TIARA: A Visual Exploratory Text Analytic System

Furu Wei [†], Shixia Liu [†], Yangqiu Song [†], Shimei Pan [‡], Michelle X. Zhou [§]
Weihong Qian [†], Lei Shi [†], Li Tan [†] and Qiang Zhang [†]
[†]IBM Research – China, Beijing, China
[‡]IBM Research – T. J. Watson Center, Hawthorne, NY, USA
[§]IBM Research – Almaden Center, San Jose, CA, USA
[†]{weifuru,liusx,yqsong,qianwh,shllsh,tltan,zhqiang}@cn.ibm.com
[‡]{shimei}@us.ibm.com
[§]{mzhou}@us.ibm.com

ABSTRACT

In this paper, we present a novel exploratory visual analytic system called TIARA (**T**ext **I**nsight via **A**utomated **R**esponsive **A**nalytics), which combines text analytics and interactive visualization to help users explore and analyze large collections of text. Given a collection of documents, TIARA first uses topic analysis techniques to summarize the documents into a set of topics, each of which is represented by a set of keywords. In addition to extracting topics, TIARA derives time-sensitive keywords to depict the content evolution of each topic over time. To help users understand the topic-based summarization results, TIARA employs several interactive text visualization techniques to explain the summarization results and seamlessly link such results to the original text. We have applied TIARA to several real-world applications, including email summarization and patient record analysis. To measure the effectiveness of TIARA, we have conducted several experiments. Our experimental results and initial user feedback suggest that TIARA is effective in aiding users in their exploratory text analytic tasks.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications—*Data mining*; H.3.1 [**Information Storage and Retrieval**]: [Content Analysis and Indexing]; H.4 [**Information Systems Applications**]: User Interfaces

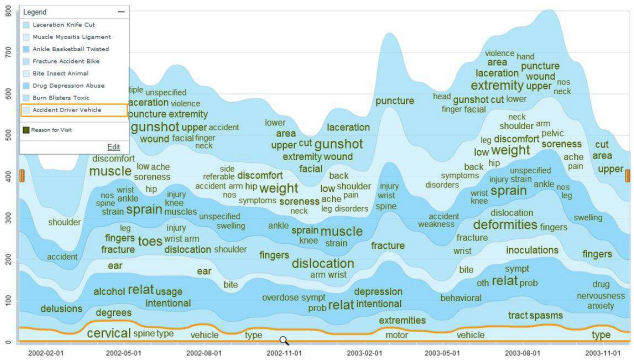
General Terms

Algorithms; Experimentation

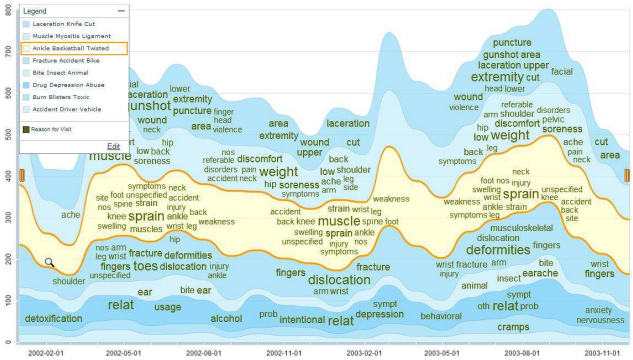
Keywords

Visual text analytics, exploratory analytics, document summarization, topic analysis, interactive visualization

KDD'10, July 25–28, 2010, Washington, DC, USA.



(a) Eight summarized topics



(b) Zoomed in the central topic to reveal more details

Figure 1: TIARA's visual summary of the "reason for visit" field of the 23,000$^+$ patient emergency room records, depicting 8 major reasons of visit.

1. INTRODUCTION

Most of the enterprise information is contained in the form of text documents. Sifting through piles of text documents to locate needed information or simply deciding which documents to preserve for business compliance becomes very costly and time-consuming. It thus becomes increasingly more important to develop effective text analytic tools that help users rapidly view, explore, and analyze large collections of text. Although a number of text analysis technologies have been developed [11, 1], the results produced by these technologies, which are often abstract and complex, may not be consumable by users. To help users comprehend

the text analysis results, researchers have employed interactive visualization to help explain such results. For example, several basic visual metaphors, such as scatter plot [10, 4] or stacked graph [14, 9], have been used to present analytic results.

To aid users in exploring and analyzing large text collections, we are building TIARA (Text Insight via Automated, Responsive Analytics), a novel visual analytic system that combines a set of rich and highly interactive visualization techniques with advanced text analytics. Previously, we have described how TIARA can automatically generate a visual summary of text analytic results [15]. Here, in this paper we focus on TIARA's text analytic engine and its support of interactive visual exploration of analytic results.

TIARA's text analytic engine first derives topics from a collection of documents using topic analysis techniques. Then it uses Lucene[1] to index each document and its associated topics to support interactive topic and text exploration. Given a user query (e.g., "visual analytics" for all the documents containing these keywords), TIARA is able to present a summary of desired documents. The summary contains a set of topics, each of which is further represented by a set of keywords (Fig. 1). To depict the content evolution within each topic over time, TIARA also derives time-sensitive keywords (Fig. 1). To help users understand and consume the derived topics and their changes over time, TIARA allows the users to view and inspect its text analytic results at different levels of granularity. For example, Fig. 1(a) shows the overview of eight derived topics, summarizing patients' *reason of visit* to a hospital emergency room; and Fig. 1(b) presents a fisheye view, where the middle topic is drilled down and expanded to reveal more details.

TIARA has been applied to several real-world applications including email summarization and patient record analysis. Throughout the paper, we use examples from these applications to illustrate the main functions of TIARA.

The remainder of the paper begins with a discussion of related work, followed by an overview of the TIARA system. We then present TIARA's text analytic engine and its support of interactive visual exploration of analytic results. We also describe two TIARA applications and initial user feedback.

2. RELATED WORK

Our work is most relevant to two bodies of research effort: topic analysis and text visualization.

2.1 Topic Analysis

Researchers have developed various approaches to topic analysis, including supervised topic annotation (i.e., document classification) and unsupervised topic modeling. Here we focus on unsupervised topic modeling since it is most relevant to our work. Among them, document clustering [11] and latent topic modeling [2] are the most common approaches.

Document clustering automatically organizes a document collection by its topic structure. Scatter/Gather [18, 12] is a cluster-based browsing technique for analyzing large text collections. It was shown to be effective in helping users build a "more coherent conceptual image of a text collection" [18].

Latent topic modeling such as LDA (Latent Dirichlet Allocation) [2] has been employed widely in text analysis. For example, in [25], LDA was used to improve ad-hoc document retrieval. LDA was also used in [6] to select summary keywords from emails. There is also work on topic trends analysis. For example, Mei and Zhai [17] proposed a general probabilistic method to build an evolution graph of themes. Wang et al. [23] presented an LDA-style topic modeling algorithm that produces interpretable topical trends.

While TIARA can benefit from all the topic modeling techniques mentioned above, it focuses more on postprocessing the topic modeling results to present the most meaningful content to the users. Furthermore, TIARA also leverages advanced visualization to make the analysis results easily comprehensible.

2.2 Text Visualization

Visualization has been used extensively to present web search results. Xu et al. [26] introduced a new visual search interface which groups the returned results of a third-party search engine (such as Google). There are web clustering engines, such as Carrot2[2], which used visualization techniques to display the search results. Readers may refer to [3] for a survey on web clustering engine. Visualization techniques were also widely used in faceted browsing. For example, Clarkson et al. [5] described ResultMaps, which used a hierarchical Treemap to organize the search results. Smith et al. [20] presented FacetMap, an interactive, query-driven visualization to present metadata-rich data stores. It provided users with a dynamic, faceted browsing experience. More recently, FacetLens [13] extended FacetMap to allow users to observe trends and explore relationships within faceted datasets.

In addition to displaying search results, there is work on visualizing other types of text. For example, MemeTracker [14] is a new framework for tracking the changes of short phrases in online text over time while [10, 4] focused on visualizing the results of document clustering. Word Tree [24] and Phrase Net [8] illustrated text content at the word and phrase level, respectively. While the above systems primarily used simple visualizations (e.g., scatter plot) to organize and display the analysis results, TIARA automatically creates more informative and powerful visual text summaries for analyzing large and complex data sets. In addition, compared to previous work, TIARA provides rich interactions to aid users in their exploratory text analytic tasks.

3. TIARA SYSTEM OVERVIEW

TIARA is designed to visually analyze the topics in a text collection and their content changes over time to facilitate exploratory text analytics. With TIARA, users can interactively view, explore, and analyze large collections of text. Here, we provide an overview of TIARA, starting with its user interface, followed by its system architecture.

3.1 User Interface

Fig. 3 shows the user interface of TIARA implemented for an email application. When interacting with TIARA, a user may enter a query term to search an email collection (Fig. 3a). The search results are presented in two views: the list view (Fig. 3b) and the topic view (Fig. 3c). The list view

[1]http://lucene.apache.org/

[2]http://search.carrot2.org

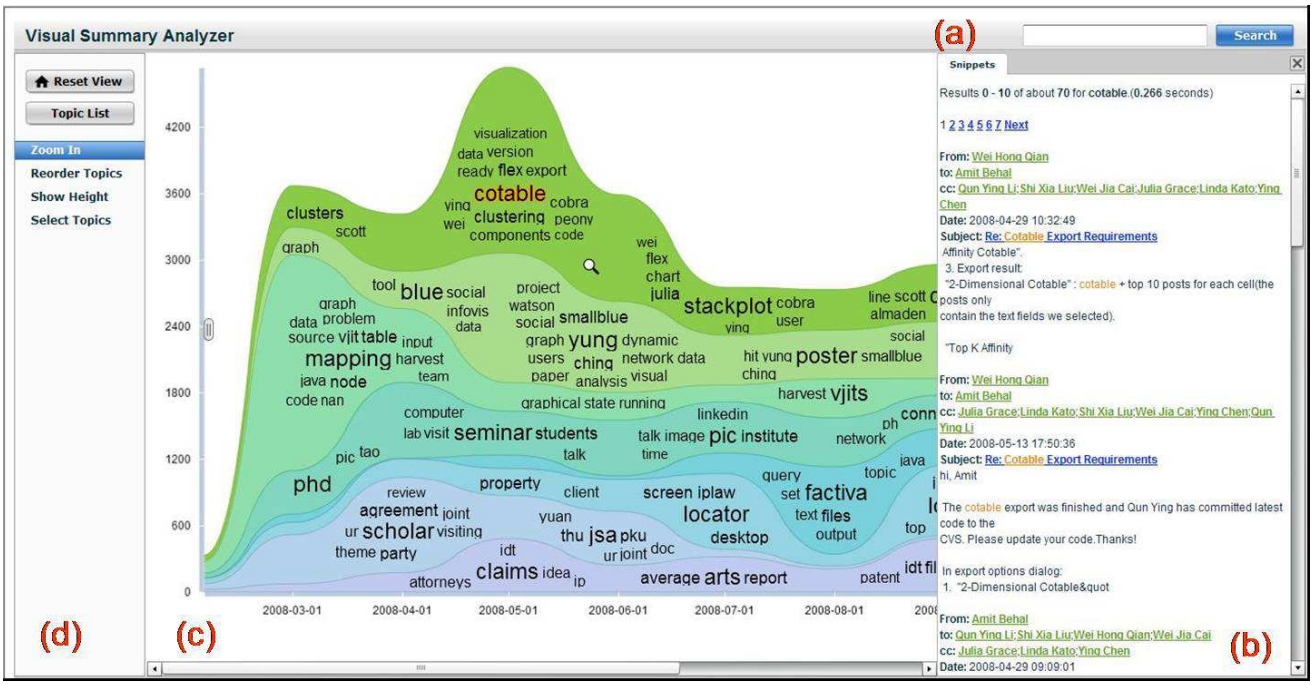


Figure 3: TIARA's visual summary of 8,000$^+$ emails. In the visualization, each layer represents a topic, which is described by a set of keywords. We show the top 8 topics out of 18 topics in total. These topic keywords are distributed along time, summarizing the content evolution over time. The x-axis encodes the time and the y-axis encodes the strength of each topic. For each topic , the height encodes the number of emails of the topic at a particular time. From the height of each topic and its content distributed over time, the user can observe the topic evolution over time.

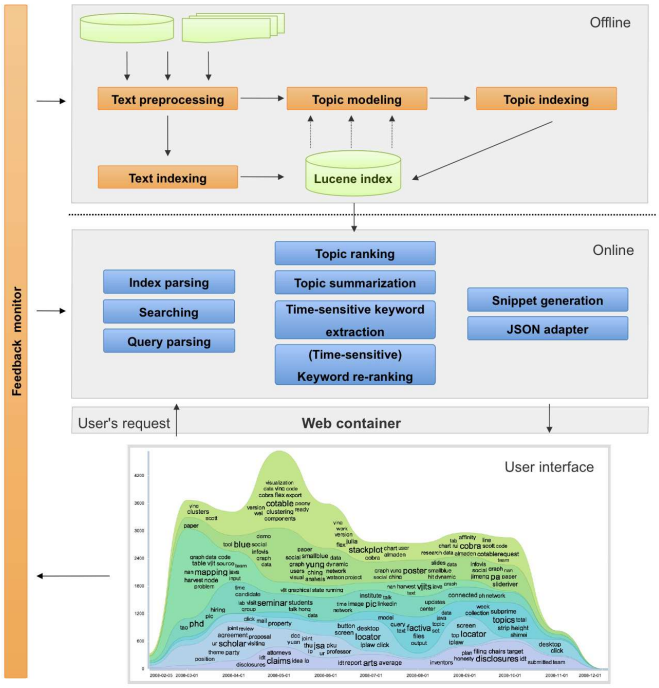


Figure 2: System architecture of TIARA

displays a list of email snippets that match the user query. In contrast, the topic view summarizes the retrieved emails visually as a stacked graph. Based on the visualization in the topic view, users can easily find the most salient topics. They can also understand how each topic changes over time, including its strength and content. Users can also interact with both the list and the topic views to perform further analysis. For example, a user may filter the search results based on the sender information in the list view; or s/he may drill down to a subset of the documents based on a topic or keyword by clicking on the topic or keyword in the topic visualization. Moreover, TIARA provides additional options for users to explore the topics. For example, a user can click on the *topic list* button (Fig. 3d) to show a linear list of topics without the detailed trend information.

3.2 System Architecture

Fig. 2 illustrates the TIARA architecture which includes an offline processing module and an online processing module.

As part of the offline processing, TIARA collects text documents from either a file system or a database. The gathered documents are then processed by TIARA's *text pre-processing* component. For example, the *text pre-processing* component in an email application performs stop word and email signature removal to reduce the noises in the emails. The *text indexing* component uses Lucene to index and store the textual content and the associated meta data. The processed text documents are then sent to the *topic modeling* component (Section 4.1), which automatically derives

a set of latent topics. The topic analysis results, including the topic-keyword distributions and document-topic distributions, are also stored and indexed by Lucene during the *topic indexing*.

TIARA's online process is often triggered by a user request (e.g., a keyword query). In *query parsing*, TIARA analyzes a user's request and then converts it to a Lucene-compatible query[3], which is then used to find the matched documents from the text index. Once documents are retrieved during the *searching* process, they are displayed in a list view and a topic view. To automatically generate the topic view, TIARA first extracts topic-related information from each document, including its topic keywords and document-topic distributions from the Lucene index using the *index parsing* component. Subsequently, the keyword-based topic summaries are generated by the *topic summarization* component (Section 4.3) and the time-sensitive keywords are selected and ranked by the *time-sensitive keyword extraction* and *time-sensitive keyword re-ranking* components (Section 4.4). The topics are also ranked by the *topic ranking* component (Section 4.2) so that the most meaningful topics can be shown first. Finally, the results are serialized and represented in *JSON* objects[4], which are then sent to the TIARA visualization to produce the final visual summary. Moreover, during the *snippet generation* stage, TIARA generates a snippet to summarize the content of each retrieved document. Users can also interact with TIARA to provide feedback to the backend analysis components or articulate new data needs, TIARA's *feedback monitor* collects user feedback to further refine and update the analysis results. We will describe the details of this feature in section 5.2.3.

In TIARA, the web container and the HTTP protocol are used to communicate and transfer data between the visualization and text analytic modules. The TIARA server is implemented as a JAVA servelet using Tomcat[5] as the web container. This design makes it easy to deploy TIARA across multiple platforms.

TIARA is designed to process text collections with time stamps. It provides a set of text processing and interactive visualization components that can help users explore and analyze text collections. Most components in TIARA can be reused in different applications. The components that often require customization are the *text pre-processing* component and the *text indexing* component. For example, in an email application, the *text pre-processing* component may perform email signature removal which is not needed in the patient record analysis application. Similarly, depending on the data fields available in an application, the *text indexing* component may need to index different structured or unstructured fields. Moreover, the TIARA system is easily extensible and configurable. We have defined a set of standard APIs for each core analytic component, such as the *time-sensitive keyword extraction* and *time-sensitive keyword re-ranking* components. With these APIs, users can define new analytic functions by extending TIARA's existing analytic capabilities.

[3]`http://lucene.apache.org/java/2_4_0/queryparsersyntax.html`

[4]http://www.json.org/

[5]http://tomcat.apache.org/

4. TIARA ANALYTICS

In this section, we explain TIARA's backend text analysis engine and the data interface between the text analytic module and the visualization module.

4.1 Topic Analysis

Topic analysis can be performed in different ways. For example, a document can be classified into pre-defined categories. Document classification however typically requires a large number of human annotated training examples which can be costly to obtain in practice. As a result, it is often desirable to use unsupervised learning methods that automatically discover hidden themes in a document collection. We have built a general TIARA framework, and it currently supports k-means clustering [11], information-theoretic co-clustering (ITCC) [21], LDA [2], and hierarchal Dirichlet processes (HDP) [27]. Other clustering algorithms [11] and topic models [1] can also be embedded easily in TIARA.

In topic analysis, we denote a text collection with N documents as $D = \{d_1, d_2, \cdots, d_N\}$. Each document is composed of a sequence of words $d_i = \{w_{i,1}, w_{i,2}, \cdots, w_{i,N_i}\}$ where N_i is the number of words in d_i. Let $V = \{v_1, v_2, \cdots, v_V\}$ denote the vocabulary of size V and K denote the number of topics. Without loss of generality, we use the document-topic distribution matrix $\Theta \in \mathbb{R}^{N \times K}$ and the topic-word distribution matrix $\Phi \in \mathbb{R}^{K \times V}$ to summarize the topic analysis results. Each row of these two matrices represents a probability distribution. This distribution can be derived from either LDA [1] or clustering. For LDA, the distributions are directly inferred from a document collection by various inference techniques such as Gibbs sampling [7]. For clustering, the document-topic distribution is a matrix of 0s and 1s where 0 means that a document does not belong to a topic while 1 means it belongs to a topic. Moreover, the topic-word distribution can be approximated by the term frequencies in each cluster.

TIARA uses a threshold-based method to assign one or more topics to a document. In particular, a topic (say $j, 1 \leq j \leq K$) is assigned to a document d_i if the document-topic probability $\Theta_{i,j}$ is greater than a pre-defined threshold ξ[6]. TIARA also stores and indexes the topic labels as well as the topic assignment for each word token in a document collection using Lucene. This information will be used later to generate time-sensitive topic keywords. For LDA, the token-topic assignment is directly inferred by LDA using Gibbs sampling. For clustering, we assign the cluster label to all the words in a document.

4.2 Topic Ranking

Both LDA and clustering are general topic models and their outputs may not directly satisfy all the information needs of users. For example, the topics derived by LDA are randomly ordered, and users need to manually navigate the entire topic list to find the ones that they are interested in the most. If there is a large number of topics, this task becomes quite difficult and time consuming. To better help users find important topics, we rank all the topics such that the most important ones will be visualized first. Several methods have been investigated in [22] for topic ranking. Here we use a strategy that has both reasonable ranking performance and modest memory and CPU requirement.

[6]We set ξ to 0.3 in our experiments.

Specifically, we consider topics that cover a significant portion of the corpus content more important than those covering less content. Moreover, we consider topics that appear in all the documents to be too generic to be interesting. As a result, the topic rank is measured by a combination of both topic content coverage and topic variance. The detailed topic ranking algorithm is presented in Algorithm 1.

Algorithm 1 *rank_topics*(Θ)

Require: Document-topic distribution matrix Θ.

1: **for** $i = 1$ to K **do**
2: $\mu_i = \sum_{j=1}^{N} N_i \cdot \Theta_{j,i} / \sum_{j=1}^{N} N_i$
3: $\sigma_i = \sqrt{\sum_{j=1}^{N} N_i \cdot (\Theta_{j,i} - \mu_i)^2 / \sum_{j=1}^{N} N_i}$
4: $r_i \triangleq (\mu_i)^{\lambda_1} \cdot (\sigma_i)^{\lambda_2}$
5: **end for**
6: Rank the topic i according to r_i {λ_1 and λ_2 are the control parameters.[7]}.

4.3 Keyword based Topic Summarization

The topic keywords derived by LDA or clustering may not be ideal for users to understand the definition of a topic. For example, when LDA is applied to a financial news corpus, common words such as Dow, Jones, Wall, Street etc., are ranked high in many topics because they are relevant to all the topics. These words however are not useful in helping users identify interesting news topics since all of them are too general. To better help users understand the content, we re-rank the keywords within a topic to refine its topic definition. Inspired by the TF-IDF weighting scheme [19] used in the information retrieval community, we compute the rank of a topic keyword for topic j using Algorithm 2. Basically, if a word occurs frequently in a topic, it is important. Moreover, if the word also appears in many other topics, the word is not important because it is too common.

Algorithm 2 *summarize_topic*(Φ, j)

Require: Topic-word distribution matrix Φ.
Require: The summary length $LENGTH$.

1: Read and parse the keywords for topic j, denoted as W.
2: **for** each word w_m in W **do**
3: The weight of w_m, denoted by $weight(w_m)$, is calculated by, $weigt(w_m) = \Phi_{j,m} \cdot log \frac{\Phi_{j,m}}{(\prod_{k=1}^{K} \Phi_{k,m})^{\frac{1}{K}}}$.
4: **end for**
5: Rank W w.r.t. $weight(w)$.
6: **return** The top $min(LENGTH, |W|$ keywords as keyword based summary. {$|W|$ denotes the size of W.}

4.4 Time-sensitive Keyword Extraction

To allow the user to visually analyze content evolution over time, TIARA automatically selects time-sensitive topic keywords for different time segments. Given a text collection, TIARA first breaks the documents into several sub-collections, each of which is associated with a particular time interval. Note that we do not use fixed time intervals to divide the collection. In general, users are more likely to be interested in the time segments during which a topic is most active, we design an algorithm that will not break a topic near the peaks of a topic layer.[8] Algorithm 3 shows the details of this method.

Algorithm 3 *split_time*($start, end$)

Require: The number of maximal segments MAX_SEG.
Require: The number of time interval $total\_time\_interval$. {Calculated by rules (defined in Table 1) w.r.t. the $start$ and end time of the topic.}

1: $step = total\_time\_interval / MAX\_SEG$
2: Calculate and collect the peaks of the topic trend.
3: $k = 0, begin = 0$
4: **while** $k < total\_time\_interval$ **do**
5: $k = step\_func(k, total\_time\_interval, step)$
6: **if** k is a peak **then**
7: $k = step\_func(k, total\_time\_interval, 1)$
8: **end if**
9: Mark k as a time tick {i.e. (begin, k] as a sub-segment of the topic trend}
10: $begin = k$
11: **end while**
12: **return**

13: $step\_func(k, end, step\_size)$
14: $incr = 0$
15: **while** $k < end$ **do**
16: **if** $++incr > step\_size$ **then**
17: break
18: $++k$
19: **end if**
20: **end while**
21: **return** k

Algorithm 4 *extract_ts_keyword*($D_{j,i}$)

Require: Topic-word distribution matrix Φ.
Require: The maximal number of keywords MAX_TS_NUM.

1: Read and parse the keywords for topic j from $D_{j,i}$, denoted as W, and collect the frequencies of the keywords TF.
2: Re-rank the TF scores by the topic-keywords probabilities. For keyword w_m, $\eta_1 \cdot \frac{TF_{j,i,m}}{\sum_i TF_{j,i,m}} + \eta_2 \cdot \Phi_{j,m} \cdot log \frac{\Phi_{j,m}}{(\prod_{k=1}^{K} \Phi_{k,m})^{\frac{1}{K}}}$.[9]
3: Rank the keywords, and select the top $min(MAX\_TS\_NUM, |W|)$ keywords as time-sensitive keywords.

Once the documents of a topic are divided into several sub-collections based on Algorithm 3, TIARA extracts time-sensitive keywords for each sub-collection. It uses the following criteria to determine the importance of a topic keyword in a sub-collection: (1) if a word occurs frequently in the sub-collection, it is important; (2) if the word occurs

[7] We set $\lambda_1 = \lambda_2 = 1$.

[8] The time tick t is a peak $iff\ h(t-1) < h(t) > h(t+1), \forall 1 < t < total\_time\_interval$, where $h(t)$ denotes the number of documents (or words) belonging to the topic at time range $(t-1, t]$.

[9] η_1 and η_2 were set empirically, and we set $\eta_1 = \eta_2 = 0.5$ in the current TIARA.

in many other sub-collections, the word is not important. Specifically, let $D_{j,i}$ denote the documents about topic j for sub-collection i, the time-sensitive keywords are selected based on Algorithm 4.

4.5 Data Interface

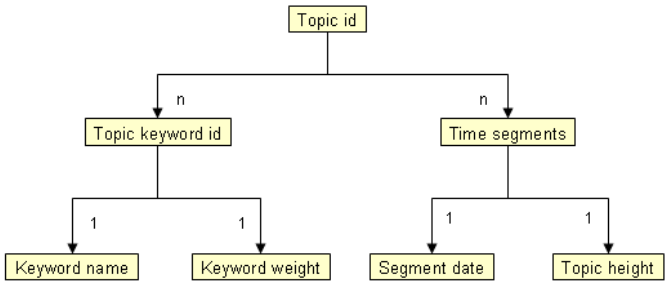


Figure 4: TIARA data interface for a topic.

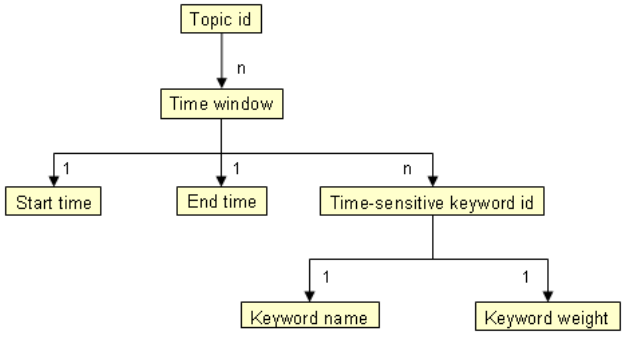


Figure 5: TIARA data interface for time-sensitive keywords.

TIARA also defines a set of standard data interfaces between its backend analysis engine and its visualization module so that both of them can be used independently. For example, if a user is only interested in using TIARA visualization s/he can still take advantage of TIARA as long as the data to be visualized are described in TIARA's data interface. Currently, TIARA's data interfaces include several JSON objects. Fig. 4 shows the schema describing a topic while Fig. 5 is the schema for a topic layer that includes a sequence of time-sensitive keyword collections.

5. VISUAL SUMMARY INTERACTION

Since the analysis results derived above are complex, it can be a challenge to explain them to average users. In the following, we explain how TIARA leverage advanced interactive visualization to visually summarize text analysis results. In particular, TIARA automatically generates a visual topic summary for a document collection. It also allows users to further interact with the visual summary to: (1) examine the data from multiple perspectives; and (2) express additional information needs. Previously, we have described how to generate a visual summary [15], here we focus on explaining how a user can interact with such a summary.

5.1 Visual Design

Previous studies on information visualization have shown that metaphor compatibility has a significant effect on comprehension [28] since it takes time and effort for users to interpret new and unfamiliar visual metaphors. Thus one of our design principles is to use common visual metaphors to convey analytic results. To visualize topics derived by LDA or clustering as well as their content changes over time, we choose a stacked graph based visual layout. Then we augment the stacked graph visualization to generate a keyword-based visual text summarization.

Fig. 3 shows a visual summary created by TIARA. Each colored layer represents a topic. Each layer is depicted by a set of keyword clouds, summarizing the topic content and the content evolution over time. The height of a layer at a time point encodes the "strength" of the topic. In TIARA, the "strength" is measured by the number of documents covering the topic. Please refer to [15] for more details on how to generate such a visualization.

5.2 Multi-level Visual Document Inspection

Since the output of text analytics is not always perfect, to address this problem, we design several useful visual metaphors and rich interactions to allow users to interactively refine and improve text analytics.

5.2.1 Keyword-based Topic Drilldown in Context

There are many keywords in each topic and sometimes it is unclear why a particular keyword is associated with a topic. To help users understand a keyword-topic relations, TIARA allows a user to drill down from a topic to a set of documents containing such a keyword. For example, initially a user may not understand why "cotable" appears in the first topic in Fig. 3. When the user clicks on the keyword, TIARA automatically retrieves a set of emails containing "cotable". After inspecting the results, the user now understand why "cotable" is related to the topic.

5.2.2 Detail Rollup in Context

Another feature in TIARA is to allow users to freely rollup from low level documents to high-level topics in context. For example, in the email application, when a user examines the emails of a topic, s/he may want to know who is involved in those emails and what has been discussed. For example, in Fig. 3, a user is reading an email sent by Amit. Since the message is very important, s/he may also want to retrieve all the emails sent by Amit. In this case, TIARA visually summarizes the content of the retrieved emails. Similarly, when a user is interested in a keyword such as "cotable", s/he may want to examine the topics associated with the word. In TIARA, the user can type "sender:Amit" or "cotable" in the search input area (Fig. 3a) to obtain the topic visualizations associated with a sender or keyword.

5.2.3 Interactive Text Analytics

The text analytic results are not always perfect, thus it is important to allow users to provide feedback to the backend system via visual interactions to compensate for the deficiencies of the analysis. We have done some preliminary work on this. Particularly, at present TIARA supports interactive

Table 1: Rules for calculating *total_time_interval*: *start* and *end* are the input time. $start_{year}$, $start_{month}$, and $start_{day}$ denote the specific *year*, *month*, and *day* of the start document, and end_{year}, end_{monty}, and end_{day} denote the specific *year*, *month*, and *day* of the end document.

Time Span	X-axis Scale	*total_time_interval*
$end_{year} - start_{year} \geq 24$	Year	$\lfloor (end_{year} - start_{year})/12 \rfloor$
$end_{year} - start_{year} \in [5, 24)$	Year	1
$end_{year} - start_{year} < 5$ & $end_{month} - start_{month} \geq 24$	Month	$\lfloor (end_{month} - start_{month})/12 \rfloor$
$end_{month} - start_{month} \in [5, 24)$	Month	1
$end_{month} - start_{month} < 5$ & $end_{day} - start_{day} \geq 30$	Day	$\lfloor (end_{day} - start_{day})/15 \rfloor$
$end_{day} - start_{day} \in [15, 30)$	Day	2
$end_{day} - start_{day} < 15$	Day	1

topic merging and splitting. For example, in Fig. 3, the user finds that both the first and third topics from the bottom are talking about "disclosure", and s/he would like to merge these two topics into a single one. Such actions will affect the topic analysis results or even the existing topic model. As for topic merging, when the user sends a merging request of two topics, the backend analytic module will automatically update the topic index by mapping different topic IDs into a single one so that both topics can be retrieved and processed together. As for topic splitting, when the user finds that one topic is still a mixture of several topics, s/he will ask the backend system to split it into several topics. This request is collected by the *feedback monitor* component and then an LDA or clustering-based topic model is employed on the documents belonging to the topic to divide them into the user-specified number of topics. We finally update the topic index which will be reflected in the later TIARA visualization.

6. EXPERIMENTS

To evaluate the effectiveness of TIARA's analytic engine, we need to examine the quality of three main procedures: (1) topic ranking; (2) keyword based topic summarization; and (3) topic segmentation and time-sensitive keyword selection. In this experiment, we used the MALLET topic modeling toolkit [16] to perform LDA topic analysis. The initial model parameters were set to the default values, and the maximum iteration was set to 1000. Previously, we have reported the evaluation results of the topic ranking and keyword based topic summarization procedures in [22], therefore, here we focus on the time-sensitive keyword extraction procedure. We will also present some experimental results on TIARA's online response time.

6.1 Experiment on Time-sensitive Keyword Selection

Time-sensitive keywords are used to convey the content evolution of a topic over time. For each time segment, we select a set of most representative keywords for a topic. To evaluate the quality of time-sensitive keyword selection, we conducted our experiments using two data sets: a personal email collection with 8326 email messages and an emergency room data set containing 23,501 patient records. We use two evaluation metrics to measure the results:

Completeness: Defined as whether we can recover the original keywords of a topic by combining the keywords associated with each time segment. Specifically, for each topic segment, we retrieve the top 15 keywords. Then the keyword union is compared with the top 50 keywords derived by the original topic model (e.g., LDA). Then, the completeness measure is defined as the F-measure between these two sets of keywords. Given a set of time-sensitive keywords $ts^j = ts^j_1, ts^j_2, \ldots, ts^j_L$ of topic j with L as the number of topic segments, and keywords based topic summary S^j for topic j, the F-measure is calculated by:

$$F^j = \frac{2}{1/precision^j + 1/recall^j} \tag{1}$$

where

$$precision^j = \frac{|ts^j \bigcap S^j|}{|ts^j|}, recall^j = \frac{|ts^j \bigcap S^j|}{|S^j|}. \tag{2}$$

The overall completeness is computed by the mean and standard deviation of the F-measures of all the topics.

Distinctiveness: Defined as whether we can distinguish one topic segment from another based on their associated keywords to avoid redundancy. Here, we use the Kullback-Leibler (KL) divergence to measure the distinctiveness of two sets of keywords. Specifically, suppose we have L segments in topic j. Each segment has a normalized keyword histogram h^j_l, s.t. $\sum_i h^j_l(i) = 1$. The Kullback-Leibler divergence between h^j_l and h^j_m is

$$D_{KL}(h^j_l || h^j_m) = \sum_{i=1}^{V} h^j_l(i) \log \frac{h^j_l(i)}{h^j_m(i)} \tag{3}$$

Moreover, the symmetric Jensen-Shannon divergence is

$$D_{JS}(h^j_l || h^j_m) = \frac{1}{2} D_{KL}(h^j_l || \bar{h}^j) + \frac{1}{2} D_{KL}(h^j_m || \bar{h}^j) \tag{4}$$

where $\bar{h}^j = \frac{1}{2}(h^j_l + h^j_m)$. Thus, we define the distinctiveness of topic j as

$$D(\{h^j_l\}^L_l) = \frac{1}{L(L-1)} \sum_l \sum_m D_{JS}(h^j_l || h^j_m) \tag{5}$$

Then the overall distinctiveness is measured by the mean and standard deviation of the distinctiveness of all the topics.

The evaluation results for both the email and the emergency room data sets are shown in Tables 2 and 3. Here, the baseline system selected time-sensitive keywords based on

Table 2: Completeness and Distinctiveness Results on Email Data (mean±std).

	Completeness	Distinctiveness
Baseline	0.452 ± 0.043	0.182 ± 0.079
TIARA	**0.657 ± 0.055**	**0.315 ± 0.082**

Table 3: Completeness and Distinctiveness Results on Healthcare data (mean±std).

	Completeness	Distinctiveness
Baseline	0.578 ± 0.053	0.114 ± 0.087
TIARA	**0.740 ± 0.073**	**0.210 ± 0.058**

term frequencies while TIARA selected time-sensitive keywords based on Algorithm 4. We use the same time range splitting algorithm described in Algorithm 3. Here *mean* denotes the average and *std* represents the standard deviation of the *completeness* or *distinctiveness* values of K topics. As shown in the tables, TIARA outperformed the baseline system on both data sets. We also notice that the completeness of the time-sensitive keywords selected for the emails is lower than that for the emergency room records while the distinctness of the selected keywords for the emails is higher than that for the emergency room records. This seems to suggest that the topics derived from the emails evolve more quickly than those derived from the emergency room records.

6.2 Response Time

We also conducted experiments to examine the response time of TIARA. In particular, we recorded six different types of response time: (1) the total response time (*Total*); (2) the time spent on search (*Search*); (3) the time spent on extracting and parsing information from the Lucene index (*Parse*); (4) the time spent on generating keyword-based topic summaries, topic segmentation and time-sensitive keyword selection (*Keyword*); (5) the time spent on topic ranking (*Topic rank*); and (6) the time spent on generating JSON objects for topic visualization (*Json*). We asked two users to participate the study. They used TIARA in analyzing both personal emails and emergency room records. We wrote a time profiling program as a TIARA plugin. TIARA then automatically recorded different types of response time for each of the 100 queries issued by the users on each data set. The results are presented in Figs. 6 and 7. In the figures, the X-axis represents the number of documents that match a user's query, and the Y-axis encodes the response time (in milliseconds).

As shown in Figs. 6 and 7, overall, TIARA is capable of producing results with reasonable response time, especially when the number of returned documents is less than a thousand (in this case, the total response time is less than 300 ms). The most time-consuming procedures are *Search* and *Parse*. Since we use the Lucene search engine, the search time is pretty standard. However, the time spent on *Parse* can be significantly reduced if we optimize the current topic keyword indexing and parsing algorithm.

7. APPLICATIONS AND DEPLOYMENT

In this section, we present two application scenarios and our initial deployment of the current system.

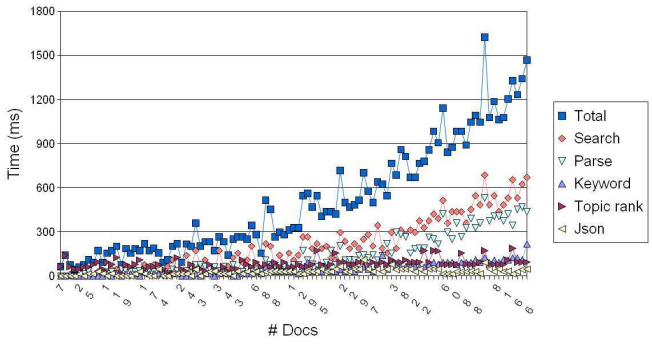


Figure 6: Response time on Email data.

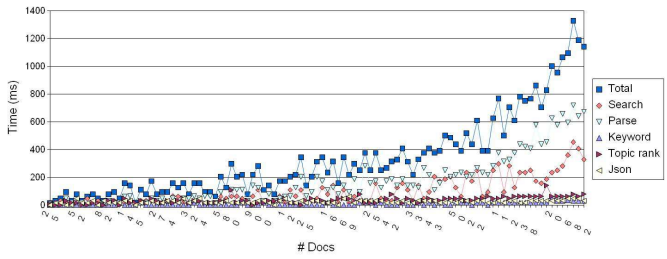


Figure 7: Response time on Healthcare data.

7.1 Applications

TIARA has been used in two real-world applications including email summarization and patient record analysis. Fig. 3 shows the TIARA user interface for the email application. Since we have discussed it in section 3, here we focus on the patient record analysis application. For more information about the email application, please see our video at `http://domino.research.ibm.com/comm/research_teams.nsf/pages/iva.download.html/`.

The data set we used in the patient record analysis application is NHAMCS (National Hospital Ambulatory Medical Care Survey) data. It contains sample cases in ambulatory care services provided by several hospitals' emergency and outpatient departments in the U.S. The current data set includes 23,000 patient records from 2002 to 2003. Fig. 8 shows its user interface. Here we show the top 8 out of 15 topics in total. Similar to the email application, a user may enter a query term to search the emergency room records (Fig. 8a). Since a patient record includes a few text fields (e.g., "diagnosis", "reason for visit" and "cause of injury") and a few structured fields (e.g., patient gender), a user may interact with the facet navigation panel (Fig. 8b) to select the information most relevant to their interests. In this case, the user is interested in the "cause of injury" and its relation to the patient's gender. The search results are then summarized in TIARA's main visualization view (Fig. 8c). Within the main visualization view there is a legend panel (Fig. 8d). It shows the main topics currently in display and the color coding scheme for the topic keywords (e.g., brown for female related keywords and dark green for male related keywords). When a user zooms into a specific topic by clicking on a topic layer (in this case, the user clicks

on the "ankle, basketball, twisted" topic), TIARA automatically subcategories the documents into two subsets based on the patient's gender and a set of time-sensitive keywords is extracted for each subset. As displayed in Fig. 8, visually, the selected topic layer is enlarged using a customized fish-eye distortion technique. The layer is also divided into two sub-layers to display keywords from each gender category. When we inspect the two sub-topics highlighted in yellow, some interesting patterns are revealed. The cause of injury for male patients tends to be sports-related, such as playing football or basketball; while for female patients, the injuries often occur when they perform routine activities, such as walking on a porch or stairs.

7.2 Deployment

Currently, there are two ways to deploy TIARA: as a web service or as a desktop application. To deploy TIARA as a web application, we hosted an application server using Tomcat as the web container for each application. Currently, two web services, one for email summarization and one for patient record analysis, have been set up and users can access TIARA via standard web browsers like Internet Explorer[10]. In this setting, users can only explore the data sets we provided.

To deploy TIARA as a desktop application, we implemented TIARA as a plugin for IBM Lotus Notes[11], an enterprise email solution developed by Lotus. The TIARA Notes Plugin was deployed at an IBM internal software hosting service site and it was made available to all the IBM employees world wide. After downloading and installing the TIARA Notes Plugin, users can directly launch TIARA in Notes to analyze their emails.

So far, 877 downloads have been recorded. We also received many comments from the users, such as, *"I was very impressed by the way it deals with un-structure data", "I like the evolution graph with tagclouds on it;", "The most impressive feature TIARA is its dynamic query and graphics rendering capability.", "TIARA visualization provides an quick overview of the documents being examined, which enables me to quickly find the active topics. It's cool!"*.

Furthermore, our previous studies [15] showed that TIARA is even more effective for business professionals. This observation has also been confirmed by the customer feedback collected by our product groups. It has been suggested that the tool is more effective for those who have some background knowledge on the data.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we present a novel visual exploratory text analytic system called TIARA that can help users rapidly view, explore, and analyze large collections of text. TIARA tightly integrates text analytics with interactive visualization to support effective exploratory text analysis. It also allows users to interact with a set of visualization widgets which help them comprehend and digest text summaries in context. We have applied TIARA to two real-world applications such as email summarization and visual patient record analysis. Our experimental results and initial user feedback received after its deployments show the effectiveness and applicability of TIARA.

In the future, in addition to the keyword summaries, we plan to add sentence-based summaries to TIARA to provide users with another alternative of analyzing text collections. Moreover, since the framework of the current TIARA system is easily extensible to multilingual documents, we are working on extending TIARA to support other languages (such as Chinese and Japanese). Finally, we plan to optimize the performance of the index parsing component to reduce the processing time and to make TIARA more responsive.

[10]Currently, we only provide access to our company employees because we can not host external servers.

[11]http://www-01.ibm.com/software/lotus/products/notes/

9. ACKNOWLEDGMENTS

The authors would like to thank their IBM colleagues in the creation of TIARA. We would like to thank Xiaoxiao Lian, Xiaohua Sun, and Kevin Brown for their work on the TIARA UI design. We would also like to thank Paul Taylor, Kenny Ng, Seiji Hamada, Hiroaki Kikuchi for their help in deploying TIARA and collecting feedback from customers.

10. REFERENCES

[1] D. Blei and J. Lafferty. *Topic Models*, chapter: Topic Models. Taylor and Francis, 2009. (in Press).

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[3] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):1–38, 2009.

[4] Y. Chen, L. Wang, M. Dong, and J. Hua. Exemplar-based visualization of large document corpus. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1161–1168, 2009.

[5] E. Clarkson, K. Desai, and J. D. Foley. Resultmaps: Visualization for search interfaces. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1057–1064, 2009.

[6] M. Dredze, H. M. Wallach, D. Puller, and F. Pereira. Generating summary keywords for emails using topics. In *IUI*, pages 199–206, 2008.

[7] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

[8] F. V. Ham, M. Wattenberg, and F. B. Viégas. Mapping text with phrase nets. In *InfoVis*, pages 1169–1176, 2009.

[9] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Trans. Vis. Comput. Graph.*, 8(1):9–20, 2002.

[10] T. Iwata, T. Yamada, and N. Ueda. Probabilistic latent semantic visualization: topic model for visualizing documents. In *KDD*, pages 363–371, 2008.

[11] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[12] W. Ke, C. R. Sugimoto, and J. Mostafa. Dynamicity vs. effectiveness: studying online clustering for scatter/gather. In *SIGIR*, pages 19–26, 2009.

[13] B. Lee, G. Smith, G. G. Robertson, M. Czerwinski, and D. S. Tan. Facetlens: exposing trends and relationships to support sensemaking within faceted datasets. In *CHI*, pages 1293–1302, 2009.

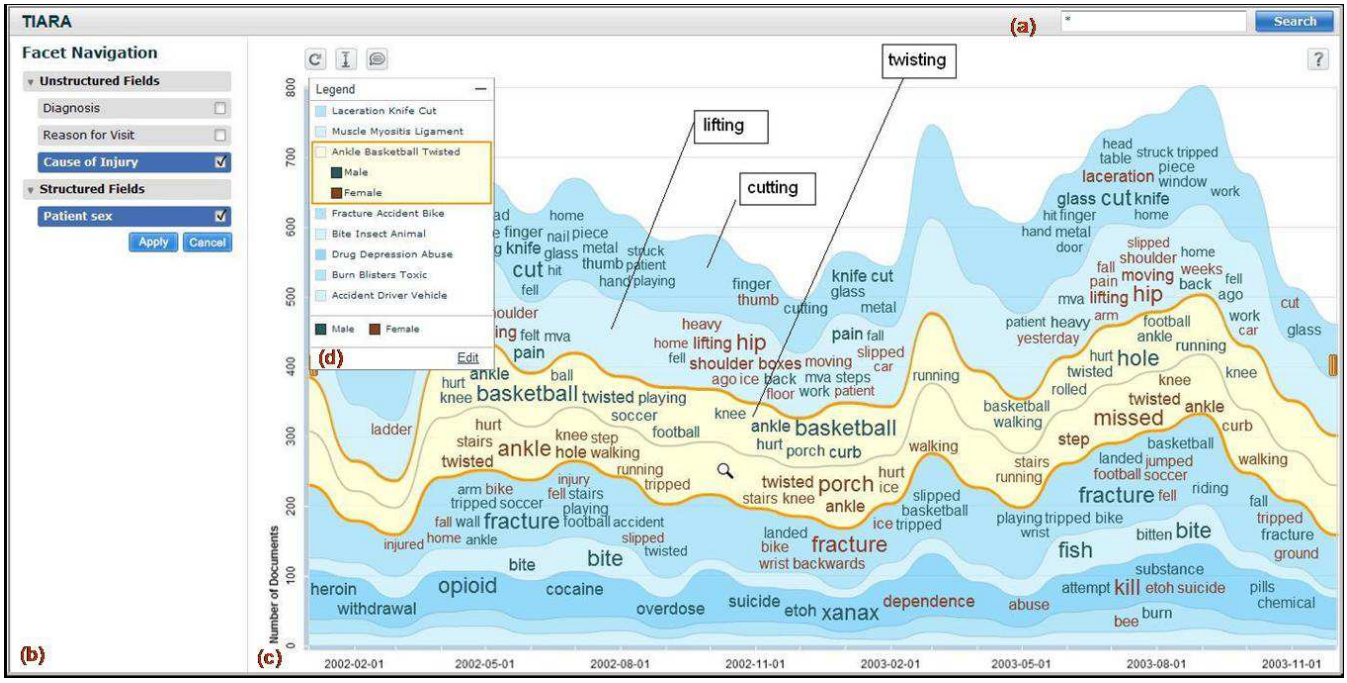


Figure 8: TIARA's visual summary of the "cause of injury" field of the 23,000$^+$ emergency room records from 2002 to 2003. We have previously shown the visual summary of the "reason for visit" in Fig. 1. In addition to visually summarizing the free-text fields (e.g. "cause of injury", "diagnosis" and "reason for visit" in the above figure), TIARA incorporates structured fields (e.g. patient gender) to explain the visual summary. As shown, the topics in the "cause of injury" field including "cutting", "lifting", "twisting", "fracturing", etc.

[14] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD*, pages 497–506, 2009.

[15] S. Liu, M. X. Zhou, S. Pan, W. Qian, W. Cai, and X. Lian. Interactive, topic-based visual text summarization and analysis. In *CIKM*, pages 543–552, 2009.

[16] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[17] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD*, pages 198–207, 2005.

[18] P. Pirolli, P. Schank, M. Hearst, and C. Diehl. Scatter/gather browsing communicates the topic structure of a very large text collection. In *CHI*, pages 213–220, 1996.

[19] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, USA, 1986.

[20] G. Smith, M. Czerwinski, B. Meyers, D. C. Robbins, G. G. Robertson, and D. S. Tan. Facetmap: A scalable search and browse visualization. *IEEE Trans. Vis. Comput. Graph.*, 12(5):797–804, 2006.

[21] Y. Song, S. Pan, S. Liu, F. Wei, M. X. Zhou, and W. Qian. Constrained co-clustering for textual documents. In *AAAI*, 2010.

[22] Y. Song, S. Pan, S. Liu, M. X. Zhou, and W. Qian. Topic and keyword re-ranking for LDA-based topic modeling. In *CIKM*, pages 1757–1760, 2009.

[23] X. Wang and A. McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *KDD*, pages 424–433, 2006.

[24] M. Wattenberg and F. B. Viégas. The word tree, an interactive visual concordance. In *InfoVis*, pages 1221–1228, 2008.

[25] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185, 2006.

[26] S. Xu, T. Jin, and F. C. M. Lau. A new visual search interface for web browsing. In *WSDM*, pages 152–161, 2009.

[27] J. Zhang, Y. Song, C. Zhang, and S. Liu. Evolutionary hierarchical Dirichlet processes for multiple correlated time-varying corpora. In *KDD*, 2010.

[28] C. Ziemkiewicz and R. Kosara. Preconceptions and individual differences in understanding visual metaphors. *Comput. Graph. Forum*, 28(3):911–918, 2009.