

Nested Year Timelines (NYT): A Hierarchical Timeline Visualization for Browsing News Topics

Michael Beswetherick
University of Washington
mbesweth@uw.edu

Joseph Kohlmann
University of Washington
kohlmann@uw.edu

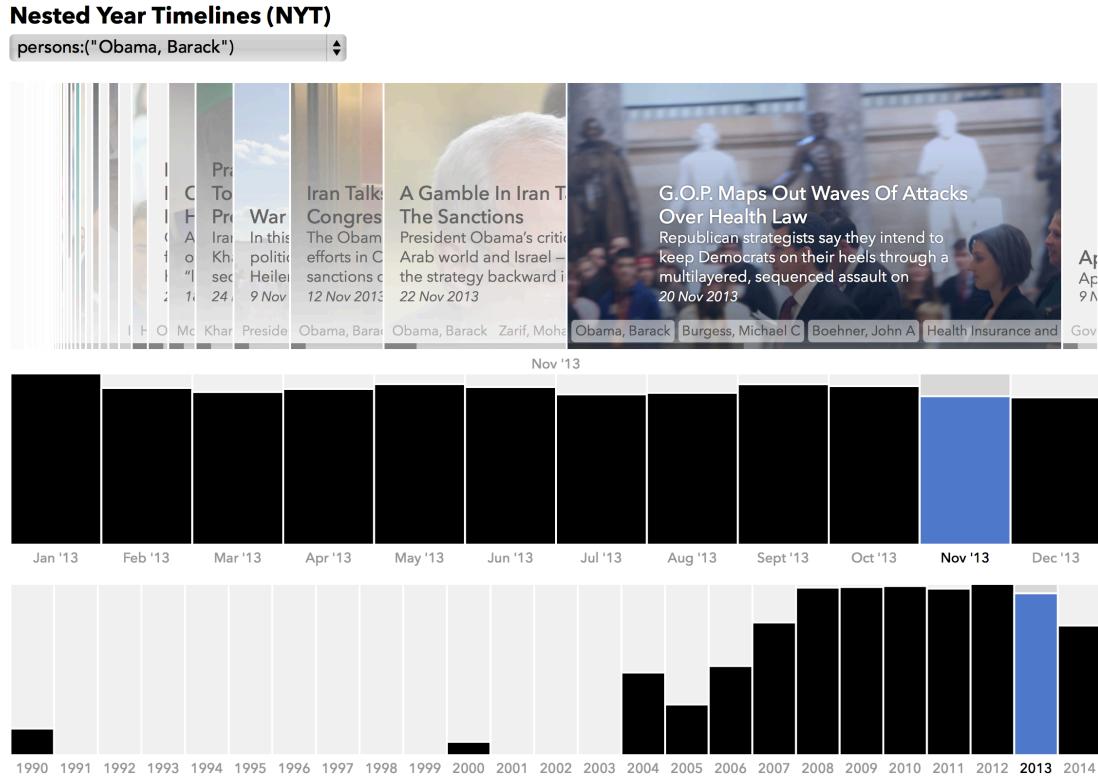


Figure 1. A Nested Year Timeline displaying nearly 20,000 articles from the New York Times referencing Barack Obama. The bar charts display the number of articles per year (2013) and per month (November '13). The article timeline on top displays the selected month of articles. Brushing and clicking support progressive exploration of dataset.

ABSTRACT

In this paper we describe the design and implementation of an interactive news system which allows for exploring articles in The New York Times. Based on a design evaluation of the current Topics Pages in The Times, we have developed a system that allows for a holistic and up to date view of a news topic. We propose timeline based browsing of search results using filters on articles' published dates. Our implemented system, Nested Year Timelines (NYT) is a proof-of-concept demonstration of what news browsing could be. We discuss future uses of timeline browsing and what is possible with it.

Author Keywords

Articles, D3, JavaScript, media, news, Python, timelines.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): data visualization.

INTRODUCTION

The current approaches to generalized browsing of news articles over time, especially those related to a specific person or organization, could greatly benefit from improved visualizations. Currently The New York Times has a Semantic API, which exposes human-readable pages for each person, organization, location, and subject descriptor in the taxonomy. These views amount to little more than walls of text, however, and they only extend to the articles associated with a term without any interlinking of concepts, unlike, Wikipedia's deeply cross-referenced articles.

We propose a visual system, named Nested Year Timelines (NYT), in which semantically relevant content is displayed in an interactive timeline that affords multiple modes of exploration. This paper begins with a design review of the current NY Times Topics pages. It is then followed by a

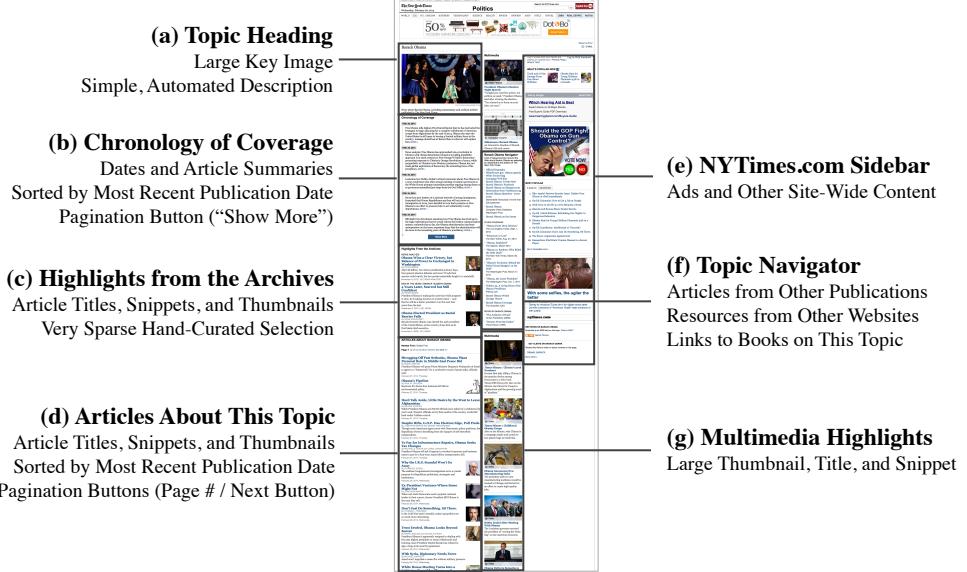


Figure 3. Content analysis of a Times Topics page [5]. The page requires significant vertical scrolling. Despite its large area, only shows 45 New York Times articles related to the topic (excluding off-site links, ads, and the site-wide sidebar).

design proposal of a new browsing system. Subsequently, we discuss implementation details of our proposed system. We conclude with a discussion of future work and the merits of timeline-based browsing.

RELATED WORK

ThemeRiver is a visualization tool based off the “time flows as a river” metaphor, encoding the popularity of a news topic in a stacked line graph to display trending themes and how they change over time [1].

“Narratives” visualizes news stories on a temporal axis and uses other methods to show related concepts [2]. Narratives can show spikes in data references using a simple line graph. With days encoded on the x-axis and the frequency of a term mentioned in an article on the y-axis, one can see when a certain term is trending.

SemaTime is an approach to temporal information viewing which uses several techniques: multiple coordinated views, zooming, categorization, and filtering [3]. While the system is highly complex, it offers many interaction inspirations.

Additional sources of design inspiration for temporal data can be found online. The New York Times “Front Row to



Figure 2. The “Front Row to Fashion Week” interactive feature, uses a fisheye lens effect to widen the runway models’ portraits while brushing.

Fashion Week” interactive feature uses a fisheye lens effect to cluster large amounts of data into one compact space [4]. The images expand to their full width when brushing over them, as shown in Figure 2.

With the advent of APIs delivering large amounts of scalable data, news visualizations have greatly improved. Since ThemeRiver was written in 2000, it lacked the dynamism a modern news visualization might have. While Narratives has the benefit of being in a modern era of data visualization, its presentation of data does not take advantage of modern tools.

We aim to take advantage of modern web capabilities such as dynamic loading of data, while keeping in mind the highlights of this related work. ThemeRiver and Narratives are great at showing how data evolve over time, and The New York Times fashion piece is great for browsing through large amounts of data.

Design Evaluation of the New York Times Topics Pages
 The New York Times has aggregate news pages for people and topics in the news, such as the President of the United States [5]. These topic pages are a blend of hand-curated and automatically acquired content, but offer little room for dynamic data exploration. The current approach to presenting articles results in a redundant, sparsely populated layout that requires significant vertical scrolling and pagination to explore.

Despite the large area the Times Topic page occupies, as seen in Figure 3, it presents a low number of relevant articles, and worse, sometimes displays the same articles twice. Redundant lists of chronologically arranged articles, as shown in Figure 3(b) and 3(d), combined with the very

abbreviated hand-curated highlights in Figure 3(c). The designers included pagination buttons in these redundant sections, perhaps in an attempt to encourage exploration, albeit though endless pages of text descriptions. Altogether, excluding non-article content appearing in Figure 3(e) and 3(f), the topic page for Barack Obama only displays 45 articles.

As evidenced by even the most rudimentary analysis of these topics pages, The New York Times has an incredibly rich data set of articles that is not being fully exposed in an optimal way to the publication’s readers. These presentation issues make it quite difficult to understand and explore the evolution of the topic’s coverage over time. The shortcomings of the topics pages have motivated us to create an effective way for readers to browse through much more of a topic’s news content, without redundant content or poor space utilization.

METHODS

We implemented concepts that form Nested Year Timelines (NYT) concept as a system of linked timelines, displaying articles hierarchically at three levels of detail: all years, all months within a year, and all articles within either a year or month. The article display uses a fisheye lens distortion effect to allow readers to efficiently skim through article titles, snippets, dates, keywords, and images. This hierarchy uses a divide-and-conquer approach to display even tens of thousands of articles retrieved from the New York Times Article Search API within a variably-sized browser window, without any vertical scrolling. Furthermore, the chronological partitioning of the data set allows us to use other visualization techniques, such as the logarithmic bar graphs shown in Figure 4, to give readers a qualitative sense of how many articles appear in a given time slice.

Nested Year Timelines is implemented as two web-based client and server components, using Python on the server side, and JavaScript, D3, and Underscore on the client side.

Server Implementation

Early experiments with the New York Times Article Search API indicated the need for search query aggregation and caching, given not only the API’s rate and per-day request limits, but also its limitation of returning a maximum of 1000 articles—10 articles each per 100 pages—per search query. We worked around this limitation by implementing an API caching system in Python, first as standalone scripts and later as a server-side application using the Python Flask microframework.

This server component responds to search query parameters similar to those used by the NYT Article Search API, determines the number of articles returned by the query. Based on the number of results, it then heuristically separates the query into separate time-delimited queries either by year or by month for particularly dense sets of results, until it has locally aggregated all article metadata for the original query. An additional pass is made to reduce

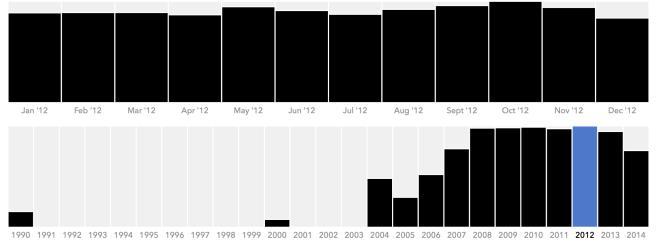


Figure 4. Two levels of articles displayed as bar charts with logarithmic scaling. The lower bar chart shows yearly coverage, while the upper bar chart does the same for each month in 2012.

the results down to a subset data that is reasonably small enough to send to the web browser client application. The server supports dynamic queries to the Article Search API as well as cached queries.

Client Implementation

The NYT user interface uses JavaScript to load in a server-side data set and use D3 to visualize the quantity of articles by year and month. D3 is also used to drive an HTML-based articles timeline, shown in Figure 5, to support rapid skimming through articles. The server backend also supports the ability to dynamically issue new search queries by clicking on the keywords associated with a given article.

NYT initially displays only the years timeline, which displays a bar representing a year for the entire range of the data set’s publication dates. We use progressive disclosure to reveal either an articles timeline or a months timeline upon clicking a year bar. Months timelines are used to further partition a year containing more than 200 articles, revealing an article timeline for the clicked month.

The visualization is very compact in terms of space utilization. The layout conforms to the reader’s browser window width at load time.

Browsing Articles with Fisheye Lens Distortion

Nested Year Timelines needed to support exploration of very large article sets—for example, “Barack Obama” is mentioned in over 20,000 articles in The Times. Even after partitioning articles into years and even months, there may be hundreds of articles on display at once. We optimized this part of the visualization by utilizing the existing D3 Fisheye Distortion plugin [6], which is Based on Sarkar and Brown’s Graphical fisheye views of graphs [7].

With this fisheye distortion for two-dimensional layouts, we were able to significantly improve the experience of skimming through articles, once again omitting pagination in favor of progressive disclosure. We modeled the article skimming interaction after the aforementioned Front Row to Fashion Week interactive feature.

Because this component of the interface involves line-wrapped text and other rich formatting, the articles timeline was implemented using HTML elements rather than the more traditional D3 approach of using SVG elements.

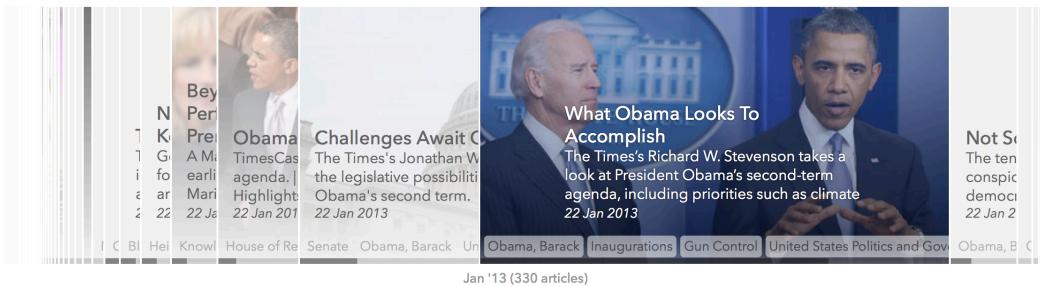


Figure 5. NYT's articles timeline, displaying 330 articles referencing Barack Obama from January 2013. Each tile features the article title, snippet, publication date, keyword links, and an article photograph, if available. The tiles' widths change according to a fisheye lens distortion effect when brushed.

RESULTS

During development we tested Nested Year Timelines with varying data sets. Very popular news subjects, such as the President, returned tens of thousands of results, which took upwards of half an hour or more to aggregate from the Article Search API. The original results produced a 43.3 MB JSON file, which we automatically reduced to a 15.9 MB JSON payload to send to the client-side application. This data payload, containing article information such as URLs, keywords, snippets, and more, represents the greatest potential for compression and optimization. Once the data is loaded, however, the client app exhibits excellent real-time performance, with the hierarchical disclosures providing dual benefits for increased human and graphics drawing performance by reducing the number of article tiles on display.

DISCUSSION

Anecdotal evidence and experimentation suggests that the visualization, or some form of it, can be highly compelling for exploring article search results within a large temporal context. The ebb and flow of subject coverage over time is made readily apparent in NYT's bar graph encodings, as are any gaps or spikes in coverage. Article skimming can also be compelling, with article images adding visual appeal that avoids what would otherwise be a very text-heavy information display.

Jumping from one set of articles to another with the article keyword links also offers a compelling new way of exploring The New York Times' rich archive. The mere presentation of these underlying keywords makes them accessible in a way that encourages exploration beyond just the current article set.

FUTURE WORK

The New York Times is a data-rich entity whose archives merit future exploration, and in this regard, the Nested Year Timeline system serves as a building block for future possibilities. While we are currently able to dynamically load semantically related timelines, some queries are too large to perform live. Though a higher request rate limit would improve performance, the server-side data aggregation approach is a workaround to the limits imposed by the current API infrastructure at The Times.

Since our timelines currently use article images as the background for our article tiles, articles without photos show up as anonymous boxes with blue backgrounds. While it is impossible for The Times to attach media content to those articles, we could improve our data encoding so that tiles without articles have the same semantic weight as tiles with images. Alternately, this could also benefit from more robust media content AP.

Nested Year Timelines is an improvement in temporal news visualization. By stepping away from the wall-of-text metaphor, we are one step closer to data-rich portraits for news topics that encourage interaction and exploration.

REFERENCES

1. S. Havre, B. Hetzler, and L. Nowell, "ThemeRiver: Visualizing theme changes over time," ... Visualization, 2000.
2. D. Fisher, A. Hoff, G. Robertson, and M. Hurst, "Narratives: A visualization to track narrative events as they develop," Visual Analytics Science ..., pp. 115–122, 2008.
3. C. Stab, K. Nazemi, and D. W. Fellner, "SemaTime - Timeline Visualization of Time-Dependent Relations and Semantics," in Advances in Visual Computing, vol. 6455, no. 53, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 514–523.
4. M. Bostock, S. Carter, E. Hinton, C. Horyn, and E. Wilson, "Front Row to Fashion Week - Interactive Feature," The New York Times, 12-Sep-2013. [Online]. Available: <http://www.nytimes.com/newsgraphics/2013/09/13/fashion-week-editors-picks/>. [Accessed: 20-Mar-2014].
5. "Barack Obama - The New York Times," The New York Times. [Online]. Available: http://topics.nytimes.com/top/reference/timestopics/people/o/barack_obama/. [Accessed: 21-Mar-2014].
6. M. Bostock, "Fisheye Distortion," bostocks.org, 21-Jun-2012. [Online]. Available: <http://bostocks.org/mike/fisheye/>. [Accessed: 21-Mar-2014].
7. M. Sarkar and M. H. Brown, "Graphical fisheye views of graphs," presented at the the SIGCHI conference, New York, New York, USA, 1992, pp. 83–91.