of seeing, in our case, "Iraq" in the previous year of headlines. A word will have a negative coefficient if it decreases the likelihood of seeing "Iraq" in the previous year. What we get, then, is a coefficient for each word for each month in the time span of the corpus. To visualize this we choose a matrix-like representation.

Visualization Description

Figure 1 shows the initial view of the visualization and the surrounding system. There are two main panels: the visualization panel at left, and the comment panel at right. The visualization is broadly a display of how likely a word is to predict the word "Iraq" in the preceding year of the New York Times headlines. Each row represents a word and each column represents a month. The color encodes the statistical significance of each word; the darker the color, the more likely it predicts "Iraq".

The user can interact with the visualization using the search box and date range selection slider at the top of the visualization. Their placement next to the title is due to the size of the window; implementation idiosyncrasies are detailed in the next section.

When the user types in the search box, the visualization automatically updates (optionally using animation) to reflect the search query. We have implemented rudimentary compound query capability, as shown in Figure 2. The logical AND is implemented by a space or "+", while the logical OR is implemented by "|". The user wished to identify the significance of words starting with both "kurd" and "kuwait", and used the query string "kurd|kuwait". When hovering over a cell, a tooltip is shown displaying the word and the statistical significance at that particular date.

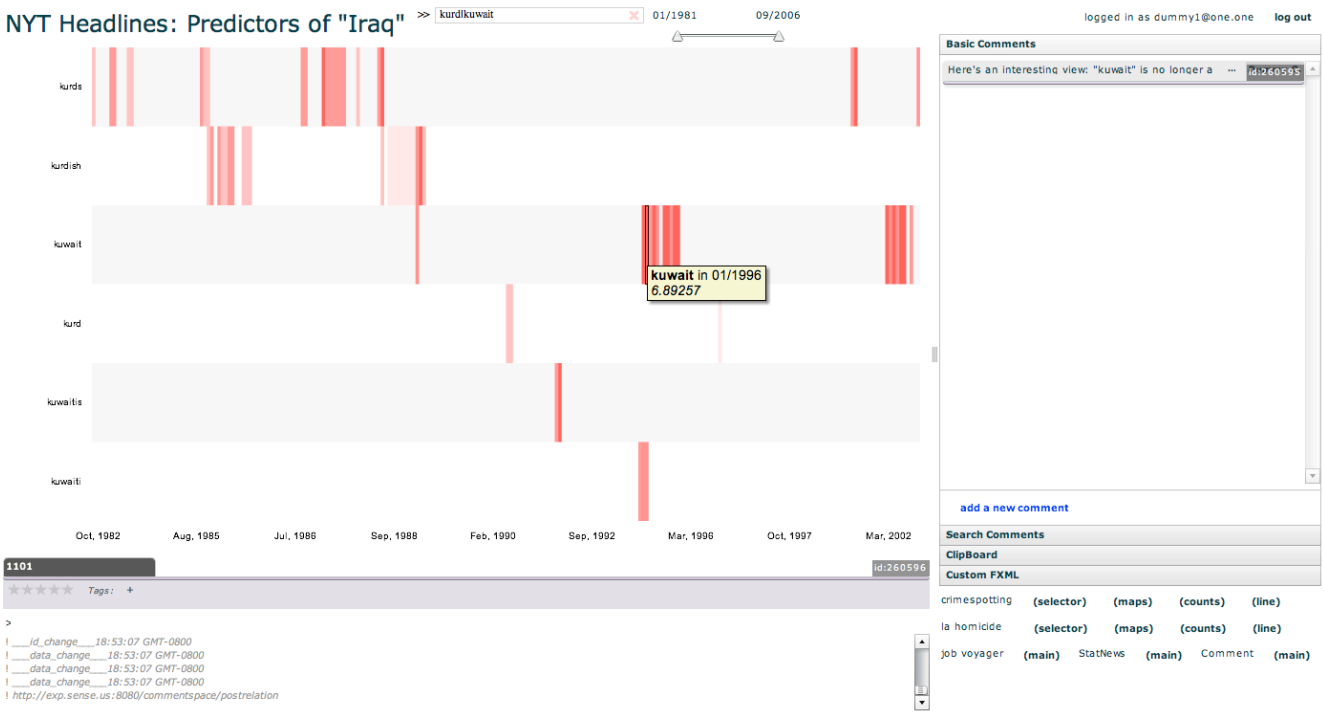

Figure 2. The user has chosen to search for words starting with “kurd” and “kuwait" using the search term “kurd|kuwait”. Hovering allows the user to discover the exact regression coefficient for the particular word in that particular month, in this case “kuwait" in January of 1996. Two issues are present here: the lack of predicting words during the Gulf War, and the decoupling between the date range slider and the text search.

The date range slider is a two-thumb slider. The user can click and drag a thumb of the slider to adjust either the starting or ending date. As the slider is dragged, a tool-tip pops up that shows the current selected date. The visualization is also updated (optionally using animation) dynamically with the slider. Figure 4 shows a zoomed in view where the date slider has been used. Here, the user has specified a new starting date as January 2002.

Users can create comments by clicking the “add a new comment” button in the comment pane. In addition, we provide functionality for users to copy links to a *view* of the visualization and paste them into the comments. At this point it is worth remembering that a visualization is a representation of underlying *data*, and so one may instead consider a visualization to be a collection of *views*. In this particular visualization, we can think of the search query and the parameters of the date slider as completely specifying a view on the data.

Each of these comments is then tied directly to the visualization state. If a user is navigating and happens upon a view that has been previously commented on before, the connected comments will show up in the comment pane. Heer et al. termed this model “doubly-linked discussion” and implemented it in their Sense.us system [9]. The future work section details how we plan to extend the current commenting system and thus how we plan to overcome the limitations inherent in this model.

Technical Implementation

The final goal of this project is to allow groups of users, possibly strangers, to comment on and analyze data through visualization in our system. To this end, we implemented the system using ActionScript 3 and Flex 3 in order to allow for web deployment, thus lowering the barrier to access.

```
<visualization>
        <data> … </data>
        <controls> … </controls>
        <operators> … </operators>
        <visualDefaults> … </visualDefaults>
        <statevars> … </statevars>
        <legend show=”false”></legend>
        <transitioner>
                <duration>0</duration>
        </transitioner>
</visualization>
```

Figure 3. Skeleton overview of the flare XML (FXML) visualization descriptor. A sample attribute is filled out for the <legend> block, and a sample child node is filled out for the <transitioner> block.

The comment pane and layout of the modules are implemented using Flex 3. The visualization is implemented using an XML descriptor of a flare visualization [5], which we term flare XML (FXML). As previously noted, users can apply copy links to specific

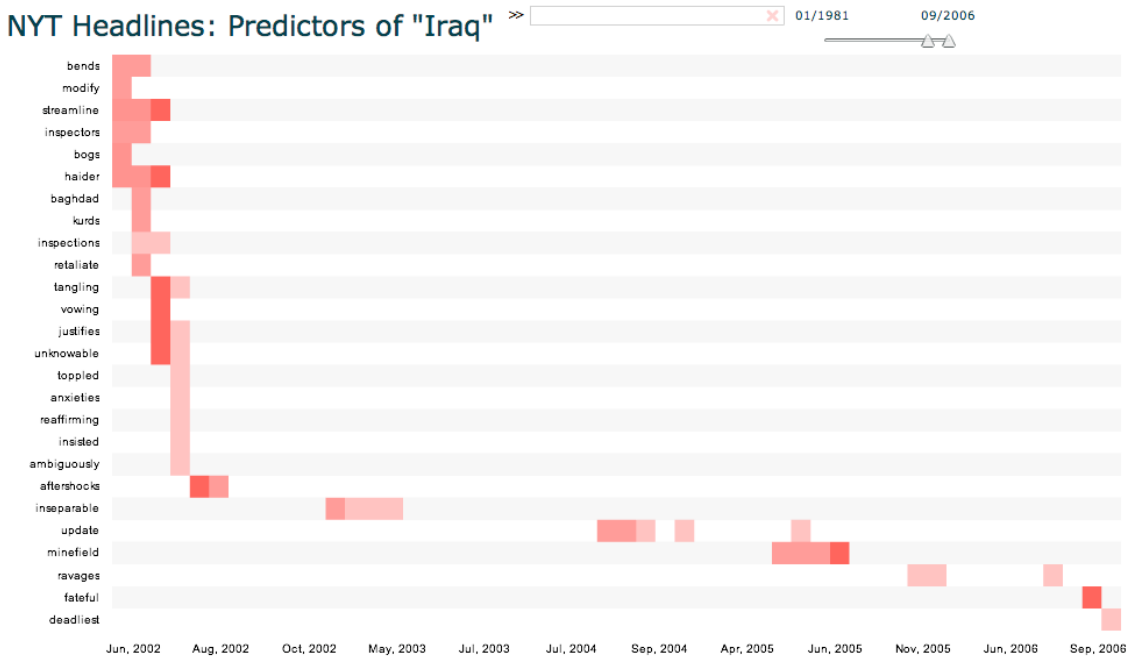

Figure 4. Zoomed in view where the user has specified a date range from June 2002 to September 2006. No word filtering is present in this display as the number of Iraq predicting words are small between 2002 and 2006.

views of a visualization and paste links into their comments, and each of these views has a specific state. In order to both assign state to a visualization and ease the authoring burden of new visualizations, we chose to fully describe a visualization through FXML.

A skeletal overview of the FXML descriptor is shown in Figure 3. The tag names are aligned with the flare methods and classes they are meant to invoke, and each tag has children corresponding to the attributes and functions of the associated flare classes. For example, we can load data from a delimited text file by putting a <delimitedText> with the following format:

```
<delimitedText
url="http://exp.sense.us/data/nytiraq6_50mod.txt"
delimiter="tab"> <field name=”year” type=”date”/> …
</delimitedText>
```

where the “url” attribute is the location of the data, the “delimiter” attribute is the type of delineation of the data, and the “field” children are an optional specification of a flare DataSchema. The other major tags are customizable in similar ways, although an exhaustive description of the FXML syntax is unnecessary for the thrust of this paper. However, the <controls> and <statevars> blocks deserve some further description.

The <controls> block allows the visualization designer to specify how users can interact with the visualization. These controls can either be custom-written, or they can be from the Flex API. Controls of both types are present in the visualization we implemented. The search box and the tooltip text describing a word and its statistical significance for a current month are custom written controls. The date range slider is a slight modification of the Flex horizontal slider, and the title of the visualization is also a control, as defined in the Flex API.

The <statevars> block allows the designer to specify any of the control attributes as a determiner of the state of the visualization. The functionality of these controls is still specified in the <controls> block. For example, in our visualization our three state variables are “dateSlider.values[0]”, “dateSlider.values[1]”, and “searchBox.query”. These variables correspond to the attributes of the date range slider and the query string in the search box.

While authoring a visualization using FXML requires a certain level of expertise, it greatly speeds up the process of creating a new visualization, especially if the data is in a flare-friendly format and if a flare layout for the data already exists. We also provide access to the FXML specification of the current visualization, which allows an expert to quickly edit a visualization description in-tool. This means that new data could theoretically be loaded live and certain visual variable encodings could be changed to some degree. As an example, it took us about two days to write a custom layout class for the StatNews data. However, once this was achieved, loading new datasets was

as easy a single line in the FXML. We could also experiment with different control placements and color encodings by quick, line-length modifications. Further plans for the FXML descriptor are detailed in the future work section.

Algorithmic Details

One of the primary challenges in dealing with this dataset was its sheer size. As no stemming was performed, there were a total of 160,624 distinct words that occurred in the corpus. This resulted in an approximately 151 megabyte text file of features, as each word was assigned a statistical significance per each month. Given the limitations of Flash it is impossible to store all this data internally. A possible solution would be to query a database. We did not implement this as we plan to have direct access to StatNews analysis results through a separate REST API, thus hopefully voiding issues of overly large datasets and expanding the possible analyses we can visualize.

In the interim, while we work out details of the API and the analyses we wish to run, we pre-processed the data to reach a Flash amenable size. We used a simple C++ script that took as input the maximum absolute value significance coefficient and output only the words that exceeded that significance coefficient at some date to accomplish this. In the figures shown in this paper we are using data thresholded at a significance of 6.5, thus reducing our data set to a total of 1987 total data points and roughly 250 unique words. A side effect of this rather high threshold is that we do not retain any negative significance values, that is, we do not retain words that predict the absence of the word "Iraq" in the preceding year of headlines.

Even with this data thresholding, we still encountered efficiency problems. Users should be able to explore the data in real time, but even with rendering less than 2000 flare DataSprites we were experiencing significant choppiness. Disabling animated transitions (shown in the FXML of Figure 3) helped to mitigate the problem somewhat, but more aggressive culling of our display was necessary. In any case, displaying data for over 200 words simultaneously did not prove very readable, as the cells were very small and the y-axis labels were inadequate to distinguish all of the words.

We suspected that most of the slowdown was occurring because of rendering. In order to bring the visualization to almost-interactive levels, we chose to apply further filtering on the data by only displaying the top 100 most "salient" words. Our definition of salience in our current prototype is simply the number of months a word was statistically associated with "Iraq", but one could imagine many other definitions of salience. This is expounded upon in the future work section. The speedup is quite noticeable, but there is still improvement to be done in this area.

DISCUSSION

Looking at the initial layout in Figure 1, one can already see a few interesting trends. However, it is not clear whether these trends are artifacts of the underlying statistical analysis. The words are ordered by date of first significant statistical occurrence, so a certain "lifespan" in predicting "Iraq" is visible. This lifespan is approximately a year, which matches with the rolling horizon analysis, so this is in itself not a particularly enlightening revelation.

However, there are some words that occur beyond their "lifespan". Looking at Figure 1, "pullback" first occurs in March of 1982 and experiences a resurgence in 1994-1995. Likewise, "Kuwait" first appears in July of 1990 and experiences two resurgences, one between 1995-1996, and another between 2001-2002. This is certainly interesting, as it indicates that Kuwait and Iraq were uniquely associated in the headlines during these two periods.

Another interesting feature to note is the conspicuous lack of any predictive words during the period of the Gulf War (1990-1991). This can be seen both in the overview of Figure 1 and the zoomed-in Figure 2. Our hypothesis for this phenomenon is that Iraq presumably dominated the headlines during the Gulf War period, and so no one word was a particularly good predictor for Iraq. This leads to a possible alternative way of looking at the data. It may be interesting to look for periods in time where many words have a relatively low prediction coefficient instead of periods in time where one single word strongly predicts the target word. This may also point to using a different metric such as co-occurrence.

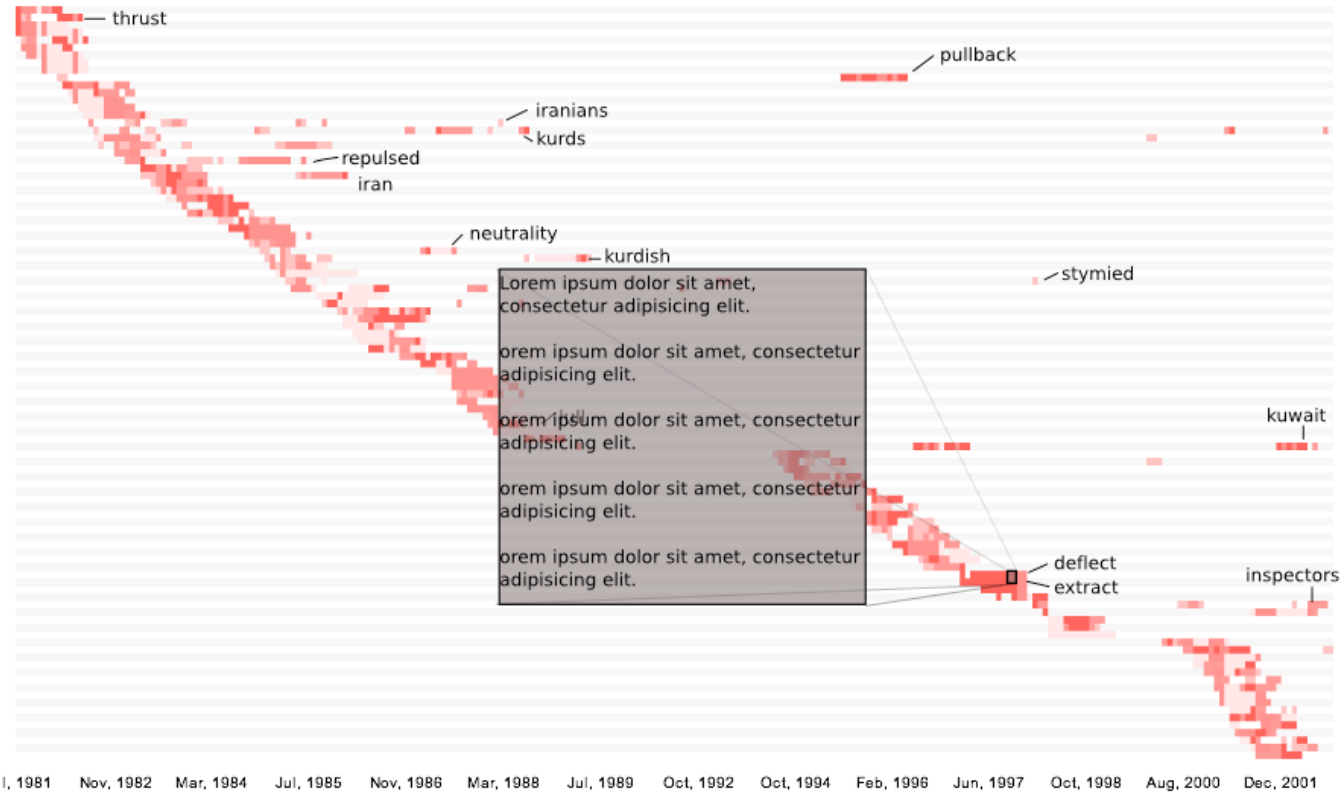

Figure 5. A proposed future interface. Note the intelligent labeling of salient words, and the retrieval of relevant headlines on demand.

THOUGHTS ON EVALUATION

We did not undertake a usability evaluation on the system, as there are still many improvements to be made. We have also not achieved full integration with StatNews at this stage. However, we outline possible evaluation and visualization design strategies for a working prototype of the system.

Our primary interests are to design an effective visualization that uses the StatNews analyses to reveal trends in the news corpus as well as how well our system can facilitate collaborative analysis. To some extent, we can measure how well we achieved both goals simultaneously.

As the discussion reveals, it is unclear as to what exactly the statistics reveal about the corpus, but more so how a social or political scientist would interpret the results and incorporate it into his or her research. We have taken the first steps of inquiry in this direction and had an enlightening discussion with Professor Sophie Clavier, a political scientist with San Francisco State University. Her suggestions are outlined in the Future Work section, and further discussion with other possible users of this data would help us achieve our first goal of effective visualization.

Evaluating how we facilitated collaboration is a trickier concept. Given some concrete use scenarios from our discussion with possible users of the system, we could conceivably evaluate our system in a two-phase process. In the first phase, we would gather a group of users familiar with the data, or at least with vested interest in the data, and ask them to explore. We could seed the discussion with a few starter comments and observe the users as they proceeded with their analysis or exploration of the data. This process is similar to the method Heer et al. pursued for their Sense.us system [9]. Our expectation is that users would discover interesting patterns in the data or formulate hypotheses and possible conclusions.

We could then use the results and comments from the exploratory phase to create a task-based evaluation. We would hopefully be able to use a number of conclusions arrived at by the participants in the first task to present a second group of participants with concrete questions. Balakrishnan et al. assigned their participants a question with a definite answer when analyzing synchronous remote collaboration [2], and we could conceivably take a similar approach.

Other methods of evaluating the efficaciousness of our system could stem from Heer and Agrawala's work in specifying design considerations for visual analytics [8]. For example, a crucial piece of any collaborative analysis system is grounding [4]. One way to enable grounding is to incorporate features that allow for easy deictic references,

that is, unambiguous ways of referring to objects [10]. The words "this" and "that" are deictic references, as are direct links to views. We can obtain a quantitative measure of how well we enabled deixis by a count of both links and deictic words, thus giving us an idea of how well we enabled grounding.

FUTURE WORK

Improvements to existing visualization

One of the most egregious shortcomings of the current visualization is the poor behavior of the axes. In Figure 1, where all the words and the complete date range is displayed, the y-axis labels are chosen automatically by flare's overlap elimination algorithm. We are therefore not taking into account the actual "salience" of a particular word, however that may be defined. The date labels on the x-axis are also not very useful in the fully zoomed-out view. In order to remain readable they must be of a certain size, but they then end up spanning long periods of time.

We thus seek to apply a salient labeling algorithm to the visualization, perhaps with user input as to the definition of salient. In culling the words to display we are assuming one definition of salience, that is, the total number of "significant" months a word possesses. One could imagine using the range of "significant" months as a different salience measure, and the selection of an appropriate salience measure could easily be handed to the user. Figure 5 shows a possible solution, where we have chosen salience to be the largest date-range. It also displays an extension to the visualization that will be possible once the StatNews API is fully complete; this is described in the next subsection. However, if the number of words is small enough (as in Figure 4), standard axis labeling is still a viable solution.

Other minor issues should be addressed. One is the unintuitive decoupling between the date range slider and both the query box and automatic word culling algorithm. For example, in Figure X, the date slider indicates that the data spans the full range of 1981 to 2006, while the visible data actually spans from 1992 to 2002. This is because the matrix cells change height and width to accommodate all of the screen space, and there are no instances of Kurd or Kuwait before 1992 or after 2002. Tying the date slider more tightly to the view would be beneficial.

Another issue relates to the expansion of the cells. When only a single cell is visible, it fills the entire display and does not provide much information of use. Setting a maximum size would enable better readability.

Finally, we will improve the animated transitions. Due to the word culling algorithm, some words do not appear until the date range has been adjusted or a search has been initiated. These words appear to appear out of nowhere, whereas really they should emerge from their putative location on the initial visualization. In addition, when searching for a word, there is no special indication of the words that match the query. Because up to a hundred words could be animating at once, it will be necessary for us to highlight the salient words during their animation in order for the user to follow their changes in location and size.

StatNews expansion and integration

Our system would be much more compelling with a corpus consisting of articles, instead of just headlines. We are in possession of the data, but all that remains is to integrate our visualization system with the StatNews Web API. We are not yet at the point where the data is returned quickly enough for interactive speeds, but a possible solution may be to cache all the data first.

Our talk with Professor Sophie Clavier showed that an essential feature to implement in our system is the ability for comparison. That is, the comparison of different news sources, or different columnists. We will need to implement the capability to display two or more visualization concurrently, although with our planned FXML extension and the porting of the comments layout to an FXML specification this may be a trivial extension. In addition, Professor Clavier desired a more accurate method to input dates. She was not so concerned with exploration as she was with specific date ranges, and much shorter date ranges than a month. For example, she was curious about the period of the Kosovo War, from March to June of 1999. Dynamic querying of this data from the StatNews API is necessary to recover this granularity of data.

Finally, we wish to allow the user access to representative headlines and articles. Figure 5 shows both the proposed labeling as well as a possible interface to relevant headlines.

FXML documentation

The FXML specification is still in its infancy and constantly evolving as we build more visualizations for our system. It is our aim to eventually have FXML mature to the point where casual developers can use it to quickly build visualizations that interface into our system. This means that we will codify and formalize the language as well as refine the existing layouts to allow the developer more flexibility.

Comment restructuring

More closely related to the collaborative aspect of this project, we plan to investigate multiple ways of restructuring the comments. Our immediate goal is to implement a basic thread structure, such as one would see in a typical web forum, but we are also interested in more specific structures. One area we have been particularly interested in is argumentation structures. Researchers in the law and AI community have researched argumentation extensively, resulting in a myriad of models (e.g. [6]) and some prior work in visualizing these models from a pedagogical standpoint [13]. However, most of these models are too complicated for our purposes, although we plan to adapt certain concepts (such as *evidence* and

warrants). We hope that by providing an argumentation structure for discussion we can allow users to more stringently formalize their thoughts and so form hypotheses and reach conclusions in a more efficient fashion.

We would also like to be able to specify the comment layout in FXML. Not only would this allow us flexibility in the layout, but it also allows us to view the comments as a visualization in of themselves. This is attractive as it generalizes our system to a set of linked visualizations, a paradigm into which web sites, video, audio, and plain text would also fit.

CONCLUSION

In this paper, we described an interactive, collaborative visualization of data resulting from a statistical analysis of the New York Times headlines. We outlined a number of implementation challenges and technical details, noting that there are still many challenges to overcome, particularly in the domain of efficiency. We also discussed some of the advantages and disadvantages of the current visualization. Finally, we outlined the many future directions we plan to take this project into.

ACKNOWLEDGMENTS

We would like to thank Maneesh Agrawala, Jeff Heer, and Wesley Willett for their insightful advice and guidance during this project.

REFERENCES

[1] Albrecht-Buehler, C., Watson, B., Shamma, D.A. TextPool: Visualizing Live Text Streams. *IEEE Symposium on Information Visualization, 2004*. INFOVIS 2004. 10-12.

[2] Balakrishnan, A.D., Fussell, S.R., Kiesler, S. Do Visualizations Improve Synchronous Remote Collaboration? In *Proc. Of CHI 2008*. 1227-1236.

[3] Bederson, B.B., Hollan, J.D. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. *Proc. of the 7th Annual ACM Symposium on User Interface Software and Technology*. UIST '94. 17-24.

[4] Clark, H.C. and Brennan, S.E. "Grounding in Communication". Chapter 7, Perspectives on Socially Shared Cognition.

[5] flare visualization toolkit. http://flare.prefuse.org

[6] Gordon, T.F., Karacapilidis, N. The Zeno argumentation framework. *Proc. Of the 6th International Conference on Artificial Intelligence and Law, 1997*. 10-18.

[7] Hearst, M. Informational visualization and presentation. PowerPoint Slides. http://www.sims.berkeley.edu/courses/is247/s02/lectures/TextAndSearch.ppt.

[8] Heer, J., Agrawala, M. Design Considerations for Collaborative Visual Analytics. *IEEE Symposium on Visual Analytics Science and Technology, 2007*. 171-178.

[9] Heer, J., Viégas, F., Wattenberg, M. Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization. *Proc. ACM CHI*, pp. 1029-1038, Apr 2007.

[10] Hill, W.C., Hollan, J.D. Deixis and the future of visualization excellence. *Proc. Of IEEE Conference on Visualization, 1991*. 314-320.

[11] Many Eyes. http://manyeyes.alphaworks.ibm.com/manyeyes/

[12] Newsmap. http://marumushi.com/apps/newsmap/newsmap.cfm

[13] Reed, C., Rowe, G. Araucaria: Software for Argument Analysis, Diagramming and Representation. *IJAIT*. 2004.

[14] Rennison, E. Galaxy of news: an approach to visualizing and understanding expansive news landscapes. In *Proceedings of the 7th Annual ACM Symposium on User interface Software and Technology*. UIST '94. 3-12.

[15] StatNews. http://statnews.eecs.berkeley.edu/

[16] TextArc. http://www.textarc.org

[17] 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 *Proceedings of the 1995 IEEE Symposium on information Visualization*. INFOVIS '95. 51-58.

[18] Wordle. http://www.wordle.net/