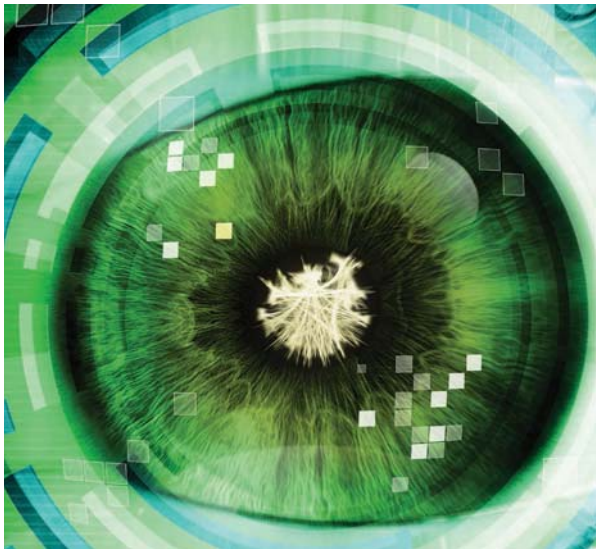
Zuerst ersch. in : Computer ; 46 (2013), 7. - S. 47-55



Real-Time Visual Analytics for Text Streams

Daniel A. Keim, Miloš Krstajić, Christian Rohrdantz, and Tobias Schreck
University of Konstanz, Germany

Combining automated analysis and visual-interactive displays helps analysts rapidly sort through volumes of raw text to detect critical events and identify surrounding issues.

Large amounts of text enter cyberspace every day—whether generated by professionals such as global and local news websites, or by the general public through blogs, social networks, and website commentary. Most generated text is publicly available, but a respectable portion is for internal use only. Responses to surveys on a retailer's website, for example, can be a source of valuable content restricted to a group or institution. User groups have also become important content creators, responding to current events, products, services, and companies.

When appropriately processed, these diverse text streams provide numerous opportunities to receive instant feedback and to monitor and improve business. A company can analyze these streams to pinpoint time-related issues, such as when customers began complaining about a purchasing experience or product quality. The company can then use that knowledge to quickly remedy the situation and thereby stem customer loss and damage to the company's reputation.

News articles, which frequently review companies, brands, and products and often invite public comment, can serve as a backchannel that reflects the consequences of a company's actions, such as updating or repackaging a product. News and social media streams can also be a source of information about emergencies, such as natural disasters or terrorist attacks, or input to political parties about their candidate's popularity.

As these examples strongly imply, timely analysis is critical to operations. In theory, all information is accessible for real-time exploration, but text processing introduces delays. Text streams contain both irrelevant and relevant information and lack semantic structure. To filter the relevant information and extract a higher-level semantic structure (events, stories, topics, sentiments, and so on), analysts require automatic text-processing methods. Relying on these methods creates a tradeoff between analytical speed and results quality; the particular tradeoff depends on the analyst's priorities.

Regardless of the structure extracted, humans inevitably must make sense of the results and draw conclusions for decision making. Interactive visualizations can bridge the gap between computational methods and human analysis requirements—an idea that has motivated the evolution of visual analytics.

Much research in applying visual analytics has focused on combining automatic analysis with visual-interactive displays to help users understand massive amounts of streamed text. Our research in this area has culminated in an approach to analyze text over time according to trends identified through density analysis. In evaluating and fine-tuning our methods, we identified several important open problems in real-time text processing and analysis, and our

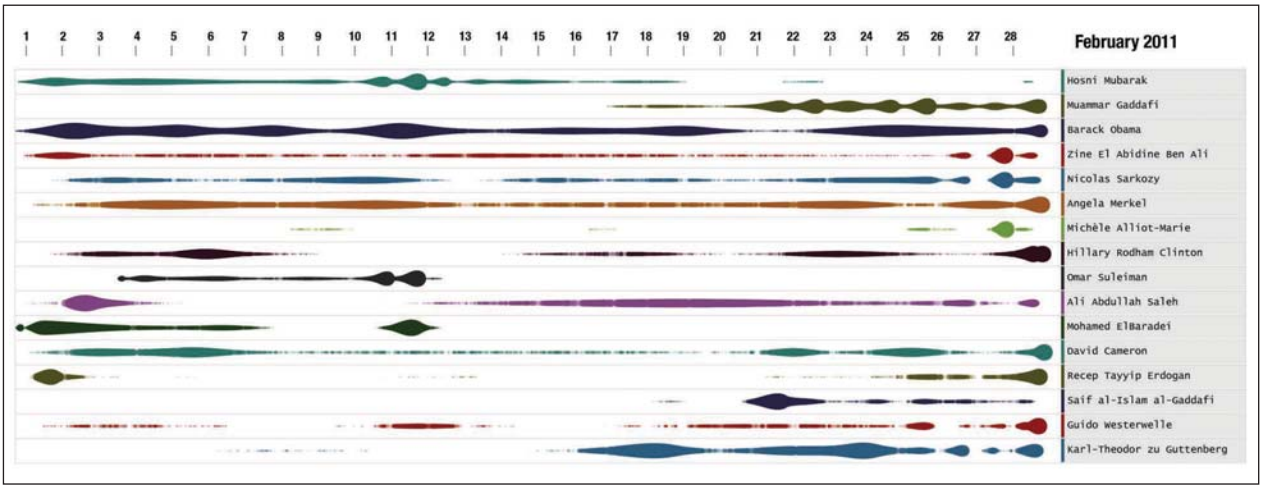


Figure 1. CloudLines temporal-density-based visualization. CloudLines has responded to the analyst's request to monitor individuals in the news during February 2011. A line comprises a series of overlapping circles, each of which represents a news article (document) that mentions the person in that row. Blips signify potentially interesting events that the analyst can explore through zooming.

survey of existing methods was the basis for developing a broad taxonomy of word-, sentence-, and document-based text-processing methods based on tradeoffs in accuracy and efficiency.

VISUALIZING NEWS STREAMS

News events are often interrelated and can evolve or trigger other events. The Arab Spring in 2011, for example, was essentially an extended wave of demonstrations and civil wars. Other events, such as the Occupy movement, occur in parallel all over the world. Such events generate multiple views and ebb and flow in importance.

News aggregators, notably Google News and Europe Media Monitor (EMM), automatically cluster published news articles into stories and rank them by importance scores, with the new information presented first. The idea is to let users explore each story from different viewpoints, not just the publisher's. As time passes, however, new information quickly replaces the old, and it becomes progressively harder to track changes in the news landscape and understand the significance of evolving stories and their interrelationships.

Common temporal visualization techniques usually aggregate news articles in defined time intervals. With this approach, the analyst must define the aggregation interval in advance: once the system has aggregated the data and displayed its visualization, the user can no longer access each news article directly. These limitations restrict the scope of exploration.

CloudLines[1] addresses these restrictions through time-interval zooming. The user can monitor the evolution of several news stories simultaneously in a dynamic display and use zooming to access each news article within a broad time window. CloudLines shows each news story, or document, as a circle on a time axis, placed according to the document's online publishing date. Documents are grouped according to temporal density, which CloudLines calculates by estimating kernel density for one-dimensional data.[1] Essentially, each circle's radius is the estimated temporal density for that document.

Figure 1 shows a CloudLines visualization of news data collected during one month using EMM, which typically aggregates news articles from more than 10,000 news websites in 60 languages. Blips signify potentially noteworthy events. In the first row, for example, which shows the results of monitoring "Hosni Mubarak" occurrences, the blips correspond to the crisis in Egypt. Muammar Gaddafi (second row) starts appearing in the news in the second half of February, which corresponds to the unrest in Libya. Barack Obama (third row), on the other hand, is steadily present in the media over the entire month.

Because the visualization can display longer periods than just a day, a month timeline can become extremely dense and obscure patterns that happen within a day or over several days. To see these patterns and access individual news items, users can apply a lens tool to zoom in on part of any interval while maintaining context within a wider time window. The user selects the width of the magnified area, as well as the magnification level according to the interval of interest.

Figure 2 shows four possible lens positions (green boxes) of the Barack Obama row, which the user has elected to explore in depth. Lens position 2, for example, shows delays between news reports, while position 3, the densest part of the line, reveals another short break between the news reports. Position 4 shows that several news articles appeared almost at the same time in the middle of the lens—a potentially interesting

Figure 2. Magnifying the Barack Obama line. By positioning a lens tool on any part of the Barack Obama time interval, the analyst can see daily patterns, such as the gap in news coverage in position 2 or the convergence of coverage in position 3, and access individual news items. These potentially interesting patterns are impossible to see without magnification, as the no-lens line (top) illustrates.

news event. Visualization without a lens shows a homogeneous line with no particularly interesting patterns.

DETECTING AND EXPLORING CRITICAL EVENTS

Automated and visual analyses have valuable intersections, which are often detectable through topics.[2] The "Visual Exploration of Topic Development" sidebar presents some methods that we and others are using to detect events of interest. One aspect of our work in this area is the development of a visual analytics system to locate target words—a system we applied to analyze nearly 87,000 customer comments that a company received between September 2007 and February 2010.

As with CloudLines, the goal is to monitor frequency behavior—but this time the frequency of target words instead of names. Again, an unexpected frequency increase could point to an issue, so the analyst's starting point is to detect an increase that might be worthy of detailed inspection. Automatic detection of potentially interesting target words is problematic because it is impossible to predict with certainty what events might suddenly become important and therefore to estimate how word relevance might change. Our system resolves this dilemma by using a combination of word tagging and temporal-density analysis.

Analysis

To acquire candidate target words, we use a common part-of-speech tagger to identify a particular noun or compound noun and then monitor the word's diachronic behavior. The word's absolute frequency within a burst matters less than the frequency's relation to what is expected. That is, a word can occur far less often relative to words that a user expects to see frequently, yet that word can still have a relatively strong burst. In fact, target words that usually rarely occur can lead to interesting discoveries, such as process flaws in shipping or trends in customer satisfaction.

Tagging and monitoring can quickly yield hundreds of thousands of target words—far more than an analyst can visually scan to identify frequencies with potentially interesting bursts. To overcome this problem, we designed an automatic detection algorithm that points the analyst to bursts that might be worth investigating to identify significant issues. To qualify as potentially interesting, the burst must have documents that have a predominance of negative sentiments and highly similar content and that occur unexpectedly close together in time. The automatic detection algorithm looks for these characteristics and provides the analyst with a ranking of proposed selections for exploration using visualization.

Visualization

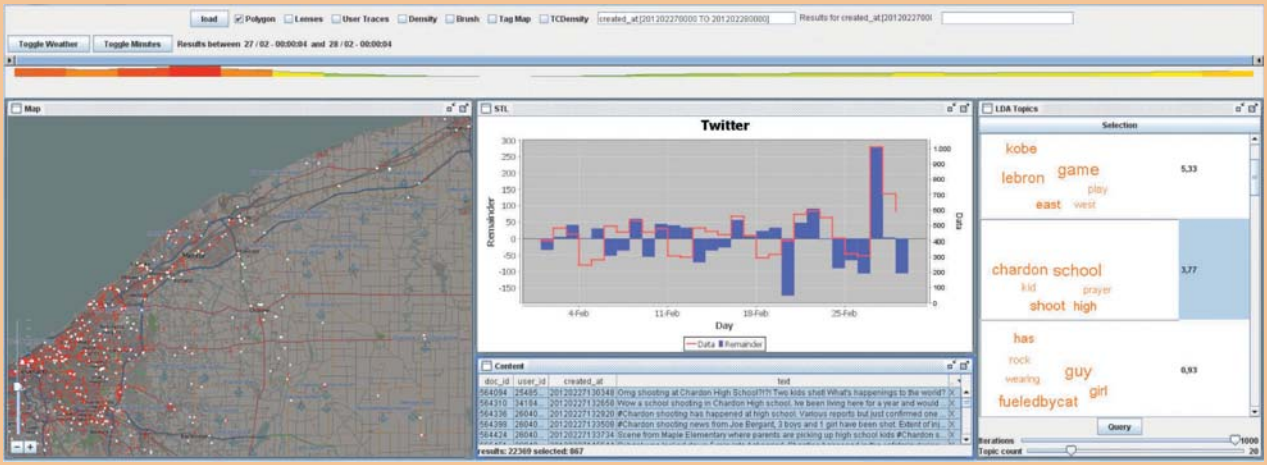
Figure 3 shows the visualization of a burst for the target word "customs." The automatic detection algorithm has pointed the analyst to a period during which customers experienced delays in receiving their orders. Further investigation reveals that the packages had been held up in customs because of missing or incorrect paperwork.

The system begins visualization by scanning the timelines of all target words to search for time-dense intervals that contain multiple documents with mostly negative sentiments and similar content. It then automatically highlights detected events and extracts words that are strongly associated with the relevant documents. To create the density function curve (series of shapes atop the bars), it orders the documents according to their time stamp in the document stream

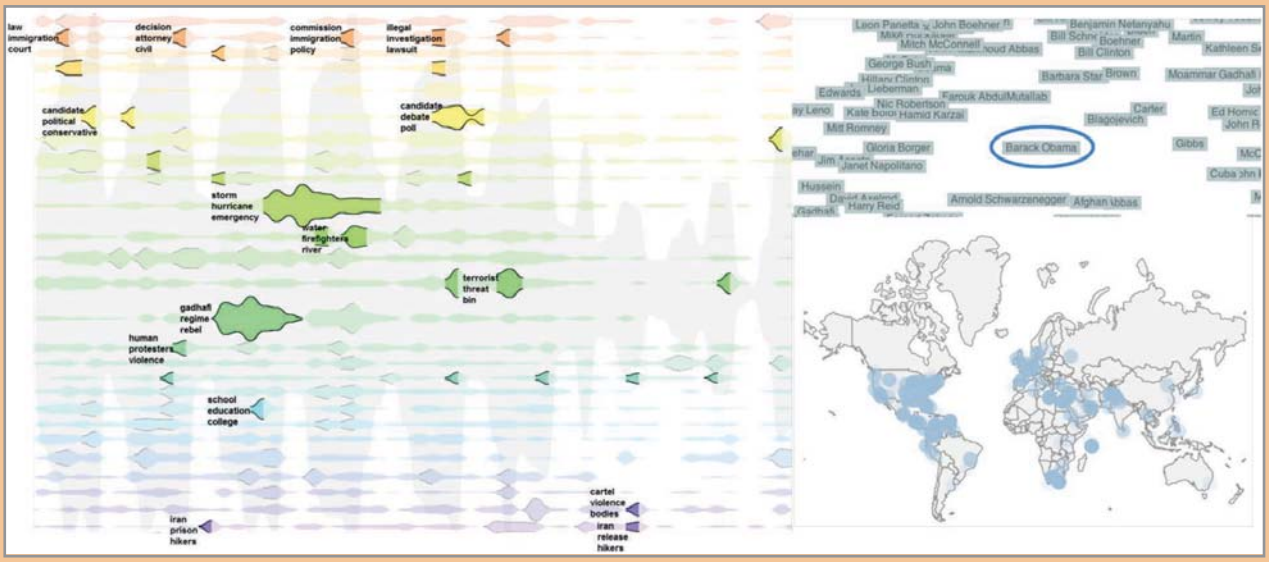
Visual Exploration of Topic Development

Much research in real-time analysis of text streams has focused on analyzing topic development along the temporal domain. For the most part, the proposed approaches begin with a search for documents with selected topics and then combine search results with additional metadata, such as the documents' geospatial distribution. This combination supports varied and rich event detection because the resulting visualization can offer a look at the topic from many perspectives.

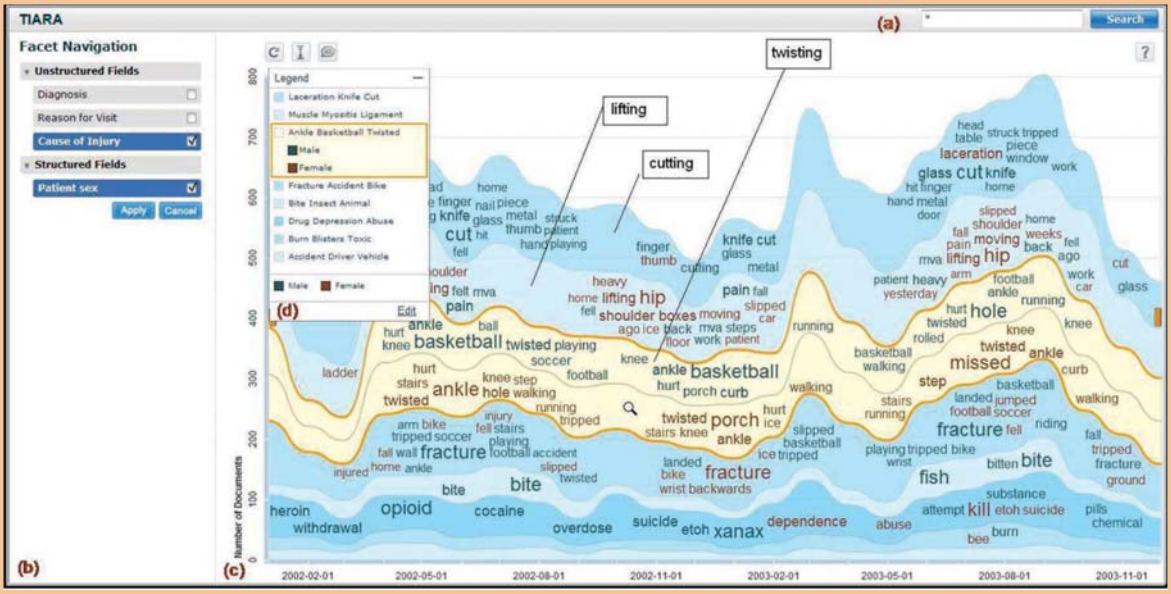
Although it is impractical to cover all of this important work, the following examples give a flavor of recent developments.



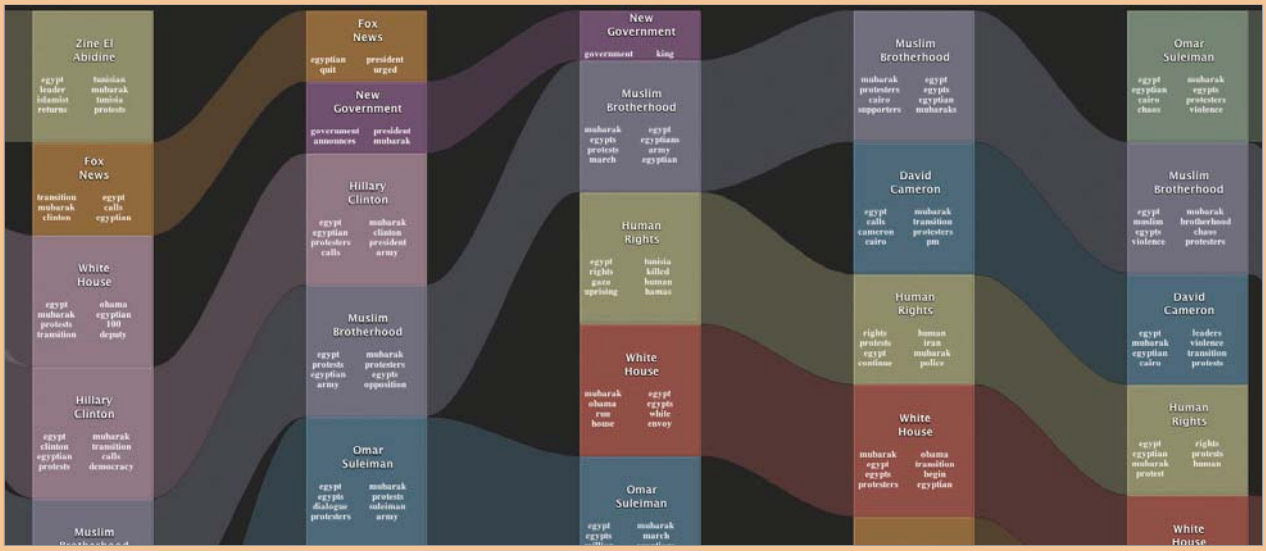
A decomposition of seasonal trends[1] detects interesting events in a stream of georeferenced microblogging text. On the left is a distribution map of Twitter messages related to the Chardon, Ohio, school shooting on 27 February 2012. The system also provides a chart estimating the event's abnormality (middle) and a word cloud of topic contents (right).



LeadLine[2] applies topic modeling to identify events involving topics, people, places, and points in time and displays these in a linked visualization oriented to text, location, and point in time. In this display, the analyst is exploring Barack Obama (oval on right screen). The system responds by showing a map of places related to Obama, as well as a listing of time-ordered events (left). The number of messages appearing in that day determines the height of each shape. The most important keywords appear by a shape.



TIARA[3] presents text related to a selected topic as a stream along a single time axis that appears on the bottom. The system displays distributions over time as a stacked graph, and major keywords appear as a word cloud within each topic. Here, the analyst is investigating trends in sports injuries, so keywords include ankle, basketball, and fracture.



Story Tracker[4] visualizes flows of important topics as color bars, giving similar topics the same color and connecting them with semitransparent curves. The system ranks topics by importance for each day, which is measured by the number of articles in each topic and the similarity of articles within the topic. Topics can change associations as the news data dictates. In this visualization, for example, the topic on Muslim Brotherhood (gray) is gaining more importance and might later merge with other topics related to protests in Egypt. Because Story Tracker supports incremental analysis, it can add data to the system pipeline when it becomes available without reprocessing the entire dataset from scratch. The user can steer text clustering and analyze stories that split and merge over time.

References

1. J. Chae, "Spatiotemporal Social Media Analytics for Abnormal Event Detection Using Seasonal-Trend Decomposition," *Proc. IEEE Conf. Visual Analytics Science and Technology* (VAST 12), IEEE, 2012, pp. 143-152.
2. W. Dou, "LeadLine: Interactive Visual Analysis of Text Data through Event Identification and Exploration," *Proc. IEEE Conf. Visual Analytics Science and Technology* (VAST 12), IEEE, 2012, pp. 93-102.
3. S. Lei et al., "Understanding Text Corpora with Multiple Facets," *Proc. IEEE Conf. Visual Analytics Science and Technology* (VAST 10), IEEE, 2010, pp. 99-106.
4. M. Krstajić et al., "Incremental Visual Text Analytics of News Story Development," *Proc. Int'l Soc. Optical Eng.*, vol. 8294, 2012; http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1283762.

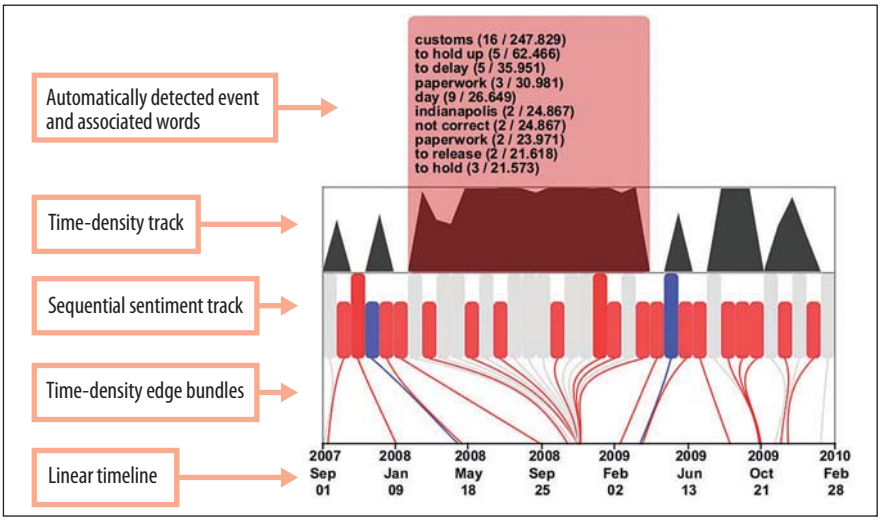


Figure 3. Visualization of 36 documents that mention "customs." The visualization consists of complementary components organized as horizontal layers: the sequential sentiment track (middle) is the visualization's core part, with each vertical bar representing a document that contains the target word. Colors indicate sentiment polarity: negative (red), positive (blue), or neutral (gray)—the higher the bar, the more certain the analysis. The region above the bars depicts a plot of pairwise time-density values. The "cone" shapes are areas of interest.

and then plots pairwise time-density values, displaying them in the time density track. The curve goes above zero whenever two documents are closer in time than expected, creating the cone's base, and reaches maximum when they have exactly the same time stamp, which appears as the cone's point.

The system ranks target words according to how interesting the word's occurrence might be. Infrequent words, such as "customs" and "packing list," are among the top-scoring target words and thus can point the analyst to surprising findings. To determine a document's sentiment context, the system uses a dictionary approach based on sentiment lists that contain negatively and positively connoted words. Different syntactic and distance-based heuristics check if a sentiment word refers to a target word. These referring sentiments then determine the target word's sentiment polarity (negative, positive, or neutral).

Some grammatical constructions are clear indicators for sentiment references and lead to a high certainty, but in many other cases, the analysis can be based only on guessing heuristics. Visually conveying certainty makes analysts aware that a target-based sentiment analysis can be uncertain. In practice, accuracy beyond 80 percent is rare, so it is important to indicate where automatic analysis might have gone wrong.

The sequential sentiment track also gives the analyst access to the full text. In Figure 4, for example, the analyst has moused over "packing list" to reveal a customer

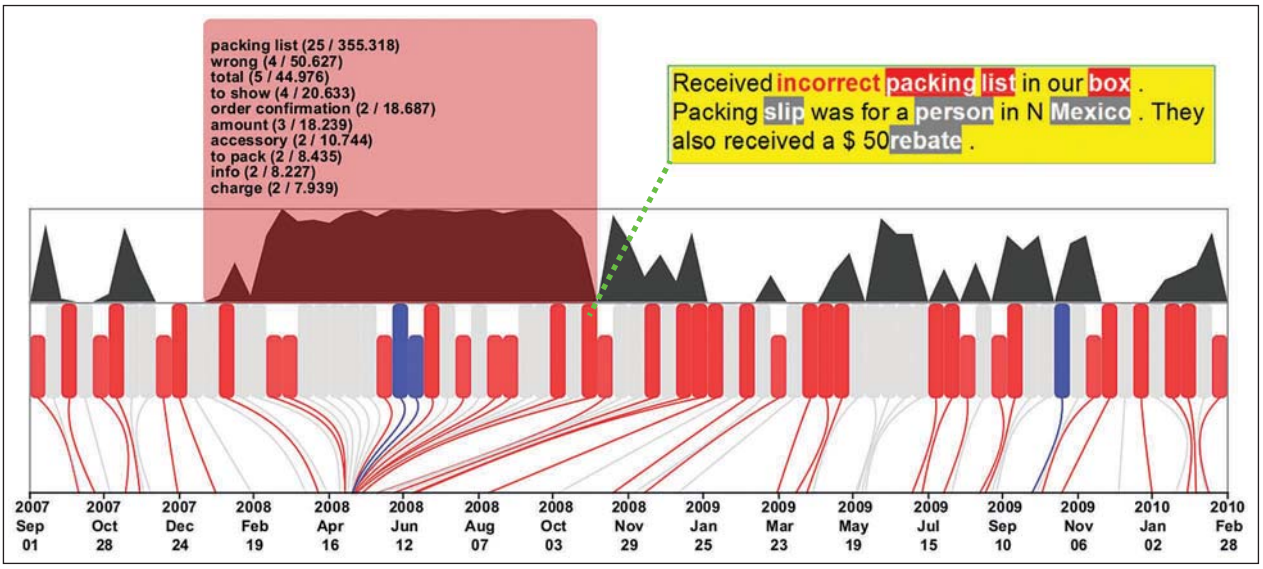


Figure 4. Visualization of 75 documents that mention "packing list." The automatic detection algorithm has pointed to an interesting burst (purple box) along the timeline, which the analyst clicks on to discover that, during a certain period, customers received packages with incorrect packing slips. Either the packing list was meant for a different customer or the charge was wrong.

complaint that the packing list was incorrect. To show distribution over time, the system draws edges from each document bar to its temporal position on a linear timeline. A force-based method bundles the edges wherever the documents are unexpectedly close in time. In this time-density edge-bundling method, two documents have a high time density if their time gap is smaller than the average time gap minus the standard deviation of the average time gap. The average time gap depends on the target word, and therefore the method scales and works in the same way for target words of different overall frequencies.

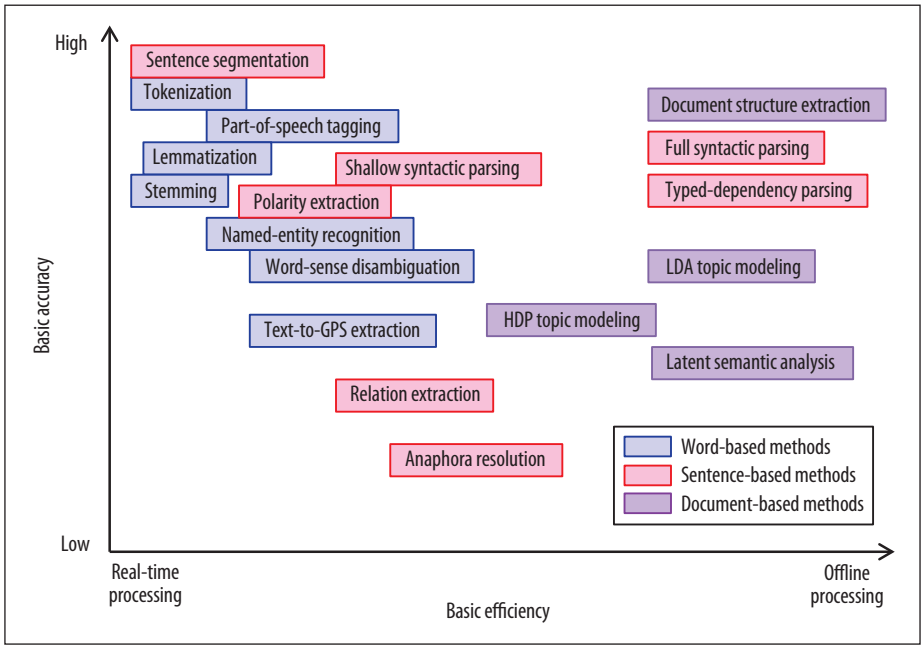


Figure 5. Taxonomy of some popular text-processing models for data analysis. The taxonomy roughly characterizes word- (blue), sentence- (pink), and document-based (purple) methods according to average accuracy and efficiency.

RESEARCH CHALLENGES

Our work and our survey of existing research efforts revealed the need for additional research in all three components of a typical visual analytics system: data management, data analysis, and visual interactive display. The most critical research requirements center on the incompatibility of offline visual analysis systems and dynamic analysis,[3,4] but implementing real-time analysis systems also has many open problems.[5]

Data management

Database systems and communication networks form the back-end infrastructure of many analytical applications. Relational database systems often do not scale well to streaming data because of their need to handle both high-bandwidth data insertions and expensive analytical queries. Indexing schemes work efficiently for small datasets or for static data, but performance can deteriorate with larger datasets and dynamic data. Hash indexes, for example, can lead to more collisions as the list of tracked topics grows.

Database schemes are typically fixed and designed for offline use. In dynamic analysis, in contrast, the data scheme can change, particularly in streaming text analysis, since the relative importance of topics, events, and characters can change suddenly and often. Such analyses require a dynamic data scheme. Text streams also contain links to nonstandard and multimedia datasets, such as to a videofile in a news document, yet many database systems are tailored to accommodate only one data type.

Data processing

Data analysis methods to detect topics, clusters, or relationships must work efficiently on dynamic data. However, the well-known KDD (knowledge discovery from database) model assumes that data analysis is done offline. Existing algorithms must be able to scale to dynamic data, gracefully trading off execution speed and results quality.

In addition, not all data-mining algorithms work incrementally. Many data analysis methods require specified parameters and pattern descriptions, but in dynamic data, pattern types shift, requiring the pattern-search parameters and even the algorithm to change as well.

Displaying dynamic data

Because most visualization is designed for static data, implementing dynamic visualization (for example, through animation) is a difficult problem. Dynamic visualization can distract users and hamper visual short-term memory. Any visualization must account for change blindness, so it must have a way to put new textual input into an established context with previously processed text content so that analysts can track developments in the text message stream.

As new data arrives, the system must constantly update the display, but it is difficult to determine the best rate that will keep analysts current without overloading them. The tradeoff is between making changes visible and keeping the display as constant as possible. Focus-and-context displays can relate a particular data interval to the overall dataset according to data selections, but it is less obvious how to keep focus and context over time. The system can display

only a limited context from the past because the amount of data is constantly growing, and the data load is changing in response. Any method must be able to tolerate these fluctuations.

Analytic processing

To support sense-making, the system must be able to derive a high-level semantic structure from the text stream and generate its visualization. The structural units can be thematic clusters, stories, topics, events, and so on. Over time, such units typically grow or shrink, overlap, split, merge, and appear or disappear, making them hard to display incrementally.

To update the display, trigger-based approaches rely on either automatic stream-driven updates or user requests to generate an update to enable deeper exploration. How to intelligently combine automatic and interactive triggers is a formidable research problem. Ideally, the system would monitor the analyst and adjust the display's detail level and temporal scope to best fit the current analytical stage and the analyst's cognitive load. Progress to address this updating issue will require advanced, automatically derived definitions of "potentially interesting" and evaluation functions to measure the usefulness of additional information in a given analytical context.

MODEL EFFICIENCY AND ACCURACY

Clearly, automatic text processing models are critical to the success of real-time visual text analysis, so those designing text visualization systems must take care to choose the right model. To assist in model selection, we created the broad methods accuracy and efficiency taxonomy in Figure 5. Our categorization is based on an extensive methods survey,[6] not on formal, exhaustive experiments, but it offers at least rudimentary guidance in selecting an applicable model.

Each model has associated methods[7,8] that trade off execution speed and results quality. To keep our taxonomy simple, we omitted the interdependence between some of the models. For example, full syntactic parsing can help improve anaphora resolution (resolving cross-references within a text) but at the cost of an additional delay.

Although the exact model position in this space might be arguable, the taxonomy reveals three general tendencies:

- most word-based methods scale quite well to real time,
- the accuracy and efficiency of sentence-based methods varies considerably, and
- most document-based methods apply only to offline processing.

The third tendency is logical; documents often have a long processing time, or they are not compatible with incremental processing. For example, to achieve good results, latent Dirichlet allocation (LDA)[9] needs to process the dataset as a whole. Dynamic hierarchical Dirichlet processes (HDPs)[10] partly overcome this problem, but cannot fully prevent topic accuracy from degenerating over time. Online algorithms for data other than documents have generated satisfactory results, relative to their offline counterparts.

Despite recent progress, real-time text analytics systems remain largely limited to shallow text analysis. Complex sense-making processes require a modular text analysis architecture, which must rely on a combination of automated methods and human interaction. To provide analysts with deeper insights, researchers must find a way to combine extracted low-level text features with thoroughly interactive visualization.

Recent research in visual analytics is evidence that new techniques, methods, and systems can help deal with the textual information overload in different application domains. For example, real-time automatic event detection in a Twitter stream[11] is already feasible and requires only limited storage because the system can discard old data as soon as that data no longer contributes to events.

New methods integrate scalable interactive visualization with automatic event detection in an environment that combines text and time. Real-time-capable visualization and processing algorithms will help domain experts and novices alike gain deeper insights from heterogeneous text streams, but realizing this goal requires tackling a broad range of research challenges.

References

1. M. Krstajić, E. Bertini, and D.A. Keim, "CloudLines: Compact Display of Event Episodes in Multiple Time-Series," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 12, 2011, pp. 2432-2439.
2. C. Rohrdantz et al., "Feature-Based Visual Sentiment Analysis of Text Document Streams," *Proc. ACM Trans. Intelligent Systems and Technology*, vol. 3, no. 2, 2012, article 26:1-26:25.
3. D. Keim et al., *Mastering the Information Age—Solving Problems with Visual Analytics,* Eurographics Assoc., 2010.
4. J.J. Thomas and K.A. Cook, *Illuminating the Path: The Research and Development Agenda for Visual Analytics*, National Visualization and Analytics Ctr., 2005.
5. C. Rohrdantz et al., "Real-Time Visualization of Streaming Text Data: Tasks and Challenges," *Proc. 1st IEEE Workshop Interactive Visual Text Analytics*, IEEE, 2011; http://vialab.science.uoit.ca/textvis2011/.
6. D. Oelke et al., "Natural Language Processing for Text Visualization," IEEE VisWeek tutorial, IEEE, 2012; http://ieeevis.org/year/2012/tutorial/visweek/natural-language-processing-text-visualization.
7. C.D. Manning and H. Schuetze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

8. S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, O'Reilly, 2009.
9. D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research*, vol. 3, 2003, pp. 993-1022.
10. A. Ahmed and E.P. Xing, "Timeline: A Dynamic Hierarchical Dirichlet Process Model for Recovering Birth/Death and Evolution of Topics in Text Stream," *Proc. 26th Conf. Uncertainty in Artificial Intelligence* (UAI 10), AVAI Press, 2010; www.cs.cmu.edu/~epxing/papers/2010/Ahmed_Xing_UAI10.pdf.
11. M. Krstajić et al., "Getting There First: Real-Time Detection of Real-World Incidents on Twitter," *Proc. 2nd IEEE Workshop Interactive Visual Text Analytics*, IEEE, 2012, http://bib.dbvis.de/uploadedFiles/submission_webpage.pdf.

Daniel A. Keim *is a full professor in the Department of Computer Science at the University of Konstanz and chair of the university's Visualization and Data Analysis Group. His research interests include visual analytics, information visualization, and data mining. Keim received a PhD in computer science from the University of Munich, Germany. He is a member of the IEEE Computer Society and a coordinator of the German strategic research initiative on scalable visual analytics. Contact him at keim@uni-konstanz.de.*

Miloš Krstajić *is a PhD student in computer science and part of the Visualization and Data Analysis Group at the University of Konstanz, Germany. His research interests include visual analytics of text streams, time-series analysis, and text mining. Krstajić received an MSEE from the University of Belgrade, Serbia. Contact him at milos.krstajic@uni-konstanz.de.*

Christian Rohrdantz *is a PhD student in computer science and part of the Visualization and Data Analysis Group at the University of Konstanz, Germany. His research interests include the visual analysis of cross-linguistic data, text time series, and real-time text streams. Rohrdantz received an MS in information engineering from the University of Konstanz. Contact him at christian.rohrdantz@uni-konstanz.de.*

Tobias Schreck *is an assistant professor of visual analytics in the Department of Computer and Information Science at the University of Konstanz, Germany. His research interests include visual search and analysis in time-oriented, high-dimensional, and 3D object data, with applications in data analysis and multimedia retrieval. Schreck received a PhD in computer science from the University of Konstanz. He is a member of IEEE. Contact him at tobias.schreck@uni-konstanz.de.*