

Contextifier: Automatic Generation of Annotated Stock Visualizations

Jessica Hullman

University of Michigan
Ann Arbor, MI, USA
jhullman@umich.edu

Nicholas Diakopoulos

Interaction Foundry
New York, NY
nicholas.diakopoulos@gmail.com

Eytan Adar

University of Michigan
Ann Arbor, MI, USA
eadar@umich.edu

ABSTRACT

Online news tools—for **aggregation**, **summarization** and **automatic generation**—are an area of fruitful development as reading news online becomes increasingly commonplace. While textual tools have dominated these developments, annotated information visualizations are a promising way to complement articles based on their ability to add context. But the manual effort required for professional designers to create thoughtful annotations for contextualizing news visualizations is difficult to scale. We describe the design of *Contextifier*, a novel system that automatically produces custom, annotated visualizations of stock behavior given a news article about a company. Contextifier's algorithms for choosing annotations is informed by a study of professionally created visualizations and takes into account visual salience, contextual relevance, and a detection of key events in the company's history. In evaluating our system we find that Contextifier better balances graphical salience and relevance than the baseline.

Author Keywords

Information visualization; annotation; time series.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Individuals rely on online news in everyday decisions, such as what car model to buy, how to stay healthy, and how to invest their money. Yet with the deluge of accessible news articles, "reading the news" means negotiating the aggregated feeds of hundreds of constantly updating news outlets. A tradeoff exists between swimming in massive amounts of news and compromising with a single article that rarely provides the broader context that can put the news in perspective against other relevant information. As such, many tools for online news such as aggregators (e.g., [21]), summarization agents (e.g., [11, 24, 27]), and even auto-generation of news stories from data (e.g., [26]) pro-

vide a contextualizing function to help a user make sense of and put into perspective online news.

Consider a user who seeks additional context as she reads an article about a company she is considering investing in. Traditionally, she would need to filter and integrate many other articles. Instead, we ask: how can we design an information artifact that puts the article content in perspective against related news about the company? How can we simultaneously ensure that the artifact is comprehensive with regard to highly influential events related to the company, and visually engaging enough to naturally draw her to interact? Supporting context in a news consumption setting requires selecting relevant information that adds useful context given the article content, providing comprehensiveness with regard to extremely newsworthy events, and producing an artifact that is visually or otherwise attentionally salient.

As increased focus on narrative visualizations in industry and academia attests, visualizations naturally perform a contextualizing function when they present data to accompany news [32]. A visualization interface that provides text annotations on points of interest (such as given a news story) can ease a user's interpretation and suggest conclusions [15, 32]. However, in the aforementioned decision about whether to invest in a company, there is a need for an on-demand visualization that provides context on the company's performance, such as through a stock time series annotated with relevant information given the article content. In the absence of skilled designers, an automatic solution is desirable, but auto-generated visualizations in most domains are typically "context blind," limiting their value. As a financial example, Google Finance [12] (Figure 1) offers



Figure 1: Auto-generated Google Finance visualization [12]. Only one form of "context"—date range—can be used to customize the graph.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2013, April 27–May 2, 2013, Paris, France.

Copyright © 2013 ACM 978-1-4503-1899-0/13/04...\$15.00.

a single graph for a company with date range parameterization and the ability to manually add other companies' series. Yet this form of customization calls for prior knowledge about potentially important comparisons.

In this paper, we consider the potential for automatic generation of narrative visualizations to provide context to online news. We demonstrate a system called Contextifier that narrows this broader challenge to financial decision making from news about companies (see Figure 2). The system automatically creates a stock timeline graph and chooses customized annotations for the line graph visualization with reference to the content in a news article. We describe how the Contextifier system and underlying algorithm combine 3 orthogonal features: linguistic relevancy, visual saliency, and analysis of article volume for a particular company. Our contributions include an analysis of professionally annotated news visualizations, the operationalization of observed principles into features that Contextifier uses to automatically generate annotated line graphs, and the evaluation of the features and their perceived value.

RELATED WORK

Graphical Annotation

Storytelling is a paramount concern in journalistic visualization and infographics as it aids in explaining complex news [5]. Directing user attention and guiding interpretation using narrative formats distinguishes narrative visualizations from other exploratory or communicative styles of presentation [32]. In such contexts, text annotations incorporated in the visualization guide users' interactions with the artifact [32], explain what the data means [5], and prioritize certain interpretations of the data [15]. Traditional advice for visualization annotation [7] overlaps with this char-

acterization, suggesting using captions to provide "a clear, direct explanation of the data that are graphed and the inferences drawn from the data" (pg. 55).

Outside of narrative visualization, graph annotation research has focused on annotation added by end-users. In the Sense.us system [13], asynchronous users communicated interesting insights about a visualization by adding textual annotations to interactive graphs. Asynchronous collaborators' freeform annotations on line graphs motivated Kong and Agrawala [19] to develop algorithms for identifying the corresponding perceptual parts of the chart. These include peaks, valleys, and slope segments, first identified as visually salient by Hoffman and Singh [14]. More recently, Kong and Agarwala [19] contribute a system for automatic generation of graphical overlays that include annotations and statistical summaries and highlighting of important marks. Our solution differs in that we automatically generate textual annotations relevant to a "context article" without requiring explicit interaction. The design of our system is also informed by how professional visualization designers use annotations to communicate news.

Kandogan's [17] just-in-time descriptive analytics (1) automatically identify visual features, such as clusters, outliers, and trends, (2) determine the semantics of such features by performing statistical analysis during interaction, and (3) enrich visualizations with annotations that describe semantics of visual features and facilitate interaction. As described below Contextifier focuses less on this "observational context" annotation and more on "additive context": information that provides background or perspective on the data. Other systems include TweetInfo [22], which annotates tweet volume graphs with relevant events from tweets.

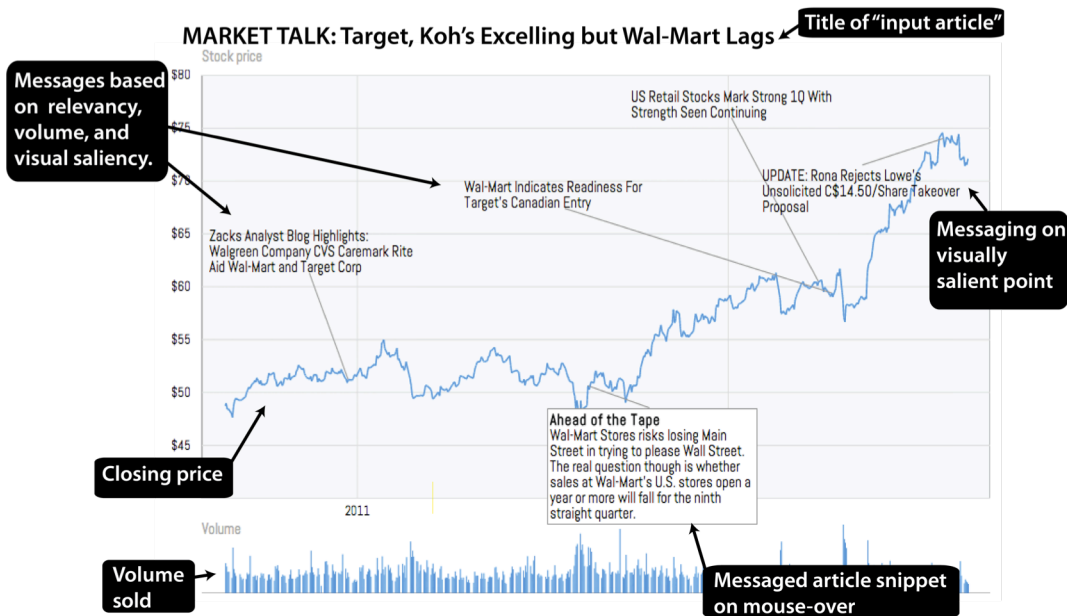


Figure 2: An annotated visualization produced by Contextifier. Graph labels and arrows point out interface features.



Figure 3: Interactive inflation graph [1] provides pop-up annotations citing a relevant news article to the topic.

In contrast to the visualization generation systems available in the financial domain we target, Contextifier considers information specific to a given user's news-based information need to choose annotations for a stock line graph. Google Finance visualizations (Figure 1) are "context-blind" in that details of a user's information need are restricted to a date range by default, resulting in "one-size-fits-all" artifacts for any given company.

Finding Related and Relevant News

With the increased amount of news online a key challenge is finding related articles given an input article. Systems include NewsJunkie [11], which focuses on finding novel information for tracking a story given a user's prior reading history. Others focus on identifying events along a timeline given a news corpus [6, 35]. The Contextifier system similarly incorporates information on volume of articles over a timeline. Additional systems for identifying and/or generating relevant news artifacts summarize news by producing timelines [34], explanations for events [25], and even weather forecasts [18] but are less relevant to the visualization generation goal of Contextifier. Shahaf et al.'s system [33] takes into account the end points of a given narrative and threads a coherent story. While this body of work provides various useful observations, none of these projects match the target stock visualization use case.

ANNOTATION CONVENTIONS & IMPLICATIONS

To inform our selection of messaging features in Contextifier with insight on how designers annotate visualizations, we conduct a qualitative analysis of text annotations on an independently-curated sample of 136 professional human-generated visualizations created between 2000 and 2012 by designers at the New York Times and The Guardian [31]. We sought to answer two questions about annotation conventions to inform our features: (1) How do annotations function in narrative visualization?; (2) what forms does annotation take; and (3) how frequently do they occur?

Two coders analyzed the sample after removing several instances that did not represent cases of abstract data being visualized (e.g. photo-montages, interactive models or simulations). An initial coding protocol described annotation types of interest to ensure the consistent coding of exemplars. Because the visualizations were typically part of a larger multimedia webpage, we limited the analysis to the

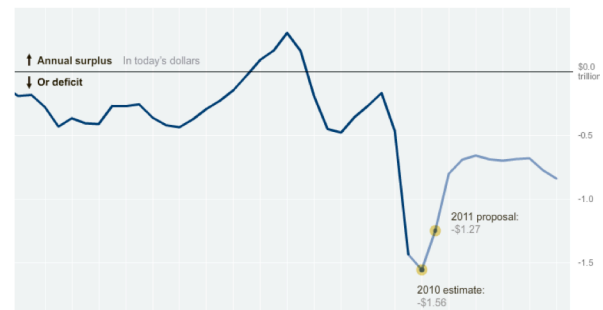


Figure 4: Observational annotation highlights the lowest budget forecast over the time span [8].

frame in which the visualization was presented. The protocol excluded conventional types of textual annotation on graphs including axes labels, creation date or author information, data sourcing or other provenance information like notes on aggregation, instructions, variable definitions, legend text, and introductory or summary text appearing above the graph. While such uses of text are interesting in their own right with regards to the relationship each holds with the visual features (see [2]), we chose to focus on uses of text annotation that are more unique to data storytelling.

Observations of Annotation Types

As coding progressed, we noted where examples appeared to represent different implementations of the same basic contextualizing function. These underlying functions were the basis for two coding categories, (1) *additive*, and (2) *observational* annotation. Of the 136 visualizations analyzed, 120 (88.2%) contained at least one of these forms.

Additive annotation provides context with reference to external information relevant to a topic. In many cases, additive messaging appears to serve a relevancy constraint by presenting external information deemed relevant to the issue at hand. Such annotations provide background, perspectives on an issue, or describe related events. An interactive line graph of inflation over time, for example, provides additional historical information from news articles that are linked to from the pop-up annotations (Figure 3).

Observational annotation gives context by supporting reflection on a data value or group in relation to information present in the visual representation. Such annotations facilitate comparisons, often by highlighting or emphasizing extreme values or other outliers. Cox's "Budget Forecasts, Compared to Reality" interactive slideshow uses observational annotation to highlight the low point of budget forecasting over a 40-year timeline (Figure 4).

We coded several visual levels to which an annotation could be anchored: (1) a single datum such as a time point in a line graph (Single Datum); (2) a group of data or region of the visual presentation (Group/Region), or (3) the entire view of the visualization at a given point in interaction (Entire Visualization). The two types of context-adding function and the three anchors were added to the revised protocol. Finally, graph type was coded to assess whether some

formats are annotated more often to provide context to news. Both coders then independently coded the same set of 38 randomly drawn visualizations from the set of 136. Cohen’s kappa was computed separately for each of the five binary codes. A kappa of 1.0 was obtained for *Type – Additive* and *Anchor – Single Datum* categories while kappas of 0.79, 0.88, and 0.88 were obtained for *Type – Observational Context*, *Anchor – Group or Region*, and *Anchor – Entire Visualization*, respectively. Each coder then independently coded half of the remaining 98 visualizations.

Implications

Our analysis of context-adding annotations on narrative news visualizations informs the design of the Contextifier system in three ways. While both additive and observational context support user interpretation, additive context messaging is more prevalent than observational context in our sample (see Table 1). This leads us to focus on generating additive context messages in Contextifier, while still incorporating methods for identifying visually salient data that users are likely to want more direct observational annotations on (e.g. peak / trough detection). Our coding also uncovered the relative prevalence of different visual scopes to which context annotations were applied. Again, the high prevalence of annotations attached to single datum over those attached to a group or region or entire visualization lead us to focus our initial development attention here, but suggests that future work consider other types of annotation anchors in automated visualization annotation systems.

Our aim is to annotate stock time series to provide historical context relevant to company news. Line graphs are a natural graph choice due to their tendency to represent temporal trends in data series [37]. In the 136 plots, line graphs were the second most prevalent graph type at 11.8% (16) after maps at 46.3% (63). Other graph types represented were bar graphs (11.0%; 15), bubble charts (9.6%; 13), scatterplots (2.9%; 4), node-link diagrams (2.9%; 4), and other graph formats including treemaps and pie charts (15.4%, 21). As we expected, line graphs were more likely to visually represent a temporal dimension than maps (100% of line graphs included a representation of time versus 40% of maps).

		Sample frequency
Annotation type	Additive	73.5% (100/136)
	Observational	49.3% (67/136)
Anchor	Single datum	74.3% (101/136)
	Group or region	50.0% (68/136)
	Entire visual view	35.3% (48/136)

Table 1: Sample frequency of annotation types and anchors.

CONTEXTIFIER: SYSTEM OVERVIEW

Our analysis of annotation conventions in professional visualizations provides several suggestions on how to select features in an automated system. These include the relative importance of adding information through annotations, restricting annotations to single data points, and making sure that annotations are not independent of the graph’s visual

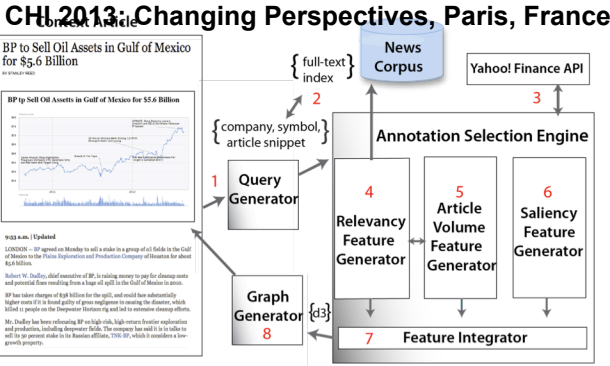


Figure 5: The Contextifier Architecture

features. Our system combines news data with an algorithm that draws on both relevant news and important information regarding the visual data representation in choosing annotations so as to create a relevant, visually engaging artifact.

User Experience

Our system begins with a single news article (which we refer to as the “input article”), and provides an integrated visual representation of relevant information external to this article. A visualization produced by the system can be seen in Figure 2. This information is provided against a visual depiction of stock performance in a visualization that is designed to be embedded directly within the news interface in which the input context article is encountered.

A user gains context from the visual presentation in several ways. An initial visual inspection of the stock closing price timeline summarizes the company’s performance over a given time period. A smaller representation of the volume of stock traded over the same time period is presented beneath, to add context to the closing price series.

The closing price timeline is also enhanced by text annotations that summarize important events and news given the time range shown (Figure 2). These text annotations perform a contextualizing function by allowing the user to skim what other company information external to the context article might be relevant given their information need. For example, a user wishing to understanding what information on prior board decisions heralds for a company’s performance given a similar decision detailed in a new “input article” would be provided with a visualization showing relevant prior news given the new management decision. This might include events leading up the decision, such as tensions between company stakeholders, or similar decisions about company reorganization. In interacting, a user can gain more information about annotations (which default to article titles) by hovering over a title to view the article snippet (Figure 2). She can also select annotations in the visualization in order to navigate to these articles for more in-depth exploration. Thus, the visualization itself becomes an entry point for news exploration.

System Architecture

Contextifier consists of four main components: a news article corpus, a query generator, an annotation selection engine (which is comprised of three feature generators and an

integrator), and a graph generator. These are depicted in Figure 5.

The flow of information is as follows (see Figure 5): the query is generated (1) and matched against the full-text index (2) and the stock series obtained (3). The feature generators compute features from either the text (4, 5) or stock series (6), which are integrated to select annotations (7). A final step (8) generates the line graph using the annotations and series. We describe the components in detail below.

News Corpus

The corpus for the Contextifier system consists of a large set of news articles and corresponding databases from July 2010 to July 2012. To build the corpus, we compiled a list eleven companies across six diverse sectors. We selected companies within each sector that had high numbers of news articles over the period of interest according to the Factiva search engine. Sectors include transportation-airlines (e.g., Southwest Airlines), retail goods (e.g., Costco Wholesale), online software & services (e.g., Yahoo!), fastfood (e.g., McDonald's), healthcare (e.g., Humana Inc.), natural resources (e.g. Exxon Mobil), and chemicals & pharmaceuticals (e.g., Pfizer). We obtained articles from Factiva, drawing from financial feeds, newspaper articles, and postings on news blogs. The number of articles per company ranged from 866 (Humana) to 7,461 (Wal-Mart Stores). After obtaining the articles, we used PyLucene [29] to create a full-text index of unigram counts for each article. We generated a database with tables of full article information and each sector's companies and their stock symbols and synonyms.

Query Generator

To identify topically-relevant articles given the input article, the query generator identifies the most frequently mentioned company in the article. The query to the annotation selection engine consists of the company name and stock symbol. The query generator selects text from the "input article" to query the corpus for similar articles. The inverted pyramid of news reporting states that the first several sentences of a news article often describe the main point of the article [10], and so the query generator uses these first three sentences as a query. It appends to this any sentences that explicitly mention the company name, stock symbol, or a manually-defined synonym for the company name (such as Delta for Delta Airlines). We call this shortened version the article "snippet."

Annotation Selection Engine

Most narrative visualizations encountered with online news contain human-generated annotations. How would an intelligent visualization generator choose annotations modeled after human-generated annotated visualizations? Based on our study of professional human-generated narrative news visualizations, we suggest that more than one dimension must be considered to create visualizations that meet relevancy, spatial, and engagement constraints. In the next section, we describe the algorithm we developed to integrate

linguistic relevance, information on article volume, and visual saliency information in annotation selection.

Graph Generator

The selected annotations (including **article dates**, **titles**, and **snippets**) are passed to a graph generator built in D3 [3], which renders the timelines and places annotations using a force directed algorithm with additional constraints. Article titles appear by default. Snippets appear on hover over titles.

VISUALIZATION ANNOTATION: FEATURE SELECTION

We describe four general desirable features of an artifact for providing context in a news setting. These are (1) **relevancy** considerations dictating that the information presented is topical to the "input article" content; (2) **spatial** considerations requiring that the chosen information be concise so that it is useful over a short use duration, (3) **comprehensiveness** considerations that require that the chosen information not neglect mentioning highly influential events, and (4) **engagement** constraints requiring that an artifact capture the user's attention, such as visually. We designed the annotation selection engine in the Contextifier system by combining the insights from our qualitative study with responses to the question, *What features are suggested by these constraints?*

Overview of Annotation Selection Engine

The Contextifier system can be distinguished from prior work in online news summarization based on its combination of lexical analysis with statistical and visual saliency analysis. The annotation selection engine consists of three analytic components to generate features that contribute valuable contextual information: a *relevancy feature generator* that relies on the keyword index and database to analyze and filter the large set of company articles to a manageable list of candidate annotations, a *volume feature generator* that identifies weeks where the most news-worthy events were likely to have occurred based on the number of company articles that were published, and a *visual saliency feature generator* to provide visually and statistically motivated suggestions. A *feature integrator* aggregates these features and identifies top ranked articles to serve as annotations candidates. By combining these features in a single algorithm, we intend to produce a visualization that better achieves a variety of information goals that range from context-driven to visually-statistically driven. We describe the implementation and benefits of each feature below.

Relevancy Feature Generator

The relevancy consideration above calls for identifying useful additive information that is relevant given the "input article" content. *Linguistic relevancy* is commonly calculated using text-processing techniques such as assessing the linguistic similarity between an article and others to find similar articles. A news article that is most central to the topic discussed in a set of articles can be found by combining a similarity measure with graph-based centrality analysis. Our approach combines both methods to identify lin-

guistically relevant information for annotations, and to find the most topically central information given a set of article text identified as relevant.

As described above, the query generator represents the “input article” via the company name, stock symbol, and concise representation of the content using the inverted pyramid schema and company mentions. The relevancy feature generator algorithm proceeds as follows:

- All articles containing the company name are retrieved from the full text index. This set of articles (characterized by indexes, titles, dates, and article body) is termed the *superficially relevant* set.
- Snippets are generated by extracting the first three sentences from the article and appending other sentences that mention the company name or stock symbol. The journalistic inverted triangle concept informs this [10].
- Articles are tokenized and stemmed (using the NLTK library [28]). Stop words are removed as are company identifiers (names and synonyms, stock symbols) that might dilute the similarity calculation between the input article and all article representations in the superficially relevant set.
- The Kullback–Leibler (KL) divergence is calculated for the input article snippet and the snippets for articles in the superficially relevant set. KL divergence is a standard information theoretic measure of the distance between two documents (or sets thereof) represented as probability distributions of the words shared in both. Typically, the measure is not symmetric, requiring separate calculations to capture the distance between the same two documents. Additionally, the method considers only the intersection of the two documents. To overcome these limitations we use a symmetric back-off smoothing method that allows analysis on the entire vocabularies of the two documents while insuring that the probability distributions sum to one [4].
- We cluster the articles by week (seven day) periods, and compute each week’s average linguistic saliency score by averaging the individual KL scores for all articles in that week.
- The average linguistic relevancy scores for each week are normalized by dividing by the maximum average KL distance for a week in the period. To transform this measure into a similarity metric, we subtract the score from one.

Article Volume Feature Generator

A count of superficially relevant article by day is provided to the article volume feature generator. We aggregate the counts by week and normalize by dividing by the maximum weekly count. We expect high points in this sequence to capture times when the company had high media attention, likely corresponding to an important event that week.

Saliency Feature Generator

This feature generator analyzes the visual saliency of the stock time series to identify time points along the series that are likely to attract the user’s visual attention. This is informed by the finding from our study of annotation conventions that about 49% of exemplars used a form of observational annotations that reacted to visual manifestations of features of the data.

Methods for analyzing visual saliency include tools for analyzing image bitmaps, such as the Itti-Koch algorithm [16] for identifying portions of the image likely to be visually salient to users. However, such algorithms are intended for analyzing photographs, which typically display richer pixel maps in comparison to the visually-minimalist line graph format. Methods of saliency analysis designed specifically for graphs like line graphs include scale-space filtering [36] in which repeated Gaussian smoothing is applied to determine what portions of a visualization are consistent despite multiple smoothings. However, a potentially simpler alternative is to use statistical saliency, referring to data that are noteworthy compared to others when represented numerically, as a proxy for visual saliency under certain conditions. We identify simple numerical analyses that consistently generate visual salience information for company stock data. Related work on graphical overlays [19] and perceptually-based annotations [20] provides support for the identification of points like peaks and troughs.

The visual saliency feature generator applies the following analyses:

- Peak and trough detection on closing price generates a binary variable indicating whether an observation is the global maximum or minimum point of the series.
- Peak and trough detection on volume traded generates a binary variable indicating whether an observation is the global maximum or minimum point of the series.
- A variable is generated to capture the change (or delta) over previous time point on closing price by computing the absolute distance between the value for the current date and the value for the previous date. Values are normalized by dividing by the maximum change value.
- A variable is generated to capture the change over previous time point on volume traded by computing the absolute distance between the value for the current date and the value for the previous date. Values are normalized by dividing by the maximum change value.

Final visual saliency scores are computed by first summing the four variables above, then clustering individual time points in the series by weeks. Week averages are calculated and normalized by dividing by the maximum weekly visual saliency score.

Feature Integrator

A feature integrator combines the scores generated by the relevancy, visual saliency, and volume feature generators. Final relevancy, visual saliency, and volume scores are rank

ordered independently, then their ranks combined into an “overall score” using a simple linear model with coefficients of 3, 1, and 0.5, respectively. We chose to set the coefficient for the relevancy score to 3 based on our own expectation that relevancy would be weighted more highly by users as well as observations based on different weightings. Volume is set to a lower value as too many articles based on high volume might detract from the coherency of the messages given the textual and visual information available to the user, increasing their confusion. We note the importance of testing this weighting in the Discussion section below.

After combining ranks, we define the selected annotations by identifying the five weeks with the lowest “overall scores”. For each of the five weeks selected, we find the most representative article to use as the text in that week’s message. The rationale behind this choice is to choose annotations that best summarize notable events or issue in the news that week, as notable events are more likely to be associated with other salient features (e.g., of the stock series). To achieve this we create an undirected graph using pairwise KL-divergences as edge weights between all articles in that week plus the input article so as to control for relevancy. We threshold edges at a maximum limit of 5.85 to reduce graph size, having found that 5.85 was the mean KL-divergence on a set of sample articles across several sectors. We then find the article with the highest degree centrality (omitting the “input article” from consideration) [30].

An example of a visualization created by Contextifier is shown in Figure 2. To test the performance of Contextifier and our design decisions we describe an evaluation below. Our study attempts to isolate the key features generated by the system to test their effectiveness relative to one another, as well to the combined feature visualizations.

EVALUATION

In evaluating the Contextifier algorithm we asked three questions: (1) Do the relevancy, visual saliency, and volume feature generators contribute unique information in annotation selection?, (2) Assuming the features are unique contributions, are visualizations produced using the key features of the system rated higher than those produced without accounting for these features?, and (3) How do user perceptions of visualizations generated using the key features, both in isolation and combined, compare and to one another? By answering these questions we hope to gain further insight to the value of accounting for linguistic relevancy and visual saliency in particular in generating annotated stock line graph visualizations.

Feature Orthogonality

To evaluate whether the feature generators made differing contributions we compared the weeks identified by each generator when we ran 900 randomly sampled articles from our test collection through the ranking system with only one ranking feature turned on at a time. We also examined the ranking of weeks calculated by the algorithms for each of the 105 weeks, comparing the ranked lists by finding the mean Spearman Rho (using a Fisher-Z transform before calculating the mean and then inverting the result, with mean unweighted as the lists all had the same number of weeks). This indicated an extremely low mean correlation between saliency ranked lists and relevancy ranked lists (0.0174) as well as between relevancy ranked weeks and volume ranked weeks (0.0193). While not perfect, saliency and volume rankings were found to display a relatively high correlation (0.75). This may be explained by salience identifying key stock shifts that often correlate with many articles published about the company. For example, a key event, such as the retirement of key personnel would likely drive both stories and stock behavior. There are situations where the two ranking strategies differ (the correlations range from .52 to .88). Nonetheless, a more thorough investigation of the uniqueness of the article volume feature will be necessary in future work, such as by asking domain experts to evaluate the annotations selected with this feature alone. We retain both features in the combined strategy that we evaluate.

Algorithm Comparison

In designing Contextifier we expected that accounting for visual saliency would cause users to feel that a graph better explained trends in the stock performance graph. We also expected that linguistic relevancy would produce graphs with messages that users perceived as more relevant to the input article content. We thus evaluated Contextifier visualizations using rating questions to specifically evaluate these expectations: ‘How well does the graph do at explaining trends and oscillations in the company stock performance?’ to evaluate visual saliency, and ‘How relevant is the information in the graph given the topic and events discussed in the news article?’ to evaluate linguistic relevancy.

To run the study required generating a set of Contextifier visualizations that differed only in the features that the algorithm accounted for. We chose 10 unique news articles from the past year across 5 companies (2 each for Wal-Mart Stores, Yahoo!, BP, McDonald’s, and Pfizer). For each context article, we created four versions as follows:

- **Random annotation (RAND):** five random articles from the “superficially relevant set”.

WSJ: Yahoo Opts to Friend Facebook



Figure 6: A visualization generated using visual saliency features only.

- **Linguistic relevance (REL):** We selected the top five weeks using linguistic relevancy scores only, then found the representative annotation for each as described above.
- **Visual saliency (VS):** We selected the top five weeks using visual (statistical) saliency scores only, finding a representative annotation for each (an example is shown in Figure 6).
- **Linguistic relevance, volume, and visual saliency (ALL)**

This generated a set of 40 unique visualizations. We refrained from testing a visualization generated using only the volume feature generator for two reasons. First, we could not expect users to accurately evaluate the contribution of the volume feature without prior knowledge about the content in the articles. Secondly, the testing for feature orthogonality described above indicated a moderate correlation between the visual saliency and volume features.

Ten students in an information science master's program in our university were solicited as subjects, and compensated using a \$25 Amazon gift card. We expect these subjects to represent individuals who seek and benefit from news visualizations based on their interest in digital informatics. Upon consenting, a subject completed the study online. She was shown the 40 visualizations in randomized order. Each of the 40 screens on which the visualizations appeared also contained the input article above the visualization, which the subject was asked to read. After reading the input article

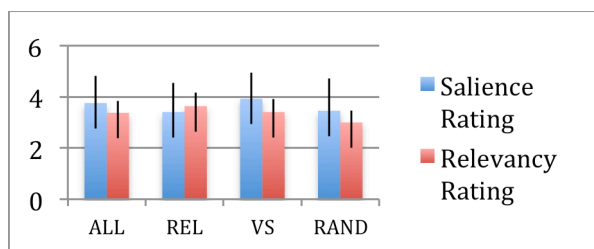


Figure 7: Mean saliency and relevancy ratings. Error bars depict standard deviation.

and examining the graph, the subject was asked to answer the two rating questions for visual saliency and linguistic relevancy using 5 point Likert scales for each.

Results

A total of 393 saliency and relevancy rating pairs were obtained across the 10 subjects. We then pooled ratings across each visualization version (ALL, RAND, REL, and VS) for both the visual saliency and the relevancy rating questions. The mean visual saliency ratings and relevancy ratings are shown in Figure 7. For visual saliency ratings, an ANOVA ($F=6.288$, $P>F=0.0004$) followed by Tukey HSD tests showed the visual saliency type (VS) to be rated significantly higher than either the RAND ($p<0.01$) or the REL visualizations ($p<0.01$). The ALL visualizations were also rated weakly significantly higher for the saliency rating question than the REL visualizations ($p=0.06$).

Similarly, an ANOVA on the relevancy ratings ($F=5.361$, $P>F=0.001$) followed by Tukey HSD tests on these responses indicated the REL visualizations performed significantly better than the random visualizations ($p<0.05$). The VS visualizations and the ALL visualizations were both rated weakly significantly higher than the RAND visualizations ($p=0.05$ and $p=0.07$, respectively).

We conclude from these results that the relevancy and visual saliency feature generators both offer significant contributions over randomly generated graphs in this domain, and are both better than alternatives when it comes to providing annotations that coincide with visually important parts of the graph and are topical to the content of the input article, respectively. Additionally, combining these features performs better than random as expected, but also appears to represent a trade-off when it comes to the objectives of visual saliency or relevance in isolation. In the next section we discuss what these results suggest for future work on refining the combined algorithm.

DISCUSSION

Assumptions and Limitations

Our intention was to explore the potential for automatic generation of annotated narrative visualizations to accompany online news. To do so we chose a specific domain: financial decision-making about the stock performance of a company mentioned in an article. We note that there are features of the financial stock news domain that make it well suited to our approach. Decision-making relevant to stocks often relies on text news about companies such as press releases as well as numerical data like traded volumes and prices, rendering relevancy and article volume (i.e., text) analyses useful. The familiar line graph format used in financial contexts offers the additional benefit of easy yet effective detection of salient visual features using simple statistical analysis of one-dimensional data (i.e., each line in a line graph). We did not, however, evaluate the effect on user perceptions of saliency of adding additional lines to the graph, making this an important step for future work as

adjustments might be required. The linguistic disambiguation accomplished by stock symbols was helpful though not necessary to our goal. We suspect that with minor adaptations the approach can be applied in domains with similar characteristics. Unemployment and economic indicators, temperatures, and sports averages represent cases where news corpora might be constructed by retrieving articles using country and economic indicator co-occurrence, location names and temperature key words, or athlete or sports team names.

On the other hand, certain features of the approach—most notably the saliency analysis—may require considerable adaptation to be used on other graph types or domains. This becomes clearer by generalizing the requirements to:

1. Annotatable data units with at least one variable that maps to a visual dimension,
2. An algorithm for labeling annotatable units with interval saliency values,
3. A database of candidate text that maps to the data units in the visualization (e.g., time, location, etc.)

As realized in the Contextifier system, annotations are text explanations or comments attached to single datum in a visual data representation, making (1) a necessary requirement. Yet requirement (2)'s saliency analysis could be designed in various ways depending on the features of the graphical format. Considering the top four types of graphs observed in our sample of annotated visualizations, we note that maps meet this requirement whenever there is at least one type of geographically-indexed data unit to be visualized (e.g., cities, subway stations). Saliency values could be applied to such data units by computing perceptual saliency using a pixel-based approach such as Itti-Koch [16]. For bar charts, annotatable data units are by default bars representing aggregated values for levels of a discrete variable. Given the relative simplicity of this graph format, we expect that saliency values can be computed simply by analyzing the differences between bar heights relative to surrounding bars. Bubbles in bubble charts could be messaged provided at least one visual feature (e.g., location along the x or y axes, or alpha value) has a corresponding interval variable in the data set. Again image-based approaches to saliency analyses may be required. Finally, (3) need not be a database of news articles, though this is certainly a common case. Quotes indexed by time, or user generated content like product reviews are just a few possibilities.

Our evaluation attests to the unique contributions of relevancy and visual saliency features, arguing for combining these in an annotation selection algorithm. As stated above, future evaluation is needed to determine whether article volume contributes to decision making from a visualization such as in the stock series context we chose. Another important task is to systematically identify the optimal weightings (i.e., aggregator coefficients) and aggregation algorithms (e.g., time scales) as they relate to the user's impression of the visualization and the use case. This calls for further elaborating the conditions under which emphasizing

particular features over others might be appropriate. For example, data-driven needs such as specific cost-benefit projection on stock performance might benefit from higher weighting to visual saliency, while content-driven motivations such as background research on a little-known company might be best matched to high relevancy weighting.

Future Work in Visualization Annotation

Contextifier offers a promising example of how news visualizations modeled after human-produced visualizations can be automatically generated in a specific context: line graphs of stock time series to accompany company news. The high prevalence of maps in news visualizations (supported by our professional visualization sample observations) suggests that spatial context is another relevant way visualizations can support understanding of news. We note that while our current implementation focuses on time-index text, the technique may be more broadly applicable. In situations where the text can be mapped to non-temporal visualized units (e.g., locations on a map) a similar idea could be used to annotate geospatial visualizations. One might create maps of a location mentioned in an article to present relevant background information, or annotate electoral maps produced to accompany a politically-themed news article with annotations providing context on prior elections and significant trends.

It also makes sense to consider how components of the technique could be adapted to annotate different anchor types like aggregations of data, regions of a graph, or the entire visual view. Expanding the relevancy feature generator could allow the technique to be applied for deeper textual analysis within articles, such as annotating specific counterarguments or criticisms to the input article's topic. Numerous possibilities exist for applying state-of-the-art natural language processing techniques toward achieving this and other more complex textual selections of annotations (e.g., [30]). Topic modeling, for example, could be applied such that the aggregate content of the database is used to suggest topics for annotation.

Contextifier's design also leaves room for exploration of how such a system could provide greater transparency and control to the user in annotation selection. Slider controls could allow a user to customize the presentation based on their preferences and needs. Future versions could allow users to customize the date range for which the visualization is generated and to re-focus the visualization based on another input article by selecting messages on the graph. The user's prior reading history might be incorporated, as in [11]. Incorporating user queries could allow more customized refinement of selected annotations.

CONCLUSION

The Contextifier system demonstrates a novel, automatic solution for producing annotated line graphs of stock performance to provide context to a news article about a company. We describe Contextifier's algorithms for choosing annotations, and the findings from a study of professionally

created news visualizations used to inform the system design. We show how accounting for visual salience and contextual relevance each offer unique contributions in the financial time series domain. We evaluate Contextifier line graphs and other line graphs and find that line graphs that take into account visual salience of the graph itself are perceived as better at explaining observed trends, while relevancy analysis of text leads to topically-appropriate graphs. We suggest that a tradeoff between data content and text-based features creates a better balance of annotations.

ACKNOWLEDGEMENTS

The authors thank Ruth Rosenholtz, Nathaniel Twarog, and Robert Finn for early comments on this work.

REFERENCES

- Allan, R. Inflation in the U.K., *The Guardian*, Aug. 14, 2012, [<http://gu.com/p/26aff/tw>].
- Barthes, R. *Image-Music-Text*. Hill and Wang, 1978.
- Bostock, M., Ogievetsky, V., and Heer, J. D3: Data-Driven Documents. *IEEE TVCG*, (2011), 2301-2309.
- Bigi, B. Using Kullback-Leibler Distance for Text Categorization. In F. Sebastiani, ed., *Advances in Information Retrieval*. Springer Berlin, 2003, 76–76.
- Cairo, A. *The Functional Art: An introduction to information graphics and visualization*. New Riders Press, 2012.
- Chieu, H.L. and Lee, Y.K. Query based event extraction along a timeline. *SIGIR*, (2004), 425–432.
- Cleveland, W.S. *The Elements of Graphing Data*. Hobart Press, 1994.
- Cox, A. Budget Forecasts, Compared With Reality, *The New York Times*, Feb. 2, 2010, [<http://nyti.ms/bbxG9g>].
- Dörk, M., Williamson, C., and Carpendale, S. Towards visual web search: Interactive query formulation and search result visualization. *WSSP*, (2009).
- Franklin, B. *Key Concepts in Journalism Studies*. Sage, 2005.
- Gabrilovich, E., Dumais, S., and Horvitz, E. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. *WWW*, (2004), 482–490.
- Google Finance. [<http://www.google.com/finance>].
- Heer, J., Viegas, F.B., and Wattenberg, M. Voyagers and voyeurs: Supporting asynchronous collaborative visualization. *Commun. of the ACM* 52, 1 (2009), 87–97.
- Hoffman, D. D. and Singh, M. Salience of visual parts. *Cognition*, 63, (1997), 29–78.
- Hullman, J. and Diakopoulos, N. Visualization Rhetoric: Framing Effects in Narrative Visualization. *IEEE TVCG* 17, 12 (2011), 2231–2240.
- Itti, L., Koch, C., and Niebur, E. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20, 11 (1998), 1254-1259.
- Kandogan, Eser, Just-in-Time Annotation of Clusters, Outliers, and Trends in Point-based Data Visualizations. *IEEE TVCG*, (2012), 73-82.
- Kittredge, R., Polguère, A., and Goldberg, E. Synthesizing weather forecasts from formatted data. *Proc. of Computational Linguistics*, (1986), 563–565.
- Kong and Agarwala. Graphical Overlays: Using Layered Elements to Aid Chart Reading. *IEEE TVCG*, (2012), 2631-2638.
- Kong, N. and Agarwala, M. Perceptual interpretation of ink annotations on line charts. *UIST '09*, (2009), 233–236.
- Liu, J. and Birnbaum, L. LocalSavvy: Aggregating local points of view about news issues. In *the WWW'08 Workshop on Location and the Web*, (2008), 33–40.
- Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., and Miller, R.C. TwitInfo: Aggregating and visualizing microblogs for event exploration. *ACM CHI*, (2011), 227–236.
- McCay-Peet, L., Lalmas, M., and Navalpakkam, V. On Saliency, Affect and Focused Attention. *ACM CHI*, (2012), 541-550.
- McKeown, K.R., Barzilay, R., Evans, D., et al. Tracking and summarizing news on a daily basis with Columbia's Newsblaster. *Proc. of Human Language Technology Research*, (2002), 280–285.
- Nack, F. and Ide, I. Why did the prime minister resign?: generation of event explanations from large news repositories. *ACM MM*, (2011), 313–322.
- NarrativeScience. [<http://www.narrativescience.com/>].
- Nichols, J., Mahmud, J., and Drews, C. Summarizing sporting events using twitter. *IUI*, (2012), 189–198.
- Natural Language Toolkit. [<http://nltk.org/>].
- PyLucene. [<https://lucene.apache.org/pylucene/>].
- Radev, D., Jing, H., Stys, M., and Tam, D. Centroid-based summarization of multiple documents, *Information Processing & Management*, 40, 6, (2004), 919–938.
- Rooze, Marije. [<http://marijerooze.nl/thesis/graphics/>].
- Segel, E. and Heer, J. Narrative Visualization: Telling Stories with Data. *IEEE TVCG* 16, 6 (2010), 1139–1148.
- Shahaf, D. and Guestrin, C. Connecting Two (or Less) Dots: Discovering Structure in News Articles. *ACM KDD*, 5, 4 (2012), 24:1–24:31.
- Swan, R. and Jensen, D. TimeMines: Constructing Timelines with Statistical Models of Word Usage. *ACM SIGKDD Workshop on Text Mining*, (2000), 73-80.
- Tanev, H., Piskorski, J., and Atkinson, M. Real-Time News Event Extraction for Global Crisis Monitoring. In E. Kapetanios, V. Sugumaran and M. Spiliopoulou, eds., *Natural Language and Information Systems*. Springer Berlin / Heidelberg, 2008, 207–218.
- Wattenberg, M. and Fisher, D. Analyzing perceptual organization in information graphics. *Information Visualization* 3, 2 (2004), 123–133.
- Zacks, J., and Tversky, B. Bars and lines: A study of graphic communication. *Memory and Cognition*, 27 (1999), 1073-1079.