Visual Analysis of News Streams with Article Threads

Miloš Krstajić
University of Konstanz
Germany
milos.krstajic@uni-
konstanz.de

Enrico Bertini
University of Konstanz
Germany
enrico.bertini@uni-
konstanz.de

Florian Mansmann
University of Konstanz
Germany
florian.mansmann@uni-
konstanz.de

Daniel A. Keim
University of Konstanz
Germany
daniel.keim@uni-
konstanz.de

ABSTRACT

The analysis of large quantities of news is an emerging area in the field of data analysis and visualization. International agencies collect thousands of news every day from a large number of sources and making sense of them is becoming increasingly complex due to the rate of the incoming news, as well as the inherent complexity of analyzing large quantities of evolving text corpora. Current visual techniques that deal with temporal evolution of such complex datasets, together with research efforts in related domains like text mining and topic detection and tracking, represent early attempts to understand, gain insight and make sense of these data. Despite these initial propositions, there is still a lack of techniques dealing directly with the problem of visualizing news streams in a "on-line" fashion, that is, in a way that the evolution of news can be monitored in real-time by the operator. In this paper we propose a purely visual technique that permits to see the evolution of news in real-time. The technique permits to show the stream of news as they enter into the system as well as a series of important threads which are computed on the fly. By merging single articles into threads, the technique permits to offload the visualization and retain only the most relevant information. The proposed technique is applied to the visualization of news streams generated by a news aggregation system that monitors over 4000 sites from 1600 key news portals world-wide and retrieves over 80000 reports per day in 43 languages.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; H.2.8 [**Database Applications**]: Data mining

Keywords

Visual Analytics, News Analysis, Data Streaming

1. INTRODUCTION

Currently, we are in the middle of the shift from static towards dynamic information analysis. In the past, data was stored in databases and analyses were carried out in regular intervals. Nowadays, more and more people and businesses demand real-time analysis for their particular fields since fast reaction on upcoming topics translates into the competitive advantages in business applications or a higher "fun factor" for non-commercial use, such as social networks.

While new technologies like RSS feeds or Twitter channels tackle the problem from infrastructural point of view, little work has been done on the representation of data streams. News streams, in particular, have their very own requirements for the visual interfaces, which result in a number of challenges for our work. First, a visual *representation* of a news article is not easy. When dealing with thousands of articles, articles cannot be shown in full detail, and in the extreme case not even the heading can be provided. Therefore, smart ways to extract the most relevant information are sought after.

Second, the usual *timeline* approach does not work, because the old news, which do not remain important, quickly lose their value and are replaced by updated or completely different information. Keeping all historic news in the display is not feasible for both space and performance reasons, therefore an appropriate abstraction of groups of news has to be found. Therefore, it is necessary to make a distinction between the old news that continuously keep the attention of the media and the latest news on one side, and the old news that are not relevant anymore on the other side.

Third, in many cases, a number of articles from various sources, report on the same topic, such as a particular business, political or sports event. While it might be interesting for an expert reader to get to know everything about such an event, there is a lot of redundancy of the news articles covering that topic. These *interrelationships* between news articles translate into the need to reduce redundancy by aggregating articles pertaining to similar topics.

Current static visualization solutions that attempt to show temporal evolution of a specific topic (or keyword), such

as ThemeRiver [12], do not easily scale to the case of data streams, where the new topics are created and the old ones disappear. Other approaches, more targeted toward evolving data, like LensRiver or the multi-resolution representations, attempt to solve the streaming problem, yet they do not embed any technique to get rid of irrelevant data.

In this paper, we propose the analysis of a news data stream using article threads, a relevance-based technique that turns the stream of news articles into threads by retaining only the important information over the course of time. The technique is time and space efficient and scales to streaming data scenario. Articles are grouped into threads by employing a similarity function based on the comparison of the most important keywords assigned to each article. Once a thread is born, its life span and evolution depend on a three parameter model that takes into account aging, the number of articles in a thread, and its duration. We demonstrate the effectiveness of the technique by applying it to the analysis of news streams coming from a news aggregation agency.

The rest of the paper is structured as follows. Section 2 presents the related work. Section 3 describes the data stream properties, while Section 4 introduces the streaming visualization techniques and algorithms. Finally, Section 5 concludes the paper with some ideas on future work and open issues.

2. RELATED WORK

Temporal analysis of news is not just a question of visual depiction of news in the time domain, but also a fundamental problem in textual data mining. An issue of considerable interest is analysis of news articles as document streams that arrive continuously over time. Each stream is not just an independent sequence of documents, but it also exhibits braided and episodic character [17]. Moreover, in today's news reporting, the most attention is paid to breaking news about the latest events, which are characterized by a fast growth of the amount of information until a certain peak is reached, and fade of media interest afterwards. A formal approach to model *burst of activity* of topics appearing as document streams is presented in [16]. Furthermore, the propagation of short quotes through news websites and blogs is analyzed in [20]. In [18] we detail our system for processing online news streams, while the focus of this article are the visualization aspects of online news streams.

2.1 News Visualization

Newsmap [24], which uses news aggregated by Google News[1], shows the data visually encoded into a TreeMap visualization, based on the amount of news in each cluster and category to which the cluster belongs to. A major drawback of these news aggregators is that they are dealing only with the latest news, i.e. they provide the data for a specific (current) point in time, there are no possibilities for temporal analysis (or it is limited) and they don't give much semantic information about the events mentioned in the news.

The Europe Media Monitor (EMM) [3] is a multilingual

news aggregator system, which collects news articles from over 2,500 sources in 42 languages. These hand-selected sources include media portals, government websites and commercial news agencies. EMM processes 80,000 - 100,000 articles per day, enriching them with various metadata, such as entities (people and organizations mentioned in the news), news categories and geographical location of the events. Websites, which give access to the data collected and processed by EMM are NewsBrief[2] and NewsExplorer[3]. In our paper, we use the EMM news aggregator as a source of the data stream.

The TextMap website, based on Lydia [21], is an entity search engine, which provides information about different entities (people, places and things) extracted from the news sources.

All of the above mentioned approaches lack, or have limited possibilities for analysis of dynamic change of the information published on-line. Also, possibilities for visual exploration of collections of news articles, which would make better use of the human visual system in detecting trends, patterns and relationships in the news space, are also limited.

2.2 Temporal and Dynamic Visualizations

In the past, a lot has been published about time series. Some examples of such work are *time series bitmaps* [19], pixel visualization techniques for time series [2], multi-resolution techniques for large time series [11], importance-driven layouts for time series [10], or spiral visualization techniques on the basis of clocks [6]. A broader overview of visualization methods for time-oriented data can be found in [1].

One of the first approaches that used visualization to depict temporal evolution of themes within collection of documents is ThemeRiver [12]. In [25], Wise et al. presented the IN-SPIRE visual analytics system, which uses spatial visualization of the large collection of documents for enhanced analysis. LensRiver [9] extends the river metaphor from ThemeRiver into an analytical system for temporal analysis of unstructured text retrieved from video broadcast news. It deals with evolution of themes over time, their hierarchical structure, and employs different visual analytics techniques to perform the analysis. Hetzler et al. [13] proposed to visualize the incremental change in the data by highlighting new (*fresh*) and old (*stale*) documents.

Spatiotemporal data visualization has also received a lot of research attention. The main research challenge hereby is to integrate temporal aspects into two-dimensional map representations. In contrast to abstract visualization techniques, the geographic dimensions thereby introduce additional constraints, which limit the design space for novel approaches and make it a difficult research field. Solutions range from usage of the third spatial dimension for time [15], over pixel placement techniques [4] to the combination of geographic and abstract visualization techniques [14].

---

[1]http://news.google.com/

[2]http://emm.newsbrief.eu/
[3]http://emm.newsexplorer.eu/

Table 1: News Item Properties

startDate	Timestamp of the first news item
endDate	Timestamp of the last item in the thread
duration	endDate - startDate
age	time since the item entered the stream
cat	category of the item
tags	list of news article tags
entities	list of entities appearing in the news
connectedFiles	pointers to similar articles
url	URL of the article

A common approach to cope with time are small multiples. Thereby, several instances of a visualization are rendered, whereas each one represent the data of one particular time interval. How to overcome layout changes in TreeMap visualizations in such a case has been studied in [22].

Yet another visualization techniques are graphs. Rendering dynamic graphs is challenging since layout choices at one point in time have an impact on the layout of future nodes, which are not known at the time of rendering of the former. The study in [7], for example, compares the quality of several offline algorithms when applying them for rendering trees in a streaming context. Furthermore, Beck et al. proposed aesthetic dimensions to heuristically improve layouts of dynamic graphs [5].

While a lot of this work has inspired us, none of these temporal and dynamic visualization techniques were directly applicable to news streams. The next section therefore presents our visualization method for information dynamics.

3. DATA STRUCTURE

We first describe the most important attributes of the data structure that we use in our real world news data stream application. The attributes are shown in Table 1 and are populated from the semantically annotated metadata that arrives in the news stream.

A data object has a dual use: first, it is used as a single news data item retrieved from the stream, with common attributes, such as the timestamp and various metadata - url, language and *tags*. The *tags* attribute is represented by a list of categories to which a specific article belongs to, and they are assigned based on the combinations of trigger words that are found in the article. In this paper, we are using these lists in our algorithm to compare the articles and create new threads. Each trigger word has a rank based on the number of appearances of the word in the article and contributes to the final rank of each tag. We use the tag with the highest rank to assign an article to a specific category.

Table 2: Entity List Attributes

name	id	type
Saad Eddin Ibrahim	177308	p
Hamad bin Jassim bin Jaber Al Thani	78594	p
Human Rights Council	152035	o
Freedom House	72269	o

Table 3: Examples of Tag List Attributes

newsitem_1

tag	rank	score	trigger words
Belgium	1	36	Brussels[2]; Belgium[1]; BRUSSELS[1];
ManMadeDisasters	3	10	accident[1]; train[1]; Trains Collide[1]; collision[1];
Brussels	2	260	Brussels[2]; BRUSSELS[1];

newsitem_2

tag	rank	score	trigger words
Iran	3	10	Iran[1];
Qatar	4	10	Qatar[1];
Palestine	2	10	Palestinian[1];
FreedomSecurity	1	29	freedom[5]; fundamental rights[1]; Freedom[2];
Society	5	33	women's rights[1]; freedom of religion[1]; rights[3]; Human Rights[1];

newsitem_3

tag	rank	score	trigger words
PropertyCrime	4	50	criminal[1];
Earthquake	1	52	Haiti[3];
ElSalvador	2	40	Salvador[2]; San Salvador[1]; El Salvador[1];
Haiti	1	29	Haiti[3]; Port-au-Prince[1];
FundamentalRights	1	41	trafficking[5]; immigration[1];

An example of the *tags* list is shown in Table 3.

Additional *entity* metadata is available in the data stream and can be used in future for finer comparison and aggregation of articles. Entities represent people and organizations mentioned in the news and the details about the recognition process can be found in [23]. An example of entity metadata found in the news is shown in Table 2. More details about the specific news data item attributes and the streaming system can be found in [18].

Second, a data object is used in event-based analysis of our stream as the thread originator, through event-specific attributes *endDate*, *duration* and *connectedFiles*.

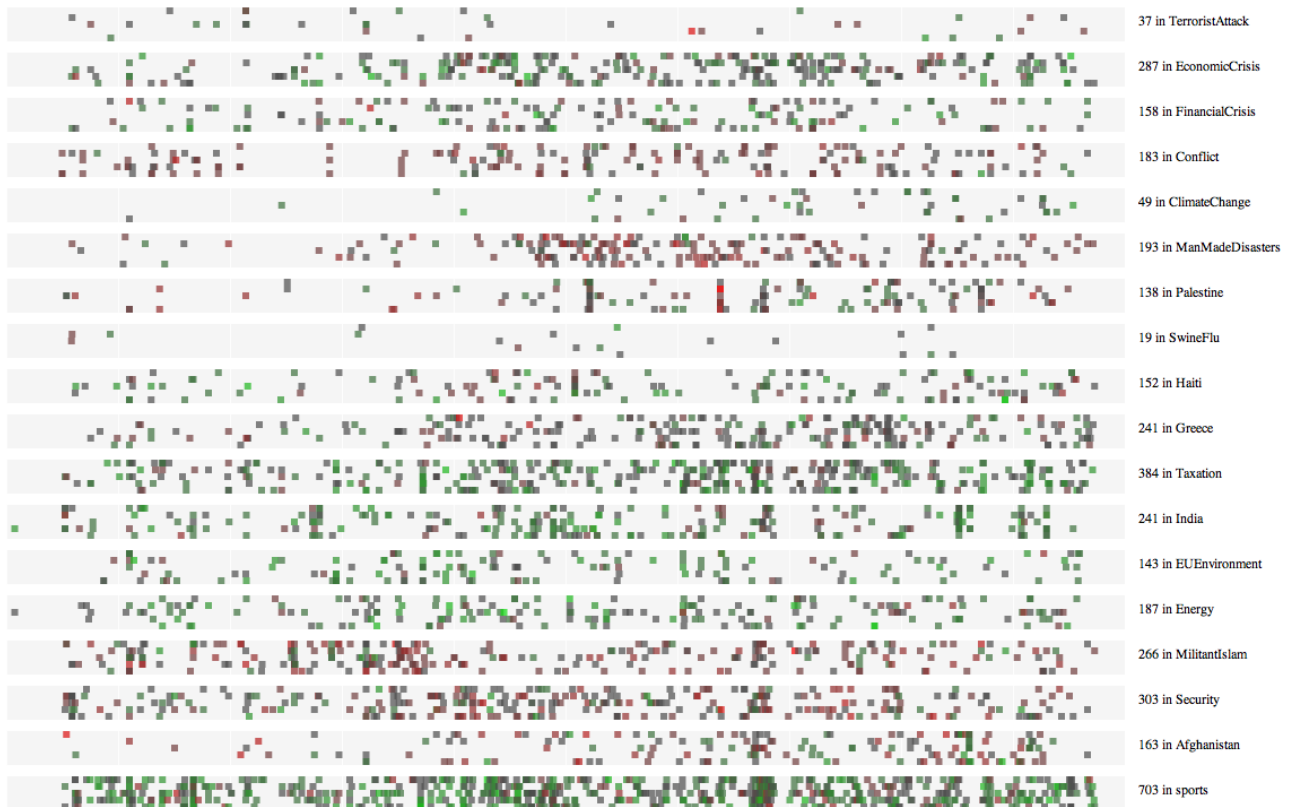Sun Feb 14 23:30:00 GMT+0100 2010   Tue Feb 16 01:30:00 GMT+0100 2010

- 37 in TerroristAttack
- 287 in EconomicCrisis
- 158 in FinancialCrisis
- 183 in Conflict
- 49 in ClimateChange
- 193 in ManMadeDisasters
- 138 in Palestine
- 19 in SwineFlu
- 152 in Haiti
- 241 in Greece
- 384 in Taxation
- 241 in India
- 143 in EUEnvironment
- 187 in Energy
- 266 in MilitantIslam
- 303 in Security
- 163 in Afghanistan
- 703 in sports

Figure 1: News stream monitoring. The dates in top left and top right corner show the beginning and the end of the monitoring period, respectively. The articles are organized in the categories, and the number of articles in each category and the name of the category is shown on the right. News for 18 different categories are shown. Color is mapped to the tonality value of the news article, where saturated green represents high positive tonality score and saturated red represents very negative tonality values. News items in grey are neutral.

4. STREAMING VISUALIZATION

In visual analytics and information visualization, research of streaming data is still at its beginning. Usual approaches employ the concept of deletion of visualized items after a certain period of time, to avoid problems caused by clutter and overplotting. In this paper we're dealing with time-series visualization of streaming data, which presents the first constraint in our design. Many visualizations use the x-axis to represent the temporal dimension of the analyzed data sets. Since most of our potential users are familiar with this mapping, we were very careful of not changing this underlying principle. In the first part of this section, we demonstrate our approach to visualize streamed data items as soon as they arrive and show the advantages and disadvantages of the approach. The visualization of data without aggregation or clustering is meaningful in monitoring tasks in which:

1. the relationships between the items are not of great importance

2. the size of the time interval in which the monitoring is performed can remain constant.

In the second part, we aggregate data items into threads to solve two problems: first, we introduce data-dependent and time-dependent removal, which makes a distinction between the relevant and irrelevant data items. Second, we improve the performance of the streaming visualization when tasks require adaptive change of the time interval in which the analysis is performed.

4.1 Naive Approach for Visualizing Streaming Data

A basic solution for visualization of the streaming data in real-time would be to plot the data point on the screen as soon as it arrives and encode the data attributes of interest to different visual features. There are two general requirements for data streaming - a time interval in which the data is considered relevant and the size of the memory pool that contains streamed data objects, and the requirement that is more strict should be applied. Additional approaches, such as the one described in [11], include multi-resolution techniques that use non-linear rescaling to cope with large data sets. In our very first approach, we concentrated on the monitoring of the latest data within specific time and mem-

ory constraints. The snapshot of the data stream was taken during monitoring interval Feb 14, 2010, 23:30 GMT+01:00 to Feb 16, 2010, 01:30 GMT+01:00 and is shown on Figure 1.

4.1.1 Visualization Environment

Real-time data are dynamic, transient, high-volume and temporal. These properties put strong constraints for creating effective visualizations [8]. Abstraction and presentation of data in a meaningful way and their unbounded characteristics in terms of amount and size create many interesting challenges for the researchers. In our first approach, we decided to set the fixed size (height) for each variable (category), and then allow free floating placement of data items along the y-axis within the subspace dedicated to the category the news data item belongs to.

The solution where no aggregation is used provides news stream visualization at the highest resolution, where every news item is displayed as soon as it arrives. In our real world application, we are interested in monitoring predefined set of categories to which news items from the stream could belong to. Each category can be thought of as a *substream*.

Each news item is represented as a 6x6 block of pixels and is located on the x-axis according to its publication date. The *alpha* value is set to 0.5 in order to show overlapping data points. The color of the item is mapped to its *tonality* score, where saturated green and red represent the items with high positive and negative tonality scores, respectively. Grey color is mapped to neutral articles.

The advantage of this very basic approach is that it gives an immediate look at the data items and their properties, such as their timestamps and tonality values. General patterns of the stream can be easily perceived, such as the changes in the amount of data that is streamed in each category and the overall tonality scores across different categories. Therefore, it would be easy to spot sudden bursts of activity within a specific category. For example, a sudden increase in the number of articles can be seen near the middle of the category *Man-made Disasters*. The inspection of the single data objects, which is possible through interaction, reads *url* attribute of the data item and leads to the full text of the article on the publisher's website and it reveals that most of the articles talk about the train crash in Belgium[4]. The items that are shown are added to the memory pool and in a news monitoring scenario, when the memory pool is full, earliest items are removed.

4.2 Visual Streaming of Article Threads

Visualization of data items without aggregation or dynamic clustering provides direct access to exploration of single items, but lacks information about relationships between the items. Also, it has the inability to make a difference between important and unimportant items, except for sudden bursts within a category. However, a sharp increase in the number of news articles that belong to the same category in short

---

[4]http://www.msnbc.msn.com/id/35403017/ns/world_news-europe/

time interval doesn't necessary have to mean that these articles are related to the same event. This change could be also caused by the fact that a specific source published different news articles in intervals that are very close to each other. The aforementioned issue can be easily visible in very generalized categories with a lot of data, such as *sports*.

Second, visualization is non-discriminative, which means that all the items that exist on the screen and in the memory are of equal importance. In many applications, removal of irrelevant items is needed, both for the analytical and performance issues.

To detect relationships between the data items, distinguish between the relevant and irrelevant ones, and optimize the performance of the visualization, we propose a data-dependent and time-dependent solution with *compareItems* (algorithm 1). This user-driven algorithm provides the opportunity to influence the performance and the resolution of the visualization by adjusting 3 different parameters for aggregation.

Algorithm 1 compareItems(a:Item,s:Threads)

```
 1:
 2:  // set ts, d, n
 3:  // set simThreshold
 4:  // simExists:Boolean flag which decides if the new
     thread should be created
 5:
 6:  mainLoop :
 7:  for each e in s do
 8:     timeSpan = a.startDate - e.endDate;
 9:     if timeSpan ≥ ts AND e.duration ≤ d AND
        e.connectedFiles.length < n then
10:        remove b from s;
11:     else if a.cat = e.cat then
12:        sim = compareTags(a,e)
13:        if sim > simThreshold then
14:           simExists = TRUE;
15:           e.alpha += 0.05;
16:           e.connectedFiles.push(a.lnk);
17:           e.width = a.startDate-e.startDate;
18:           e.endDate = a.startDate;
19:           break mainLoop;
20:        end if
21:     end if
22:  end for
23:
24:  // create new thread
25:  if simExists = FALSE then
26:     drawItem(a);
27:     s.push(a);
28:     simExists = FALSE;
29:  end if
30:
```

The algorithm creates and displays recent threads that consist of similar data items, i.e. in case of news stream these are the articles that are reporting on the same event, while keeping the most important (relevant) threads from the past and removing the irrelevant ones. The usual approach is to use *aging*, where each item is removed when its age reaches
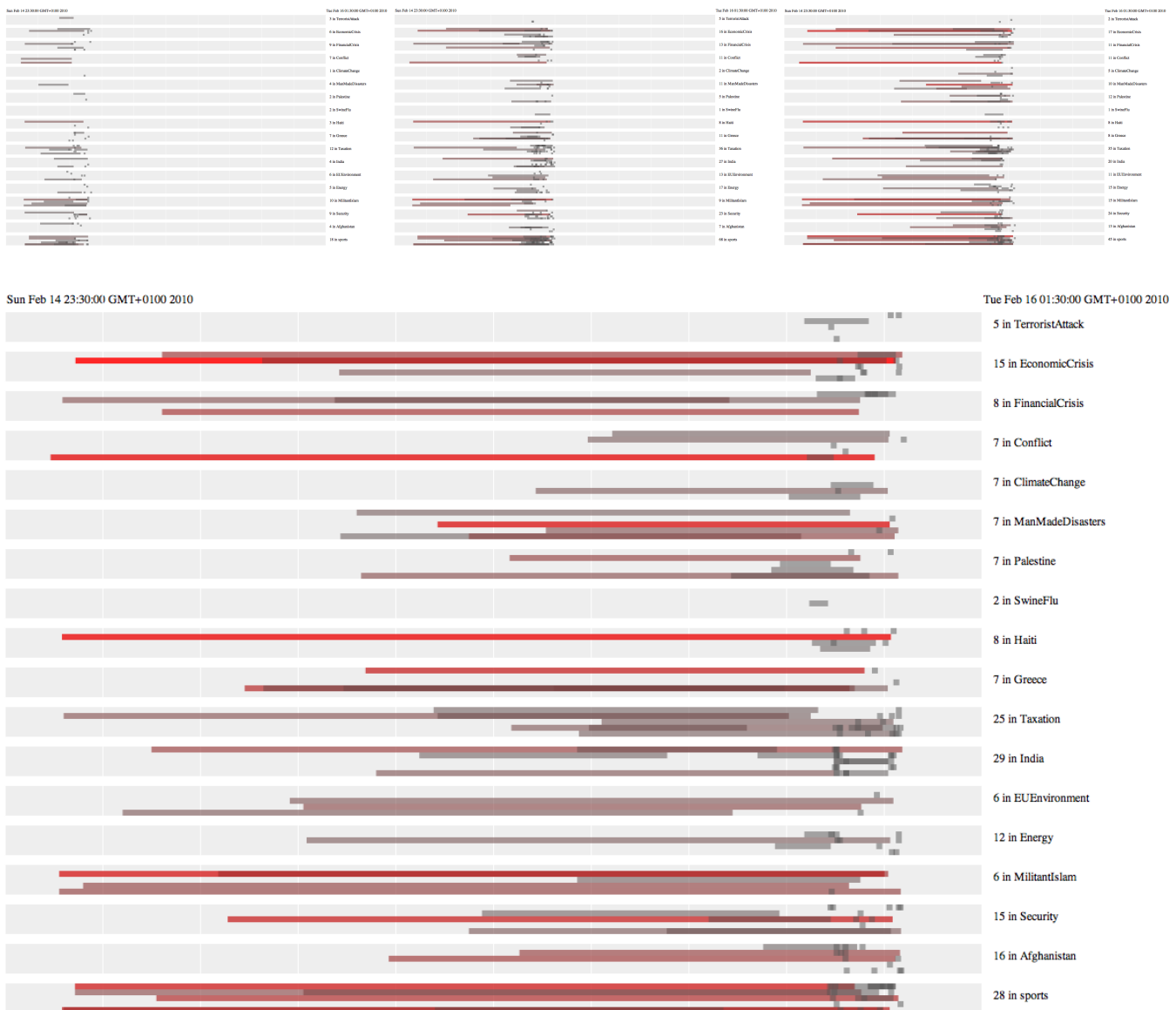
Figure 2: Streaming timeline visualization with aggregation of news articles into threads. Snapshots are taken at 4 different points in time with following algorithm parameters: ts = 240 (min), d = 200 (min) and n = 10. The position and the width of each thread is determined by its *startDate* and *endDate* attributes. The color of each thread is chosen by the number of items in the thread, going from grey to highly saturated red. The snapshots show how the algorithm works, keeping only the relevant threads from the past in the streaming visualization and deleting the unimportant ones. At the same time, small threads that are emerging at present time are maintained.

a specific threshold. In our case, the challenge is that we're not working with data points that are discrete in time, but threads, which are created from discrete data points based on their similarity. Therefore, there should be an obvious difference between two threads, i.e. *relevance*, even if they have the same age. The key idea is to keep relevant events and dismiss the others by taking into account 3 criteria. The thread has to be *old*, *short*, and *sparsely populated* to be deleted. These parameters can be set by the analyst.

4.2.1 The compareItems Algorithm

The algorithm receives two arguments: a, the new item in the stream, and s, the array of threads that have been already created. The algorithm depends on 3 parameters that can be set by the user: the time interval ts, the minimum duration d and the minimum number of items n in a thread e from s.

Thread Removal. The parameter ts is checked against *timeSpan*, which represents the time interval between the

timestamp of the new data item *a.startDate* and the timestamp *e.endDate* of the last item in a thread *e*. If *timeSpan* is greater or equal to this threshold, *e* is considered for removal. We consider them less relevant than the recently updated ones. Basically, the parameter *ts* proposes those threads that didn't acquire any new items in a specific period of time as the candidates for removal.

The other two parameters *d* and *n* are only thread-dependent, but the conditions are checked in conjunction with the first one. That way the removal procedure is called only when the new data arrives and not in specific time intervals. The attribute *duration* of the thread *e*, from the array *s*, is the time interval between the first (*e.startDate*) and last (*e.endDate*) data item in the thread. If *e.duration* is less or equal to the threshold *d* and the number of items in a thread *e.length* is less or equal to the threshold *n*, the thread *e* becomes a candidate for removal. If all three conditions are met, the thread *e* is removed from the visualization and memory. By adjusting these parameters, we can select different relevance measures for the data we're working on.

Adding a new article to a thread. Next, Jaccard similarity coefficient is calculated if the item *a* and the thread *e* are assigned to the same category. If the coefficient is above a certain threshold *simThreshold*, *a* is added to *e* and the visualization gets updated. The color of the event is adjusted to reflect the number of items in it, and its length is extended to include *a.startDate* as the new *e.endDate*. Note that only the pointer to data item *a* and not the whole data object is added to the thread *e* in order to maintain performance of the streaming visualization. At this resolution, we are not interested in any other item detail within a specific thread.

Creating a new thread. Finally, if *a* is not similar to any thread *e* from *s*, *a* is added to *s* as the originator of a new thread.

The algorithm works globally in a way that it removes the irrelevant events from all the categories when the new item arrives, regardless of its category assignment, but it could be easily adjusted to work locally, within a category, if the streaming data would favor this approach.

4.2.2 Visual Encoding

The application of the algorithm on the news data stream is shown in the Figure 2, with subfigures containing snapshots taken at four different points in time. Timeline visualization is employed to show the length of threads, which are organized in 18 different categories of interest (shown on the right). The following parameters are used: *ts = 240* (minutes), *d = 200* (min) and *n = 10*.

Figure 3: Color scale used for connectedFiles.length attribute.

A single thread of articles is represented as a rectangle whose width is determined by thread's *duration* attribute, while the position is determined by *startDate* of the first item in the thread and *endDate*, which is the timestamp of the last item in the thread. Therefore, article threads resemble the representation used in Gantt charts. The total number of threads in each category is shown in the list on the right. The colorscale used to map the number of articles in a thread is shown on figure 3. In the first snapshot, most of the threads are short and grey. As the time passes, it can be seen how only the relevant threads with large number of articles remain on the screen, while the irrelevant items are deleted. By adjusting the parameters of the algorithm, different settings for relevant threads can be achieved.

5. CONCLUSIONS

In this paper, we have presented our work in progress on news streaming visualization framework. We have introduced a relevance-based technique to visualize streaming data for real-time analysis using time and data dependency between the items. The aggregation of items into threads is performed on the fly by the user-driven algorithm which takes into account 3 different parameters to create relevant and remove irrelevant threads. We presented the application of the technique using timeline-based visualization in news monitoring domain. The proposed technique can be also used in non-timeline based visual representations. We have also presented a simple high-resolution streaming visualization that provides details of the stream on the level of a single data item for monitoring the latest events.

The contribution of the approach is twofold: first, on the analytical side, the observer can monitor the incoming streaming data, while maintaining an overview of the important events from the recent past. Second, on the performance side, the size of the data that was processed by the algorithm can be kept small for efficient usage in real-time.

Our future work involves researching better drawing methods to facilitate different analytical tasks in real time and implementation of decay and degree-of-interest functions that would automatically adjust the parameters for creation of the threads. We will work on developing a better solution for ordering of the visual objects and adaptive size of the monitored variables. Finding more sophisticated methods for dynamic clustering of the data together with multi-resolution techniques would provide more semantically rich representation and lead to better understanding of the underlying data.

6. ACKNOWLEDGMENTS

7. REFERENCES
[1] W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60, 2008.
[2] M. Ankerst, D. A. Keim, and H.-P. Kriegel. Circle segments: A technique for visually exploring large

multidimensional data sets. In *Visualization '96, Hot Topic Session, San Francisco, CA*, 1996.

[3] M. Atkinson and E. Van der Goot. Near real time information mining in mulitlingual news. In *WWW '09: Proceedings of the 18th international conference on World Wide Web*, pages 1153–1154. ACM, 2009.

[4] P. Bak, F. Mansmann, H. Janetzko, and D. A. Keim. Spatiotemporal analysis of sensor logs using growth ring maps. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2009.

[5] F. Beck, M. Burch, and S. Diehl. Towards an Aesthetic Dimensions Framework for Dynamic Graph Visualisations. In *Proceedings of the 2009 13th International Conference Information Visualisation*, pages 592–597. IEEE Computer Society, 2009.

[6] E. Bertini, P. Hertzog, and D. Lalanne. SpiralView: towards security policies assessment through visual correlation of network resources with evolution of alarms. In *IEEE Symposium on Visual Analytics Science and Technology, 2007. VAST 2007*, pages 139–146, 2007.

[7] C. Binucci, U. Brandes, G. Di Battista, W. Didimo, M. Gaertler, P. Palladino, M. Patrignani, A. Symvonis, and K. Zweig. Drawing Trees in a Streaming Model. In *17th International Symposium on Graph Drawing, September 22-25, 2009, Chicago*, pages 292–303. Springer, 2010.

[8] G. Chin, M. Singhal, G. Nakamura, V. Gurumoorthi, and N. Freeman-Cadoret. Visual analysis of dynamic data streams. *Information Visualization*, 8(3):212–229, 2009.

[9] M. Ghoniem, D. Luo, J. Yang, and W. Ribarsky. Newslab: Exploratory broadcast news video analysis. In *VAST '07: Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 123–130. IEEE Computer Society, 2007.

[10] M. Hao, D. Keim, U. Dayal, and T. Schreck. Importance-driven visualization layouts for large time series data. In *IEEE Symposium on Information Visualization (InfoVis 2005)*, 2005.

[11] M. Hao, D. Keim, U. Dayal, and T. Schreck. Multi-resolution techniques for visual exploration of large time-series data. In *Eurographics/IEEE-VGTC Symposium on Visualization, 23 - 25 May 2007, Norrkoeping, Sweden*, 2007.

[12] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[13] E. G. Hetzler, V. L. Crow, D. A. Payne, and A. E. Turner. Turning the bucket of text into a pipe. In *INFOVIS '05: Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 12. IEEE Computer Society, 2005.

[14] Y. Ivanov, C. Wren, A. Sorokin, and I.Kaur. Visualizing the history of living spaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1153–1160, 2007.

[15] T. Kapler and W. Wright. Geotime information visualization. In *Proc. IEEE Symp. on Information Visualization*, pages 25–32, 2004.

[16] J. Kleinberg. Bursty and hierarchical structure in streams. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 91–101. ACM, 2002.

[17] J. Kleinberg. *Temporal Dynamics of On-Line Information Streams*. Springer, 2006.

[18] M. Krstajic, F. Mansmann, A. Stoffel, M. Atkinson, and D. Keim. Processing online news streams for large-scale semantic analysis. In *1st International Workshop on Data Engineering meets the Semantic Web*, 2010.

[19] N. Kumar, N. Lolla, E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *SIAM 2005 Data Mining Conference*, pages 531–535. SIAM, 2005.

[20] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM, 2009.

[21] L. Lloyd, D. Kechagias, and S. Skiena. Lydia: A system for large-scale news analysis. In *String Processing and Information Retrieval: 12th International Conference, SPIRE 2005, Buenos Aires, Argentina, November 2-4, 2005: Proceedings*, pages 161–166, 2005.

[22] F. Mansmann, D. A. Keim, S. C. North, B. Rexroad, and D. Sheleheda. Visual Analysis of Network Traffic for Resource Planning, Interactive Monitoring, and Interpretation of Security Threats. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2007)*, 13(6), 2007.

[23] R. Steinberger and B. Pouliquen. Cross-lingual named entity recognition. *Linguisticae Investigationes*, 30(1):135–162, January 2007.

[24] M. Weskamp. Newsmap. *Webdesigning Magazine*, June 2004. http://www.newsmap.jp.

[25] J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information from text documents. *Information Visualization, IEEE Symposium on*, page 51, 1995.