SERRI Report 89990-01

Final Report: Guided Text Search Using Adaptive Visual Analytics

SERRI Project: Smart Search Analytics

Project Principal Investigators:
Chad A. Steed, Christopher Symons, James Senter, and Frank DeNap

This material is based upon work supported by the US Department of Homeland Security under US Department of Energy Interagency Agreement 43WT10301. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the US Department of Homeland Security.

SERRI Report 89990-01

SERRI Project: Smart Search Analytics

FINAL REPORT: GUIDED TEXT SEARCH USING ADAPTIVE VISUAL ANALYTICS

Chad A. Steed, Christopher Symons, James Senter, and Frank DeNap

Oak Ridge National Laboratory

Date Published:

September 2012

Prepared for
US Department of Homeland Security
under US Department of Energy Interagency Agreement 43WT10301

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

ACKNOWLEDGMENTS

The authors would like to express their appreciation for the assistance and support of the Tennessee Fusion Center. Without input from Fusion Center staff, this project would not have been completed successfully.

CONTENTS

FIGURES

SOUTHEAST REGION RESEARCH INITIATIVE

In 2006, the US Department of Homeland Security commissioned UT-Battelle at the Oak Ridge National Laboratory (ORNL) to establish and manage a program to develop regional systems and solutions to address homeland security issues that can have national implications. The project, called the Southeast Region Research Initiative (SERRI), is intended to combine science and technology with validated operational approaches to address regionally unique requirements and suggest regional solutions with potential national implications. As a principal activity, SERRI will sponsor university research directed toward important homeland security problems of regional and national interest.

SERRI's regional approach capitalizes on the inherent power resident in the southeastern United States. The project partners, ORNL, the Y-12 National Security Complex, the Savannah River National Laboratory, and a host of regional research universities and industrial partners, are all tightly linked to the full spectrum of regional and national research universities and organizations, thus providing a gateway to cutting-edge science and technology unmatched by any other homeland security organization.

As part of its mission, SERRI supports technology transfer and implementation of innovations based upon SERRI-sponsored research to ensure research results are transitioned to useful products and services available to homeland security responders and practitioners.

For more information on SERRI, go to the SERRI Web site: www.serri.org.

EXECUTIVE SUMMARY

This research demonstrates the promise of augmenting interactive visualizations with semi-supervised machine learning techniques to improve the discovery of significant associations and insights in the search and analysis of textual information. More specifically, we have developed a system—called Gryffin—that hosts a unique collection of techniques that facilitate individualized investigative search pertaining to an ever-changing set of analytical questions over an indexed collection of open-source documents related to critical national infrastructure. The Gryffin client hosts dynamic displays of the search results via focus+context record listings, temporal timelines, term-frequency views, and multiple coordinate views. Furthermore, as the analyst interacts with the display, the interactions are recorded and used to label the search records. These labeled records are then used to drive semi-supervised machine learning algorithms that re-rank the unlabeled search records such that potentially relevant records are moved to the top of the record listing. Gryffin is described in the context of the daily tasks encountered at the US Department of Homeland Security's Fusion Center, with whom we are collaborating in its development. The resulting system is capable of addressing the analysts' information overload that can be directly attributed to the deluge of information that must be addressed in the search and investigative analysis of textual information.

1. INTRODUCTION

The goal of visual analytics is to turn information overload into opportunity by taking advantage of human flexibility, creativity, and background knowledge as well as the great computational power in modern computers [1]. Such approaches are necessary in search-based, investigative analysis of textual information where the analyst is typically overloaded with too many search results. In most cases, relevant results that reside in the lower portions of a ranked list will be overlooked. To improve the likelihood of finding relevant information, the analyst will benefit from a visual analytics-based approach that highlights significant information and propagates potentially relevant records to the top of the list.

For analysts at the US Department of Homeland Security (DHS) fusion centers, realizing these improvements is key to successfully completing their mission. The DHS fusion centers form a national network with critical links within the state and local area to receive, analyze, gather, and share threat-related information. A key challenge for fusion center analysts is the inability to acquire all relevant data despite its potential availability. Furthermore, the regional information support task is intractable when the analysts are forced to manually sift through too much national, international, or cross-regional information. Of particular concern is the inability to recognize obvious threats, connections, trends, etc. due to either a lack of information or information overload.

Although there are several challenges related to the tasking at the fusion centers, the focus of the current work is to help the analyst cope with information overload. Even in the case of region-specific analysis, the amount of data that might contain relevant information is typically on a scale that makes manual human sorting impossible. A standard search using relevant terms can return hundreds of thousands of potential documents. If the information is not found among the first several results, it will almost inevitably be overlooked. The regional analysts have hundreds of potential data sources to which they regularly turn, and each of them can return large amounts of data. The result is a recurring "needle in the haystack" problem that impedes successful regional analysis.

To address this problem, we have developed a visual analytics system called Gryffin. Gryffin combines inferential visual interfaces to assist the analyst in conducting search-based analysis of textual sources. Gryffin distinguishes itself from the body of related systems described in the literature by its tight integration of semi-supervised machine learning with inferential visual representations. Gryffin captures interactions with the information shown in the visual interface to drive these learning processes that re-rank and highlight records of potential significance. By moving potentially relevant records from the lower portions of the list to the top, the analyst has a much greater chance of discovering the most significant information. In the current work, we have focused on a single source for fusion center analysis, the DHS Daily Open Source Infrastructure Report.[*] These reports are generated each business day as a summary of the information in open-source publications that relate to critical national infrastructure issues.

The remainder of this paper is organized as follows: Following an overview of related work in the search and investigation of textual information in Section 2, Section 3 provides an overview of the Gryffin system and its architecture. Section 4 describes the envisioned workflow with Gryffin. Section 5 introduces the details of the interactive visualizations in Gryffin. Section 6 provides details on the semi-supervised machine learning algorithms used to re-rank the unlabeled search results. Additionally, a practical case study involving real world information is presented in Section 7 to demonstrate the enhanced analysis capabilities in Gryffin, followed by the concluding Section 8.

[*]The DHS infrastructure reports are available online at http://www.dhs.gov/dhs-daily-open-source-infrastructure-report.

2. RELATED WORKS

Gryffin is an evolutionary development that builds upon our successful work with the Piranha [2] and VIPAR [3, 4] text analysis systems. Whereas these earlier systems focused on multi-agent clustering for analysis, Gryffin integrates adaptive learning as the core capability. The Gryffin client interface benefits from recent advances in the realm of visual analytics focused on the search, investigation, and visualization of textual information. The highly successful practice of utilizing multiple coordinated views is often found in the visual analytics literature [5] and is a central feature in Gryffin. Linked views foster more creative analysis because analysts are given an opportunity to view the data in different ways. The technique is enhanced when combined with brushing and focus+context methods [6]. With respect to focus+context, Gryffin utilizes a variation of the visual bracketing approach described by Roberts and Suvanaphen [7]. Acting as a sliding window for exposing varying levels of detail in list-based search records, the bracketing technique is effective in maximizing screen space and is derived largely from distortion-based presentations [8]. Gryffin also relies heavily on the utilization of visually constructed queries [9].

There are a number of important visual analytics systems that inspired features utilized in Gryffin. The Jigsaw system is a good example of an interactive approach that utilizes entity co-occurrence analysis in reports to guide the investigation of textual data [10]. The IN-SPIRE system is a novel approach for understanding textual information that projects documents into a rather unique view that supports the comparison of search queries [11]. PatViz assists users in constructing complex queries visually to assist in the exploration of patent result sets [12]. Like Gryffin, PatViz is designed to ease the burden of forming complex queries by relying on multiple coordinated views. ResultMaps is also designed for enhanced searches, but is focused on hierarchical metadata for digital library search engines [13]. The WebSearchViz tool is designed to visualize Web search results to improve navigation and exploration with a metaphor inspired by the solar system [14]. VizCept [15] was also developed for addressing the challenge of searching textual information. VizCept combines individual workspaces with collaborative visualizations for supporting keyword searches. Here the focus is on a web-based, collaborative environment.

Gryffin combines variations of the techniques introduced in said works with semi-supervised machine learning approaches. Gryffin utilizes practical views rather than complex transformations to facilitate adoption and use in an operational environment. In the spirit of achievement described by Keim et al. [16], the advancement of Gryffin lies in the synergy between automated analytics, visual representation, and intelligent interactions. The authors are not aware of any similar systems that combine visual analytics with local adaptive learning to improve the display based on the interactions of the analysts with search hits. By learning from the user's criteria, Gryffin dynamically improves the chances of finding relevant information for a guided search experience that promises to reduce the knowledge discovery timelines in investigative search processes.

3. GRYFFIN—A TOOL FOR INVESTIGATIVE ANALYSIS OF TEXTUAL INFORMATION

3.1 Overview

The Gryffin client interface (see Fig. 1) offers multiple coordinated views and automated analytics to assist in the investigation of search results from a body of textual information. At the top of the interface, a toolbar is provided which includes a search panel for inputting search terms and a button (with a gear icon) for executing the re-ranking of unlabeled search results. Directly below the toolbar, date information from the search hits records are used to display the frequency of hits over time.

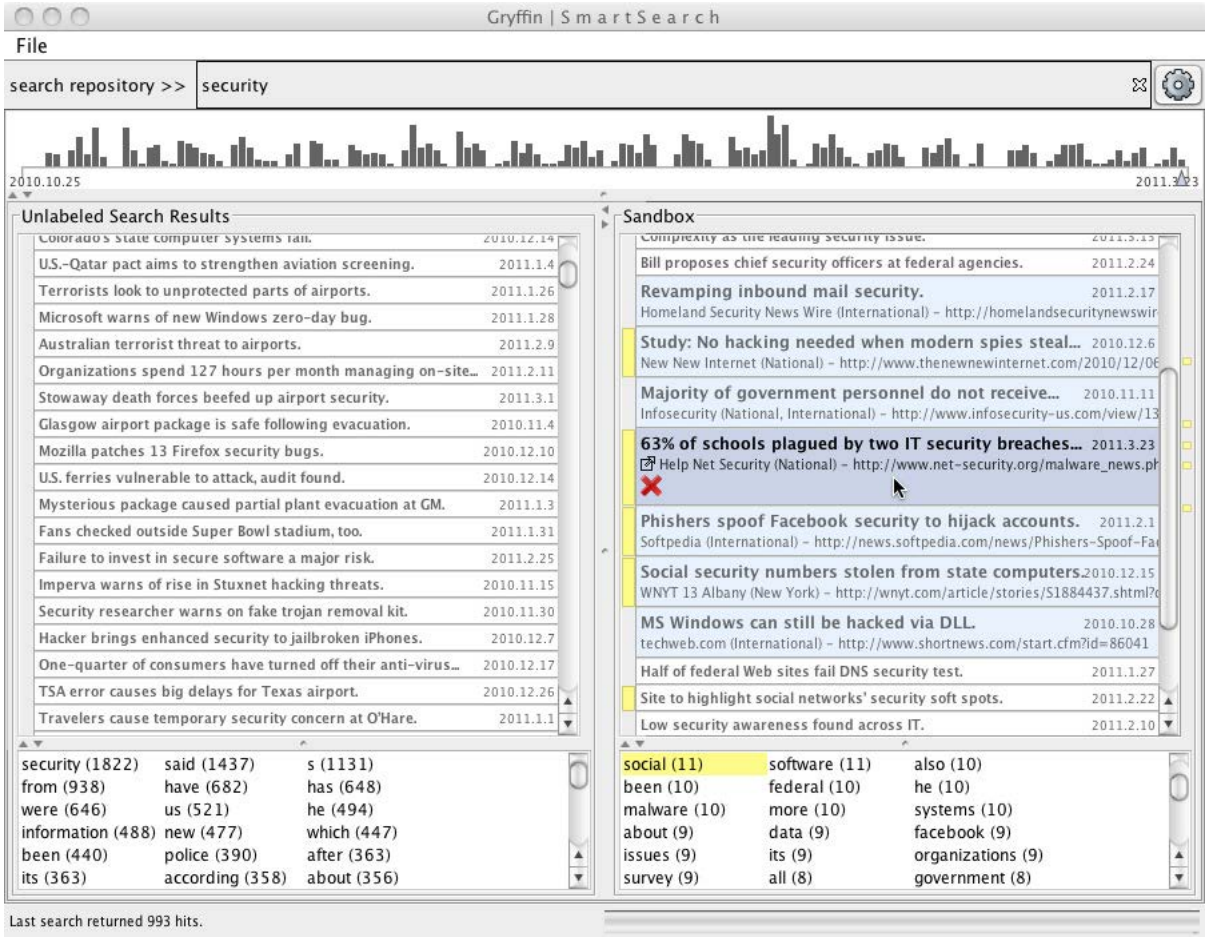


Fig. 1. The Gryffin client interface consists of three main views for investigating search results: a temporal view (top), a list-based textual view (middle), and a term-frequency view (bottom). Below the temporal view, two panels separate unlabeled records from relevant records in the Sandbox. In this figure, the client is shown after searching for the term "security" and selecting the term "social" in the top terms panel.

Beneath the temporal view, two panels are arranged: The Unlabeled Search Results panel and the Sandbox panel. Both of these panels provide a list-based display of the textual information from the search hits as an interactive panel with focus+context capabilities. At the bottom, the panels also provide a term-frequency view of the records currently shown in the list view. The top terms are listed in descending order with the term text and the frequency of occurrence in the listed records. As its name implies, the Unlabeled Search Results panel is populated with search hit records that are not labeled. After a search is executed, the search hits are used to populate this panel. The Sandbox panel is populated as records in the unlabeled panel are marked relevant by the user, providing a cache for managing significant records.

3.2 Architecture

From an architectural perspective, Gryffin is implemented as a client-server system with the configuration shown notionally in Fig. 2. On the server side, the system hosts information fusion, parsing, persistence, and search services. The benefit of locating these services on the server is that we can scale the server component to include more powerful high performance computing capabilities as necessary to address increasing volumes of data. The client component provides a

graphical interface to the server capabilities via an interactive display with integrated semi-supervised machine learning capabilities to re-rank the search results based on user interactions with the client. Both the server and client entities are implemented in the Java programming language.

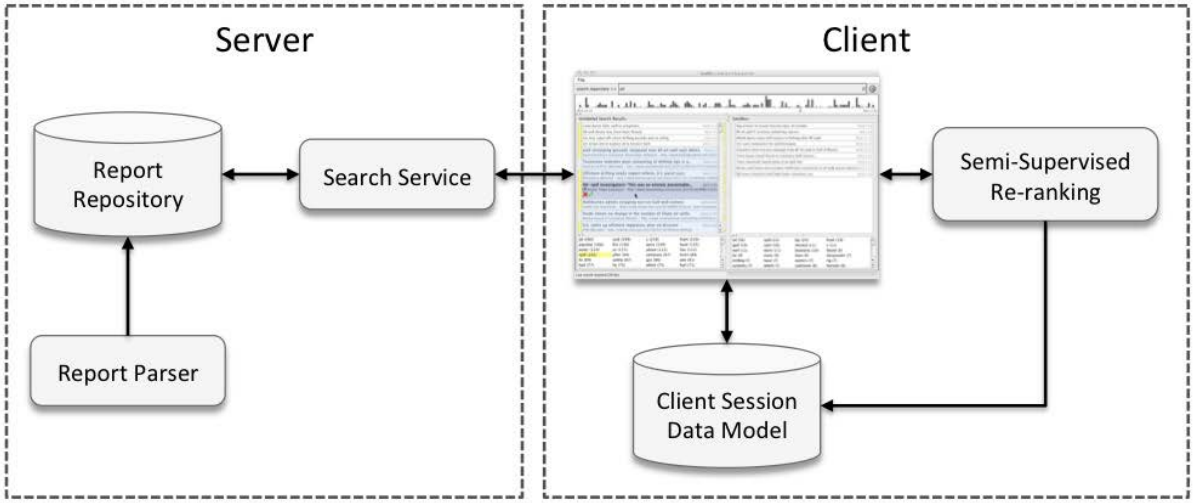


Fig. 2. Gryffin is implemented with a client-server architecture in which the server hosts a data repository that can be searched from the client via a web service. The client maintains a local data model for recording interactions with records and feeding the re-ranking algorithms.

The server maintains a data repository that is populated with the DHS infrastructure reports which are posted daily on the Internet in the Adobe PDF file format. These reports are downloaded and processed by the Apache Tika content analysis toolkit[*] to extract structured text from the PDF report files. The structured text is then parsed to build individual summaries (metadata and text) from the reports. The summaries are then stored in an Apache Lucene index[†] which acts as the Gryffin data repository.

Using the client interface's search panel, the analyst will enter search terms which are transmitted to the server via a web service. The server uses the Lucene indexing and search capabilities to query the repository of report summaries. The search hit records are gathered in a collection on the server and sent to the client for subsequent analysis. The client system contains several innovative visual query capabilities as well as analytics for adaptive learning to assist the analyst in the investigative analysis of the results.

4. GRYFFIN WORKFLOW

As represented in Fig. 3, the Gryffin workflow is an iterative process that is designed to sift through the high volume of information to find the relevant items for a topic of interest. The analyst begins the process by entering a search term in the client interface. This search term is transmitted to the server where it is analyzed and used to extract relevant records from the data repository. A by-product of this process is a set of scores for the hits which are used to order the records.

[*]Apache Tika is a content analysis toolkit that is available at http://tika.apache.org.

[†]Apache Lucene is a text search engine written in Java that is available at http://lucene.apache.org.

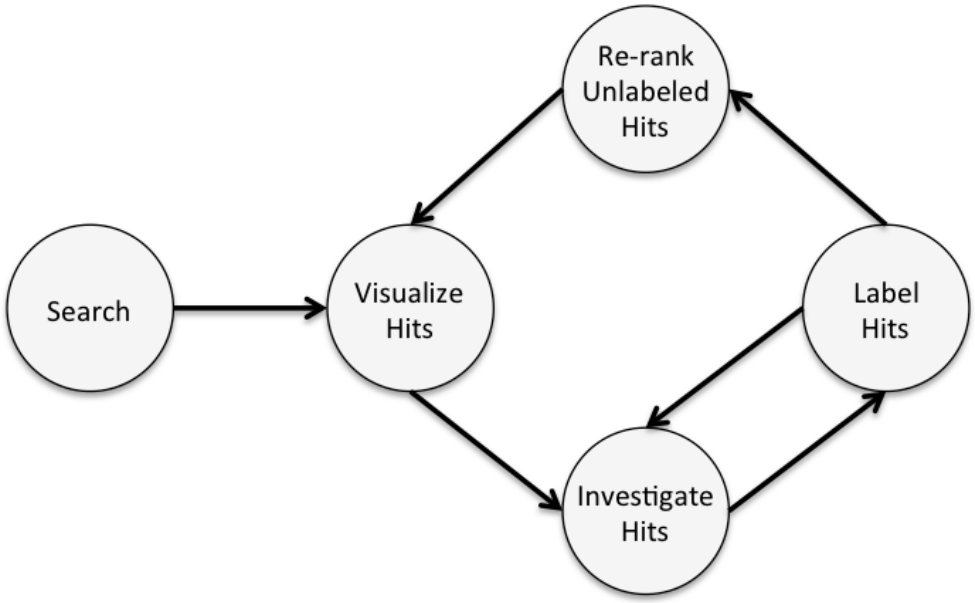


Fig. 3. The typical workflow with Gryffin is an interactive process starting with a search and progressing to a cycle of visualization, investigation, interactive labeling, and re-ranking of the unlabeled records.

A collection of hits is returned to the client for visualization and investigation. At this point, the analyst will benefit from utilizing the various interactive graphics and linked views of the search records to roll up the information into high level summaries and drill down to the details of individual records, as necessary. During this investigation, the analyst will label the records as being either irrelevant or relevant. Irrelevant records are removed from the display while relevant records are moved into the Sandbox view–an area for more focused analysis and collection of significant information.

As the analyst continues with the labeling, he or she will have built a collection of labeled records within the client's local data model. The analyst can execute the semi-supervised machine learning algorithms that re-rank the unlabeled records to essentially force potentially relevant records to propagate up to the top of the list. Without this adaptive learning, the records further down in the list may never be evaluated. The newly ranked unlabeled records are redisplayed and the analyst may continue investigating and labeling the records. The analyst can reexecute the learning algorithms as necessary to account for newly labeled records or perform a new search to build a new group of unlabeled records. When the analyst is finished, the sandbox will contain a collection of relevant records that may be analyzed or condensed into a report mechanism for archival or presentation.

5. USING GRYFFIN TO VISUALIZE TEXTUAL SEARCH RESULTS

The Gryffin client interface features a number of interactive views of the search records. These views are coordinated to facilitate investigation of search records from different perspectives. In the remainder of this section, the details of the temporal view, list view, term-frequency view, and the overview bar are described.

5.1 Temporal View

One important piece of metadata that is parsed from the DHS infrastructure reports is the date field. For each article summary, the date field is parsed and stored with the records in the data repository. In the client interface, these dates are used to display the frequency of hit records for each date in the temporal view at the top of the frame. In Fig. 4, the temporal view is shown after a query on the keyword “oil.”

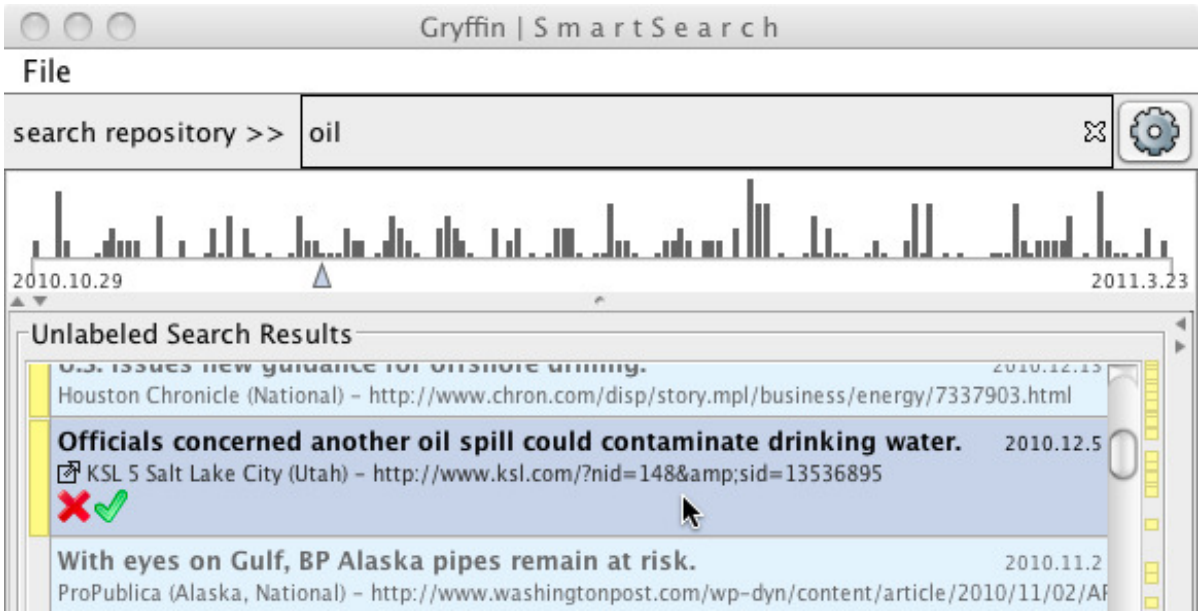


Fig. 4. The date fields from each record are used to build a temporal view of the number of records that occur on a given day. The heights of the bars shown in the timeline indicate the frequency of records. The date of the record currently under the mouse cursor is indicated with a triangle filled with the same color as the highlighted record.

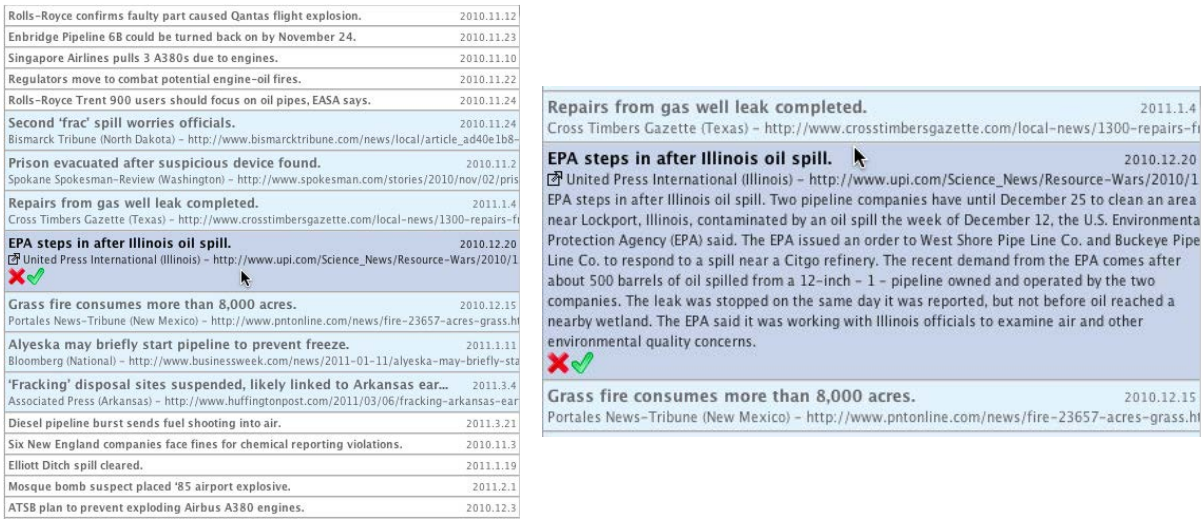
The temporal view uses the x-axis for mapping dates in ascending order from left to right. Bars are drawn for each day that has one or more associated records, where the bar height indicates the count of articles for each date. When the mouse hovers over a record in either of the list views, a blue triangle is shown on the timeline axis to indicate the record’s location on the timeline (see Fig. 4). The color of the indicator triangle is the same color as the background of the record under the mouse cursor to visually link the list and temporal views. The temporal view gives an intuitive overview of the trends for the given terms over time. Spikes and gaps are readily discerned in the display to indicate trends.

5.2 Search Record List View

Below the temporal view, the Unlabeled Search Results and the Sandbox panels are shown. These panels contain a central list view of the search records in a textual form. Given the high number of search hits that are often encountered, the list view utilizes a visual bracketing approach similar to the technique introduced by Roberts and Suvanaphen [7]. This sliding pane is controlled by the proximity of records to the mouse cursor and is designed to maximize the utilization of space thereby accommodating more records into the limited display space.

As shown in Fig. 5a, the location of the mouse cursor is tracked and utilized to show this focus+context view of the records. Records under the mouse cursor have a highlighted background color (a light blue shade), and darker font colors to communicate more prominence to the analyst. The record position on the temporal time axis is also indicated with a triangular marker that is filled with

same blue color as the highlighted record background. The highlighted record displays the title, date, source, location string, and source URL. If the user double-clicks the in the record, the detailed text of the summary is shown below the source string (see Fig. 5b).



(a) Focus+context list view **(b) Detailed record view**

Fig. 5. Gryffin uses a focus+context technique to render records in the list view. The record under the mouse is highlighted while the three records above and below it are in the sliding focus window. In (b), the record has been double-clicked to show the detailed summary text.

The record under the mouse includes a link button below the title string and in front of the source string (see Fig. 5). This button can be clicked to open the referenced URL in a system browser. This capability provides a details-on-demand mechanism to the analyst facilitating the examination of the source web site and the full text that was used to construct the DHS report summary. The summaries captured in the reports are gathered from the referenced websites and typically do not contain the full set of information. For example, images, video, and related articles can be gathered and viewed by following these external links.

Two additional buttons are displayed below the summary string for the records highlighted with the mouse cursor. The X-shaped delete icon can be clicked to remove irrelevant records. Coincidentally, this action will cause the client to internally label the record as irrelevant in its local data model. The delete icon button is available in both the Sandbox and Unlabeled Search Results panels. A check icon button is displayed to the right of the delete icon. When clicked, the check icon will label the record as relevant and move it into the Sandbox panel. The check icon is only displayed for records in the Unlabeled Search Results panel, since items in the Sandbox are known to be relevant.

The three records above and below the currently highlighted record are rendered with a less saturated highlight color and a lighter font color. The font sizes are the same as the highlighted panel, and the title, date, source, location, and URL strings are displayed. These records are rendered as an intermediate, focus effect and no buttons are displayed. Any additional records that are visible in the list view are rendered in the context state, which uses a smaller font and a white background. Only the title and date information are rendered for the context records.

The mechanism of rendering the search records with a sliding focus window provides more space and information for records nearest to the mouse cursor. The shading implements a scheme that mimics the so-called aerial perspective shading that is encountered in artistic landscape paintings and nature. Objects in the distance are shaded with less contrast and smaller elements and objects nearby are more prominent.

The list view provides the analyst with the ability to observe high-level summaries of the textual records and details for highlighted records. The analyst can use the view to drill down to the full set of information for each record by following the link to the referenced source URL.

5.3 Term-Frequency View and List Overview Bar

As shown in Fig. 6, the term-frequency information for all records in the list panels is shown as a list of terms at the bottom of the frame. The top 100 terms are shown as descending rows and columns with the most frequently occurring terms at the top left, the next most frequent term directly to its right, and so on until the least frequent term is displayed as the rightmost term of the last row.

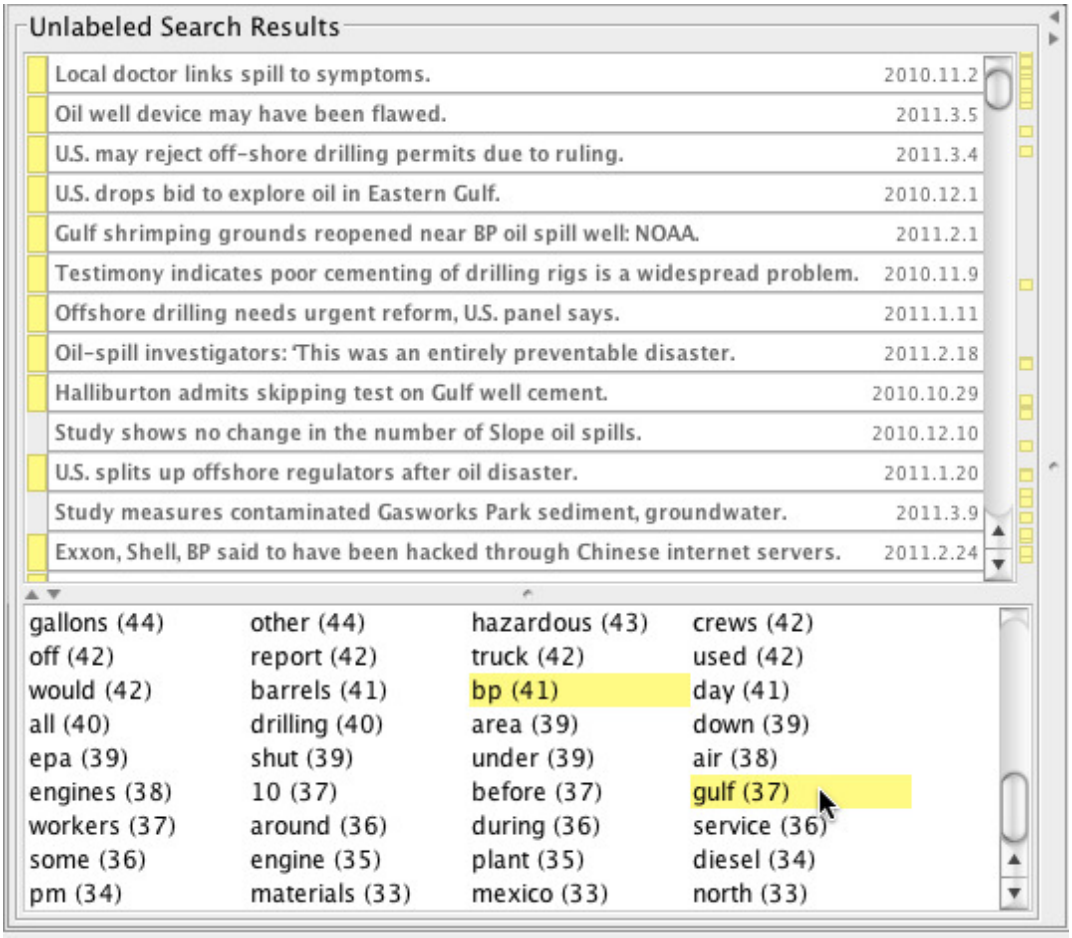


Fig. 6. The term-frequency display (bottom) with linked overview bar (right of the scroll bar) in the record list view panel.

The term-frequency view is interactive. When the analyst clicks on a term, an overview bar is displayed to the right of the scroll bar which shows the location of the records that contain the selected term in the list. In Fig. 6, the locations of records with the selected terms “bp” and “gulf” are shown in the overview bar. The records that contain these terms are indicated in this overview bar with small, yellow boxes. The analyst can click on one of these boxes to cause the record list view to jump to the record. In the list view, the records containing the selected term(s) are also rendered with a small rectangle to visually link the list, overview bar, and top terms views. This ability to select and highlight records using the term-frequency list is an innovative way to conduct investigative analysis. Furthermore, terms can be combined to show multiple collections of records (Boolean AND queries) as demonstrated in Fig. 6.

5.4 The Sandbox View

The Sandbox view (see Fig. 1) is shown to the right of the Unlabeled Search Records panel using a split pane which can be hidden or resized as necessary. When an item is marked as relevant in the Unlabeled Search Results list view, it is removed from that panel and added to the Sandbox list view. Like the Unlabeled Search Results panel, the Sandbox panel displays a term-frequency view below the list view. The list view and term-frequency view include the same visual query capabilities and functionality as mentioned in the preceding sections.

The Sandbox panel is intended to provide an area for collecting relevant records and conducting more focused analysis. At the end of an investigation, the analyst can export these records to an archive for external review, archival, or presentation.

6. ADAPTIVE RE-RANKING OF UNLABELED SEARCH RESULTS

6.1 Tracking User Interactions

As the analyst interacts with the records, Gryffin uses the actions to label relevant and irrelevant records in its local data model. When the analyst clicks the check icon button for a record, the record is labeled relevant in the data model. On the other hand, when the remove button is clicked for a record, it is marked irrelevant. In this manner, the client keeps a running collection of which records are labeled (relevant or irrelevant) and unlabeled as the analyst continues to interact with the interface. The information gathered from these interactions is then used to feed the semi-supervised machine learning algorithms that re-rank the unlabeled records in the list view in a manner that clusters potentially relevant records near the top of the list view.

6.2 Re-Sorting the Records

The biggest challenge when applying learning algorithms in this application is the ever-changing nature of the problems under investigation. In other words, each new investigation initiated by an analyst is potentially unrelated to previous investigations, and therefore, in order to re-rank records for a search, the learning algorithm will typically start with no labeled data. Clustering methods are often used in such cases, but purely unsupervised methods like clustering do not take advantage of an analyst's ability to indicate relevance. The concept of relevance is simple on the surface, but it encodes complex and powerful information that depends upon human experience and expertise.

Since a piece of information that is relevant to a previous investigation is probably not relevant to the current one, prior labeling efforts cannot be applied in most situations. Another challenge for learning algorithms in this setting is that if too much user input is required, the learned model will not become useful until late in the process, when it is less likely to be used. In addition, if the learning requires too much input, it may not provide the time savings that would motivate an analyst to use the tool. Therefore, any learning process must operate effectively with very little guidance.

In order to achieve this effectiveness early enough in the search process, it is necessary to optimize the use of any information that can contribute to learning. One available resource is the set of unlabeled search records. Despite the fact that their current value to the concept being investigated is unknown, these records can help capture structure that can be used to optimize the learning process. The general area of learning that is most applicable in this case is known as semi-supervised learning, where the concept-specific information encoded in the labels is augmented with unlabeled data. However, despite the ability of well-known semi-supervised methods to dramatically reduce the need for labeled documents, most methods are not quick enough in terms of either computation time or sample complexity (the number of labeled samples required). The approach applied in Gryffin deals

with this obstacle by applying well-known, graph-based semi-supervised learning techniques with a modification that makes more optimal use of the labels.

6.3 Modified Graph-Based Semi-Supervised Learning for Optimal Label Usage

Most graph-based methods of semi-supervised learning [17] work off of a manifold assumption [18]. In other words, even if the representation of a sample point (search record) consists of many features (e.g., a search record is a set of feature-value pairs, where each feature is a unique word stem and its value is the number of times it occurs in the record) and is therefore high-dimensional, the intrinsic dimensionality of the problem space is much smaller. If one can find a way to map a point from this high-dimensional space to a low-dimensional one that captures the same variations, then it becomes much easier to learn a model for deciding relevance based on a few labeled search records. Unlabeled search records are used to find such a mapping. Even though textual records are thought to follow a cluster assumption, the same principles can apply, and experimental evidence suggests that it makes little difference to these methods as long as local proximity is important and it can be preserved during dimensionality reduction. In graph-based semi-supervised learning, a graph is used to represent the manifold, and then a method that operates on a matrix representation of the graph, such as Laplacian eigenmaps [19], is used to find a mapping into a low-dimensional space that preserves the local proximity encoded into the graph. In Gryffin, the unnormalized graph Laplacian is used to represent the graph and allow transformation to a new space.

$$L(u,v) = \begin{cases} d_v, & \{\text{if u=v}\} \\ -1, & \{\text{if u and v are adjacent}\} \\ 0, & \{otherwise\} \end{cases}$$

The biggest issue when applying these methods is noise (relative to the target concept being learned) that can dominate the graph construction, leading the method to preserve proximity based on this noise. In Gryffin, noise can include meaningful words that are not pertinent to discerning between relevance and irrelevance. Since this noise can dominate the graph construction, we use the labels to affect the graph construction globally (across all nodes, whether labeled or not). A large number of random subspaces [20] selected from the original feature space are used to build classification models using the small amount of labeled data. These subspace models are used to smooth the graph construction based on fine-grained label-based similarity. Essentially, similarity across a large number of these subspace models can strengthen the bond between two samples, while dissimilarity can weaken the bond. Details of this modified learning approach can be found in a prior publication [21].

Once a graph is built, the generalized eigenvector problem is solved to obtain the mapping. In this case, because the transformation is nonlinear, a linear classifier in the new space is very effective. The approach used from this point is the same as reported by Belkin and Niyogi [22] in which the coefficients for the model are set by minimizing the sum of squared error on the labeled data.

7. CASE STUDY

To demonstrate how Gryffin benefits the analyst, this section describes a brief case study that resembles a typical investigation at the fusion centers. Having the DHS infrastructure reports from 1 November 2010 to 24 March 2011 parsed and indexed in the data repository, suppose that the analyst is interested in records pertaining to the Deepwater Horizon oil spill, which occurred 20 April 2010. During the time period for which we have records, a government study was released as well as other articles on the findings and long-term effects from the disaster. This particular subject works well for a case study because the information is small enough to track individual records and yet large enough to demonstrate the effectiveness of the approach.

To start our study, we enter the term “oil” in the search panel at the top of the display. As shown in Fig. 7, this search results in the server returning a collection of 226 records from the data repository. Initially, we turn our attention to the temporal view which shows a few spikes in the frequency of the search term over time. These spikes can indicate key events for areas of concern.

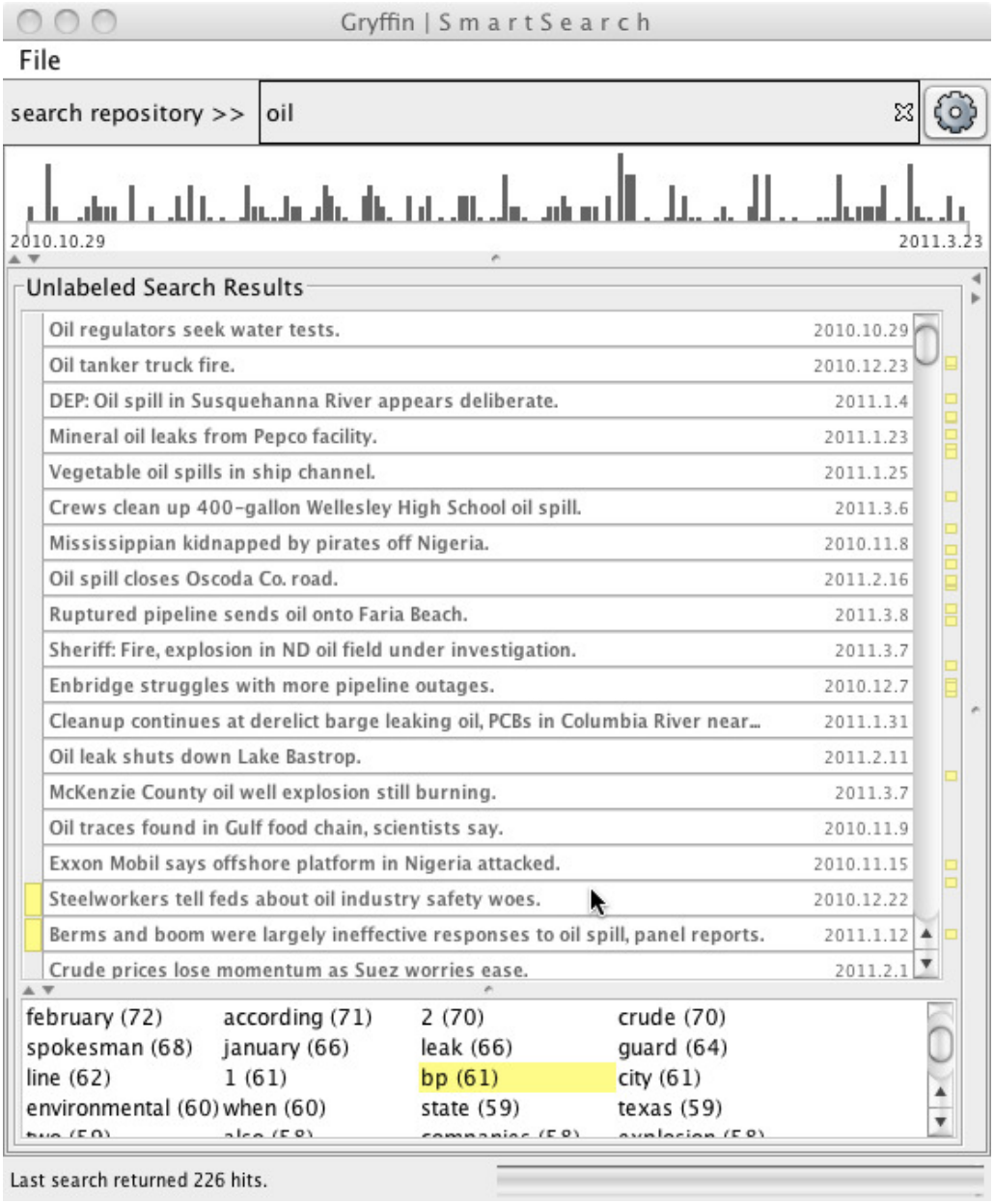


Fig. 7. Searching for the term “oil” returns 226 records. The temporal view shows a few spikes in frequency of records that contain this term. In the term-frequency panel, the “bp” term has been selected and the records containing this term are marked with a yellow indicator box in the list and overview bar. We notice the records listed first are mostly irrelevant to the search focus and most of the records with “bp” are scattered throughout the overall list.

For example, the anti-government protests in Egypt show up as the most significant spike in this figure around the 2 February 2011 point. Regarding, the Deepwater Horizon oil spill, several spikes appear that represent key events, such as the Oil Spill Commission’s report released in early January of 2011.

In Fig. 7, the Unlabeled Search Results panel is shown and the term “bp” is selected in the term-frequency view thereby highlighting the records containing this term in the overview bar and the list view with a yellow indicator. From experience, we know that this term is likely to be associated with records related to the oil spill. The markers in the overview bar indicate that the initial ordering of the search results–which is determined solely by the search scoring on the server–places the relevant articles in a somewhat scattered arrangement. Because some records pertaining to the disaster may not include the “bp” term, the actual scattering may be worse than we show here. We also observe from the record titles that the first 14 records are not relevant to our subject of interest. We can see in Fig. 7 that the first record that is relevant to our subject is the one entitled “Oil traces found in Gulf food chain, scientists say.”

In Fig. 8, we clicked on the overview bar (note the location of the mouse cursor) to scroll the list view to the location of two records near the end of the list which contain the “bp” term. We know these articles are relevant based on the title and summary, but the score from the search resulted in their placement at the bottom of the list. In a typical search, the analyst might not find relevant records near the bottom of the list resulting in these and other relevant records with low search scores being missed altogether. Now in this case, we are dealing with hundreds of records and we could imagine that an analyst could examine each one, although it would take a significant amount of time.

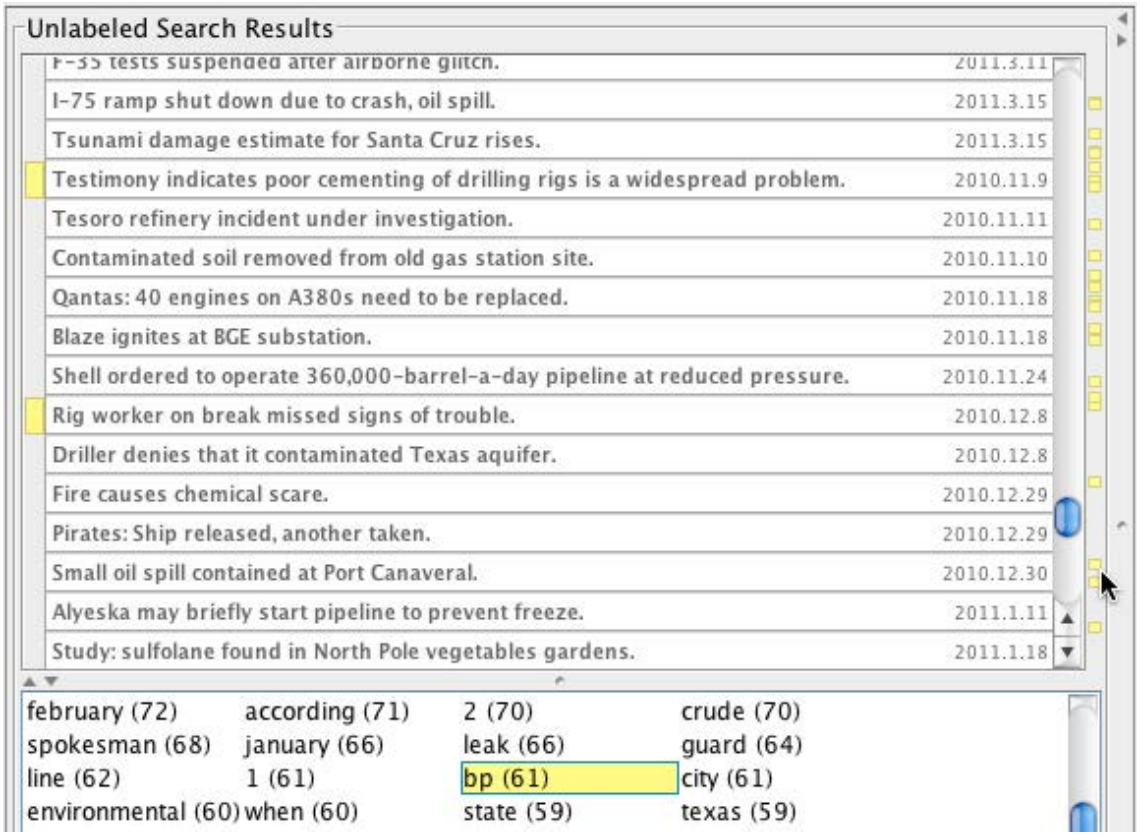


Fig. 8. The lower portion of the unlabeled records list after the initial search. Using the overview bar, the markers for two records were clicked to jump to the location in the list view. It is observed that the two highlighted records, which are relevant, are in the lower portion of the list. In a typical search, these records are likely to be missed.

But what if we consider a typical search that returns thousands of hits, as we show in Fig. 1? The result is information overload for the analyst, and this situation necessitates the integration of practical analytics to assist in identifying relevant records and propagate these records to the top of the list.

Returning to the analysis, we are now focused on labeling several articles at the top of the list. We utilize the visual representation and interaction techniques described in Section 5 to determine the relevancy of the articles. In some instances, the record titles are obscure and we need to focus on the record to see location and/or the summary text. We may also need to follow the link button to the referenced URL to glean more insight on the records of interest. We note that the analyst is ultimately in control of determining which records are relevant or irrelevant thereby harnessing the creativity, background knowledge, and intuition of humans. The purpose of our analytics demonstrated next will be to guide the analyst to records of potential importance.

As described in Section 6, the activity of labeling the records via the analyst's interactions with the interface is used to feed a semi-supervised machine learning algorithm that effectively re-ranks the unlabeled records such that relevant records are moved to the top. In this instance, we label several articles at the top as irrelevant and three records as relevant and we are ready to execute the re-ranking process.

The re-ranking process is started when the analyst clicks the gear icon in the toolbar which is to the right of the search panel. The re-ranking processes the labeled and unlabeled records as described in Section 6 and generates a new ordering. The record list view is then redisplayed, and in this case, we have the configuration shown in Fig. 9. In this figure, the "bp" term is again selected in the term-frequency view to highlight the ordering of records likely to be relevant to the oil spill. The overview bar shows that the majority of records with this term are now residing near the top of the list. Upon inspection, we find that most of the records near the top of the list are indeed relevant to our search. Furthermore, several records near the top that do not have the selected term are also relevant. Comparing the re-ranked list to Fig. 8, we find that the two records we found near the bottom of the list that were relevant have now been moved into the top portion of the list, an action that greatly improves the chances of an analyst finding them.

At this point, the analyst continues to label records, again concentrating on the top of the record list to further refine the collection of relevant records in the Sandbox panel and remove irrelevant records. The analyst reexecutes the re-ranking process on the larger set of labeled data to further improve the search. Figure 10 shows the configuration of the panels after the second re-ranking. Again, the records with the "bp" terms are packed much tighter at the top of the list of unlabeled records. Furthermore, using the overview bar, we can check several highlighted records at the bottom of the list and see that they are in fact irrelevant to our search based on the titles. These results indicate that the labeling and re-ranking processes are successful in propagating records up in the list of unlabeled records, thereby greatly increasing the chances of finding relevant information and decreasing knowledge discovery timelines.

Fig. 9. After the first execution of the re-ranking algorithm, the unlabeled records are reordered in the list view. The "bp" term is selected in the term-frequency panel which highlights the records on the overview bar and the record listings. We notice that the majority of these records, which are likely relevant to our search, are now located near the top of the display. Also, several records shown do not contain the selected term, but are relevant based on the titles shown.

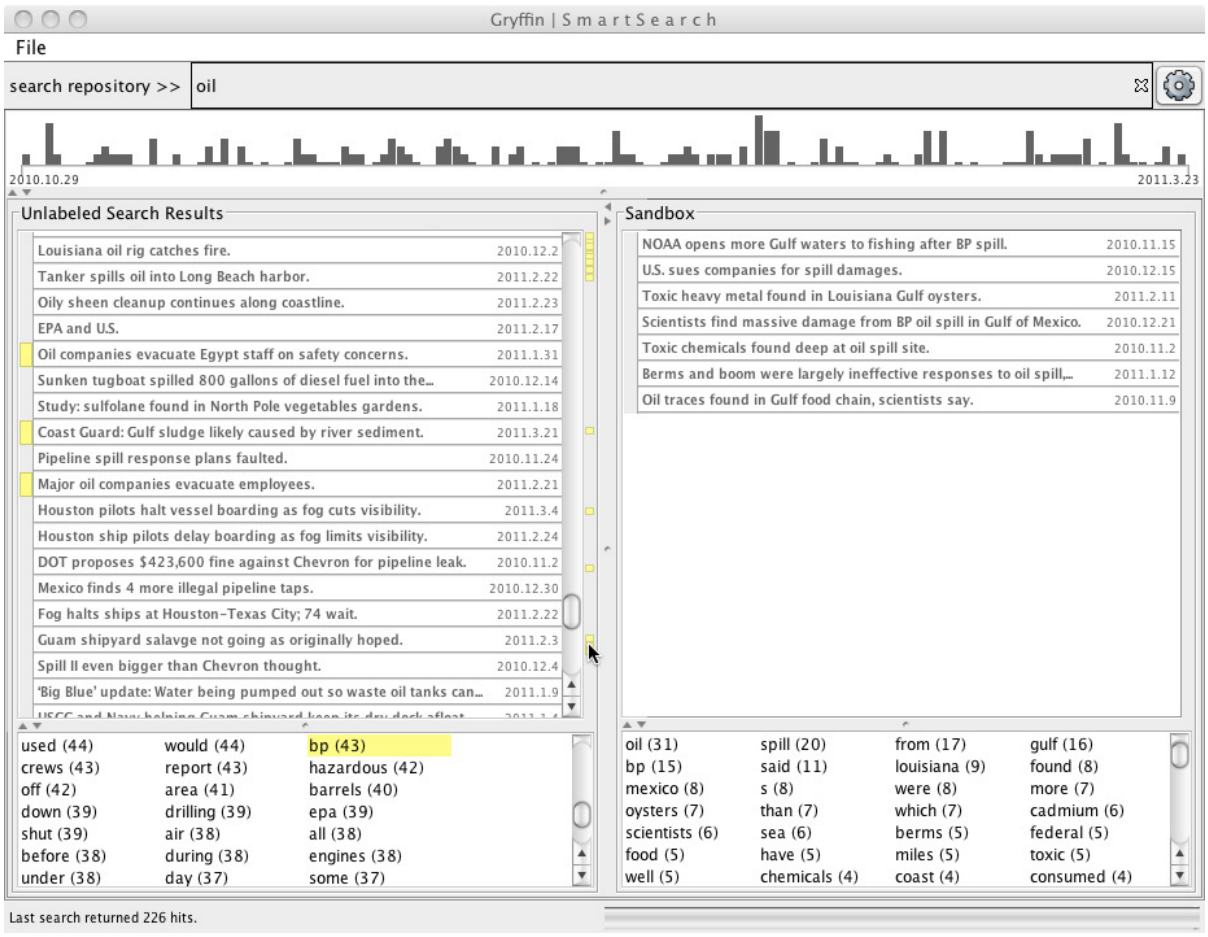


Fig. 10. After the first re-ranking, we concentrate on the top of the list of unlabeled records and continue labeling relevant and irrelevant records. We then reexecute the re-ranking process and have the configuration shown in this figure. We observe that the records appear packed event tighter now based on the occurrence of the "bp" term. In the figure, we have clicked on the two records at the bottom of the list with the term "bp" and we observe from the titles that they are not related to the oil spill topic. This figure also shows the Sandbox after our analysis.

8. TEXT STREAM VISUALIZATION TOOLS

We have developed two additional applications to visualize data from text streams. Stream Trend View displays parallel frequency graphs over time for multiple terms entered by the analyst, while Swarm View displays streamed documents as "birds" in chronological groups, allowing the analyst to observe their interaction based on similarity of content. These programs are currently applied to DHS infrastructure reports but can be adapted to other unstructured information streams from such sources as news feeds, social media, or web pages.

8.1 Stream Trend View

The Stream Trend View tool is designed to help analysts recognize trends in massive, frequently updated collections of text. The analyst enters a search term into the search box, and then a graph (see Fig. 11) of the frequency of that term in the collection of streamed documents over time is shown in a manner similar to the IN-SPIRE applications term graphs [23]. Up to

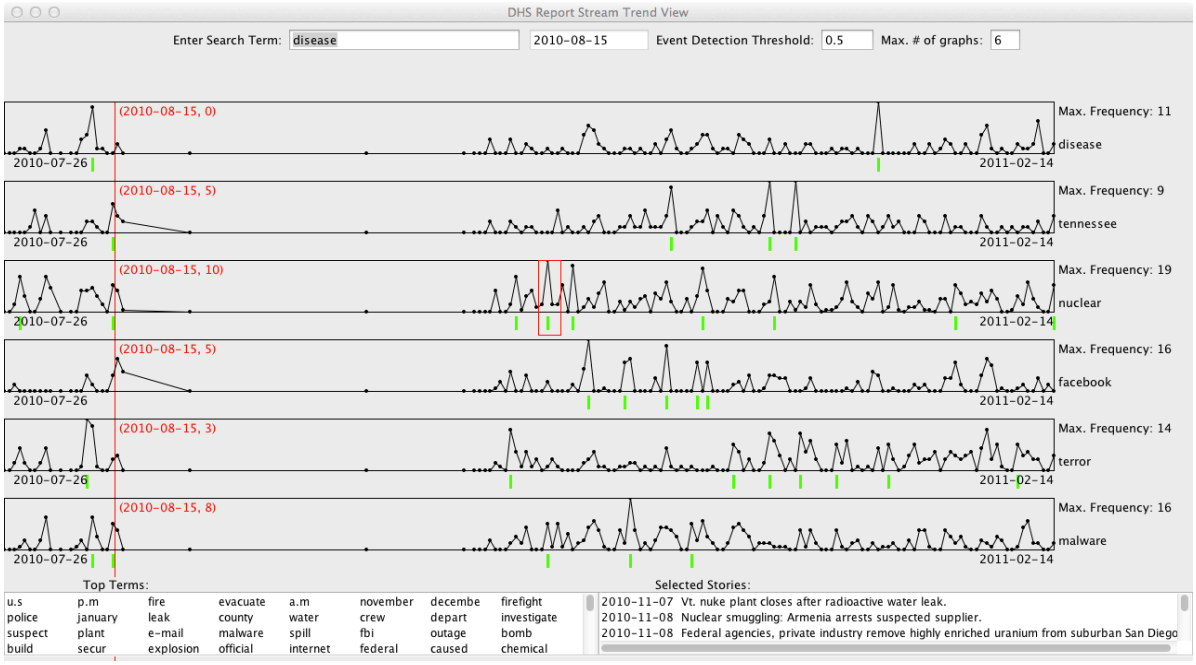


Fig. 11. The Stream Trend View interface allows comparison and analysis of frequency graphs for multiple terms.

20 graphs can be displayed simultaneously, allowing the analyst to investigate a wide variety of stream topics. The graphs are parallel, and moving the mouse displays a vertical line accompanied by the date and frequencies corresponding to the mouse's location, allowing comparison across graphs.

Several features assist the analyst in investigating groups of terms. First, each graph highlights possible significant events related to that graph's term. If the increase in term frequency from one date to the next is greater than half (or another event detection threshold chosen by the analyst) of the maximum frequency, then a green bar appears below the date of the increase on the graph. Also, a list of the top 100 terms for all the textual documents that have been processed provides search suggestions and overall context. Clicking one of these terms places it in the search box. Finally, dragging a box over a graph displays a list of all stories within the selected timeframe containing the selected graph's term. Selecting a title from the list allows the analyst to read that story.

Stream Trend View's current architecture (see Fig. 12) supports the DHS infrastructure reports but can be extended to support other document sources. Summaries of individual news stories are parsed from a directory of DHS report files. A stream simulator converts the summaries to Lucene documents, which store the date, title, and content of a summary, and then adds them one at a time to a data model. When the analyst enters a search term, a new graph panel is created, receives a list of the data model's document collection, and generates a list of data points based on the frequency of the search term in each document's content. Then, when the stream simulator adds a new document to the data model, each frequency graph is updated to include the new data.

The primary purpose of using a stream simulator instead of a fixed document set is to demonstrate that Stream Trend View can be used for many types of streams because of its capacity to update graphs as new stories are added. Replacing the PDF directory and stream simulator components with code to produce and feed Lucene documents from internet text streams would allow the data model and GUI components to remain intact as the stream source changes.

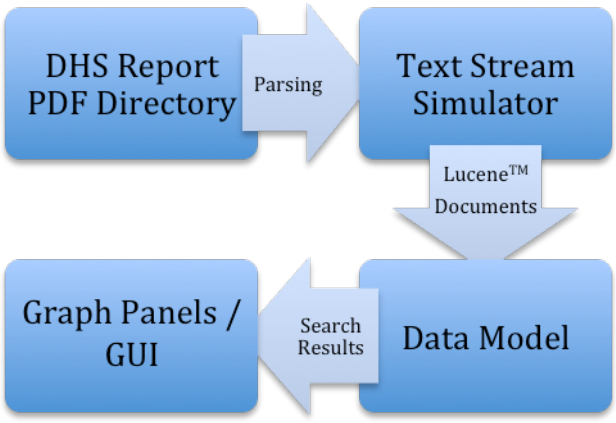


Fig. 12. The DHS reports are converted to streamed documents which are stored in a data model for subsequent search and visual analysis.

8.2 Swarm View

Swarm View (see Fig. 13) displays the degree of similarity in a group of contemporary stories. Each story is represented as a "document bird" which attracts or repels other document birds based on similarity as determined by TF-ICF Document Vectors. As the birds influence each other similar stories tend to cluster together, allowing the analyst to see relationships.

Swarm View's interface has several features to aid in investigation. Pausing the swarm allows the analyst to investigate instantaneous clusters before they change. Also, the analyst can cycle through groups of 100 document birds in chronological order, allowing access to the entire data set even though the swarm panel can only support a limited number of birds. Dragging a box around birds displays their documents' titles and allows the analyst to read their stories by clicking the view button. Finally, the analyst can "save" a document bird of interest. Saved birds are colored red for easy tracking, and they remain present after a new set of birds replaces the current set. This permanence allows investigation of a particular type of story across longer timespans.

Swarm View's architecture (see Fig. 14) follows the same pattern from stream to data model to output as Stream Trend View's. The same DHS Report stream simulator dispenses Lucene documents to a data model, which keeps a sorted list. Swarm View's data model handles 100 sequential documents at a time and creates corresponding document birds, which store position, direction, and speed as well as a document and a document vector. The birds are displayed on a swarm panel, and meanwhile, a swarm runner thread constantly updates the document birds' positions, directions, and speeds by calculating how each bird is influenced by nearby birds. The influence algorithm combines directional attraction by similar birds, directional repulsion by dissimilar birds and any birds that are too close, and speed changes to approach the average speed of nearby birds.

Like Stream Trend View, Swarm View can easily be adapted to any type of text stream that dispenses Lucene documents, allowing application to news and social media. Future work also includes accommodating larger groups of birds on higher performance computers, increasing the convenience of investigation.

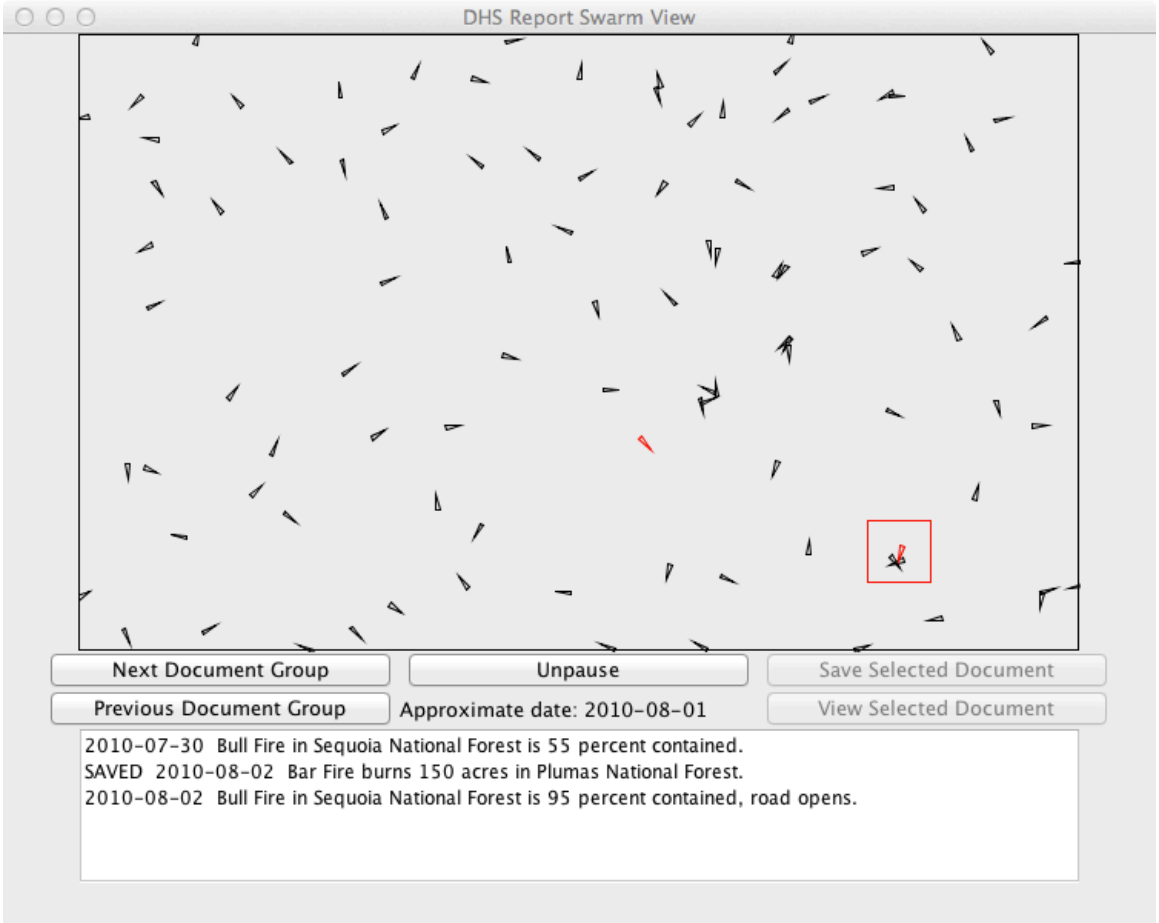


Fig. 13. Swarm View's visual interface allows investigation of clusters of similar document birds.

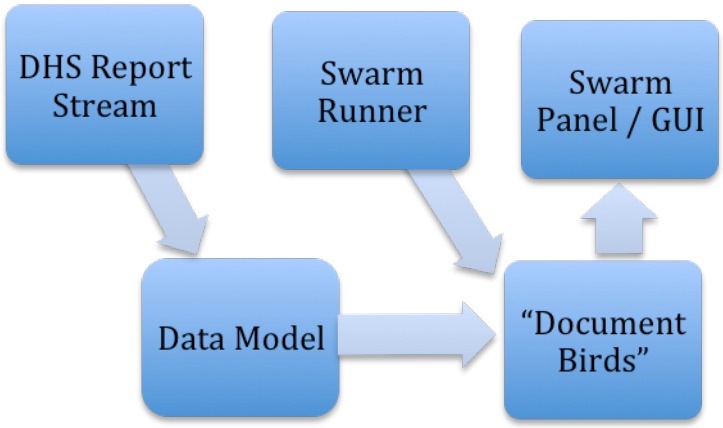


Fig. 14. Swarm View converts streamed text into groups of animated document birds that cluster based on similarity over time.

9. CONCLUSION

We have presented Gryffin, a novel system for investigation search and analysis of textual information that goes beyond current comparable systems by integrating the interactive display of information with semi-supervised machine learning. Gryffin highlights potentially relevant information, drawing directly from the analyst's interactions with the system in an intuitive and practical system. The current system supports the daily tasks faced by analysts at the DHS fusion centers. We plan to continue working with fusion center experts to refine and improve the Gryffin system. In addition to the integration of additional data sources, we are currently developing approaches to handle data provenance in the analysis, collaborative workspaces, and new visualization techniques. Gryffin will also be employed in a broader context of application areas, such as biomedical, climate, and multimedia information analysis.

10. REFERENCES

1. Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., and Ziegler, H., "Visual analytics: Scope and challenges," in *Visual Data Mining*, Simoff, S. J., B¨ohlen, M. H., and Maxeika, A., eds., 76–90, Springer, Berlin, Germany (2008).
2. Reed, J. W., Potok, T. E., and Patton, R. M., "A multi-agent system for distributed cluster analysis," in Proceedings of the International Workshop on Software Engineering for Large-scale Multi-Agent Systems, 152–155, IEEE Computer Society, Edinburgh, Scotland (2004).
3. Potok, T. E., Elmore, M., Reed, J. W., and Sheldon, F. T., "Vipar: Advanced information agents discovering knowledge in an open and changing environment," in Proceedings of the World Mulitconference on Systemics, Cybernetics, and Informatics, 9, 28–33, IIIS, Orlando, FL (2003).
4. Potok, T. E., Elmore, M. T., Reed, J. W., and Samatova, N. F., "An ontology-based html to xml conversion using intelligent agents," in Proceedings of the Hawaii International Conference on System Sciences, 1220–1229, IEEE Computer Society, Big Island, HI (2002).
5. Roberts, J., "State of the art: Coordinated and multiple views in exploratory visualization," in International Conference on Coordinate and Multiple Views in Exploratory Visualization, 61–71 (July 2007).
6. Furnas, G. W., "Generalized fishey views," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 16–23, ACM, New York, NY, USA (1986).
7. Roberts, J. C., and Suvanaphen, E., "Visual bracketing for web search result visualization," in Proceedings of International Conference on Information Visualization, 264–269 (2003).
8. Leung, Y. K., and Apperley, M. D., "A review and taxonomy of distortion-oriented presentation techniques," ACM Trans. on Computer-Human Interaction 1, 126–160 (June 1994).
9. Ahlberg, C., and Shneiderman, B., "Visual information seeking: Tight coupling of dynamic query filters with starfield displays," in Conference on Human Factors in Computing Systems, 313–317, ACM (1994).
10. Stasko, J., Gorg, C., Liu, Z., and Singhal, K., "Jigsaw: Supporting investigative analysis through interactive visualization," in IEEE Symposium on Visual Analytics Science and Technology, 131–138, IEEE Computer Society, Sacramento, CA (2007).
11. Wong, P. C., Hetzler, B., Posse, C., Whiting, M., Harve, S., Cramer, N., Shah, A., Singhal, M., Turner, A., and Thomas, J., "In-spire infovis 2004 contest entry," in IEEE Symposium on Information Visualization, (Oct. 2004).

12. Koch, S., Bosch, H., Giereth, M., and Ertl, T., "Iterative integration of visual insights during patent search and analysis," in IEEE Symposium on Visual Analytics Science and Technology, 203–210, IEEE Computer Society (2009).
13. Clarkson, E. C., Desai, K., and Foley, J. D., "Resultmaps: Visualization for search interfaces," IEEE Transactions on Visualization and Computer Graphics 15(6), 1057–1064 (2009).
14. Nguyen, T. N., and Zhang, J., "A novel visualization model for web search results," IEEE Transactions on Visualization and Computer Graphics 12(5), 981–988 (2006).
15. Chung, H., Yang, S., Massjouni, N., Andrews, C., Kanna, R., and North, C., "Vizcept: Supporting synchronous collaboration for constructing visualizations in intelligence analysis," in Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on, 107–114 (2010).
16. Keim, D. A., Bak, P., Bertini, E., Oelke, D., Spretke, D., and Ziegler, H., "Advanced visual analytics interfaces," in Proceedings of the International Conference on Advanced Visual Interfaces, 3–10, ACM, Roma, Italy (2010).
17. Chapelle, O., Scholkopf, B., and Zien, A., eds., *Semi-Supervised Learning*, MIT Press, Cambridge, MA (2006).
18. Lin, T., and Zha, H., "Riemannian manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5), 796–809 (2008).
19. Belkin, M., and Niyogi, P., "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, 15(6), 1373–1396 (2003).
20. Beryll, R., Gutierrez-Osuna, R., and Quek, F., "Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, 36, 1291–1302 (2003).
21. Symons, C. T., and Arel, I., "Multi-view budgeted learning under label and feature constraints using label-guided graph-based regularization," in International Conference on Machine Learning, Workshop on Combining Learning Strategies to Reduce Label Cost, (2011).
22. Belkin, M., and Niyogi, P., "Semi-supervised learning on riemannian manifolds," *Machine Learning*, 56, 209–239 (2004).
23. Berry, M. W., and Kogan, J., *Text Mining: Applications and Theory*, pp. 169–172, John Wiley & Sons, Ltd., United Kingdom (2010).

Southeast Region Research Initiative
National Security Directorate
P.O. Box 6242
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6252

www.serri.org