DI-DAP:An Efficient Disaster Information Delivery and Analysis Platform in Disaster Management

Tao Li[1,2], Wubai Zhou[1], Chunqiu Zeng[1], Qing Wang[1], Qifeng Zhou[3], Dingding Wang[4], Jia Xu[2]
Yue Huang[2], Wentao Wang[1], Minjing Zhang[1], Steve Luis[1], Shu-Ching Chen[1], Naphtali Rishe[1]

[1]School of Computing and Information Sciences
Florida International University
Miami, FL, USA 33199

[2]School of Computer Science and Technology
Nanjing University of Posts and Telecommunications
Nanjing, China 210046

[3]Automation Department
Xiamen University
Xiamen, China 361005

[4]School of Computer Science
Florida Atlantic University
Miami, FL, USA 33431

ABSTRACT

In disaster management, people are interested in the development and the evolution of the disasters. If they intend to track the information of the disaster, they will be overwhelmed by the large number of disaster-related documents, microblogs, and news, etc. To support disaster management and minimize the loss during the disaster, it is necessary to efficiently and effectively collect, deliver, summarize, and analyze the disaster information, letting people in affected area quickly gain an overview of the disaster situation and improve their situational awareness.

To present an integrated solution to address the information explosion problem during the disaster period, we designed and implemented DI-DAP, an efficient and effective disaster information delivery and analysis platform. DI-DAP is an information centric information platform aiming to provide convenient, interactive, and timely disaster information to the users in need. It is composed of three separated but complementary services: *Disaster Vertical Search Engine*, *Disaster Storyline Generation*, and *Geo-Spatial Data Analysis Portal*. These services provide a specific set of functionalities to enable users to consume highly summarized information and allow them to conduct ad-hoc geospatial information retrieval tasks. To support these services, DI-DAP adopts *FIU-Miner*, a fast, integrated, and user-friendly data analysis platform, which encapsulated all the computation and analysis workflow as well-defined tasks. Moreover, to enable ad-hoc geospatial information retrieval, an advanced query language *MapQL* is used and the query template engine is integrated.

DI-DAP is designed and implemented as a disaster management tool and is currently been exercised as the disaster information platform by more than 100 companies and institutions in South Florida area.

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983355

1. INTRODUCTION

Disaster such as hurricanes, earthquake, tsunami, and other natural calamitous events are destructive and devastating and can cause enormous physical damage, loss of life and property around the world [37]. An untold amount of time and effort was also spent on repeated preparation and recovery efforts causing fatigue and anxiety in the affected communities.

For over seven years, disaster management researchers at Florida International University have collaborated with the Miami-Dade Emergency Operations Center (EOC), South Florida Emergency Management and industry partners inluding Wal-Mart, Office Deport, Wachovia, T-Mobile, Ryder Sytems and IBM to develop prototype systems to help South Florida public and private sector entities manage and exchange information in a disaster situation.

According to our preliminary study in disaster management situations [14, 23, 28, 37, 36, 13], information plays a critical role. Indivisual, household, industry representatives, disaster management officials, and government agencies are eager to find valuable information to help them understand the current situation and recovery status and to timely respond to the situations for better prevention, preparation, response, and recovery. As a result, a critical task in disaster management is to effectively analyze and understand the disaster-related situation updates and to provide efficient support for uses to find the needed information at any time.

Many real-world systems, such as Web EOC [11], Ears [3] and E-Team [17] have been designed to provide effective information collection, reporting, and sharing. Other research efforts such as [10, 26, 8, 19, 29, 6, 16], use social media to support real-time information collection for fast response.

However these tools are essentially rooted in traditional databases and regard different information sources as independent, with little consideration on the dynamic and complex nature of disasters information [9]. For example, during a hurricane, unforeseen circumstances and unplanned complications always happen, and often require immediate action to minimize the risks and costs. For household owners, finding a shelter might not be in their original plan, but as the situation continuously worsens, they may want to reassess the relative risks of seeking a shelter or staying at

home. These risks are dependent upon the interaction of information such as dynamically changing weather, road, and traffic conditions and the proximity and availability of emergency service providers who can accommodate the refugees and assist with search-and-rescue operations. Because conditions are continuously evolving and information may come from various channels, we need intelligent techniques that summarize the updates of relevant information to help users improve situational awareness and disaster recovery.

1.1 Motivation for Disaster Information Delivery and Analysis Platform

Based on the long and fruitful cooperation with South Florida Emergency Management and industry partners [28, 37, 36, 13], we have the opportunity to gain insights into the needs and desire for individuals, households, industry representatives, disaster management officials, and government agencies. Furthermore, we realize that more powerful and efficient disaster management platform can be designed and implemented to greatly mitigate the disruption caused by disasters, with the rapid development of data mining techniques and easy access to heterogeneous data resources such as social documents and geospatial data. The emergency managers and business continuity professionals eagerly desire that a powerful and intelligent data analysis platform specifically be designed for disaster management domain and aim for efficient information acquisition from heterogeneous data resources including news/articles/blogs from web, announcements from governments, business reports from company, social network microblogs and geospatial dataset on areas affected by the disaster, such as shelters' location, house price change and criminal rate change. They agreed that such a system can significantly help them facilitate their disaster management and reduce recovery efforts. In order to efficiently and effectively deliver high-quality disaster related information, several challenging yet desirable information management issues have been brought up.

1. Real-time disaster information. Timeliness is the most crucial and critical requirement for information delivery and analysis in disaster management. Out-dated information might mislead users, and put users in more dangerous situation and cause severe consequences. Moreover, people always prefer the latest news and situation reports related to their search interests when being affected by disasters or not.

2. Heterogeneous information resources. News portal is no longer the only information source in disaster situation. Microblogs or social network applications make fast information dissemination and response. Information from different channels can largely accelerate the information discovery process and thus high quality service can be delivered.

3. Information summarization. Heterogeneous information resources provide us the access to a myriad of disaster-related documents. Meanwhile, they cause the so-called *information overload* problem. Information summarization alleviates this problem to some extent by compressing a given collection of documents into a concise summary.

4. Diverse Information presentation. With development of advance technologies, presenting textual results are no longer effective for information delivery. Advanced visualization methods focusing on combining different aspects or dimensions become more popular. Additionally, in disaster management, geolocation is considered as one of the most important characteristics, which can greatly help improve situation awareness.

To sum up, users expect that the disaster management platform is able to assimilate massive and heterogenous information sources and intelligently deliver the most representative nuggets for decision support. Besides, Users often prefer a disaster management platform as a service to provide them the privilege to run customized tasks and obtain insights of disaster situation.

1.2 Proposed Solutions

To address aforementioned challenging issues, we identify the following four key tasks to fully utilize the advantages of heterogeneous information resources.

1. Design and develop effective and dynamic concept hierarchy generation and reuse methods in disaster management domain to help domain experts, the crawler and search engine perform efficiently. By formalizing and sharing the knowledge, concept hierarchy supports domain experts and knowledge engineers for modeling data and concept modeling and can help implement intelligent system. However, building the hierarchy from the scratch is expensive and requires lots of huge human efforts. Iterative concept hierarchy generation and reuse becomes a challenging but essential task.

2. Design and develop intelligent focused web crawling techniques to manage the data acquiring process and to increase the information coverage and relevance. Utilizing general search engines to collect heterogeneous data sources often generate too many irrelevant documents and unexpected amounts of URL seeds. Intelligent crawling strategies are thus needed to guarantee the high quality and relevance of indexed web contents, particularly in domain-specific tasks.

3. Design and develop disaster event summarization techniques for identifying disaster evolutionary path. In disaster management, people are interested to know the development of the events as well as the changes of different phases as the event evolve over time. However, traditional research efforts on document summarization focus on generating a compressed summary delivering major information of original documents [15, 20, 24, 13]. In disaster management, we aim to integrate text, temporal information, and spatial information and generate storyline-based summaries to reflect the evolution of the given disaster.

4. Design and develop effective techniques to support flexible geo-spatial data analysis. To support flexible geo-spatial data analysis, a SQL-like language *MapQL* is designed in our system. However, it is not easy for non-professional users to compose a MapQL query for performing analysis tasks. Therefore, automatic query template generation is needed to perform and optimize data analysis.

In this paper, we design and implement DI-DAP, an effective disaster information delivery and analysis platform, to provide an integrated solution. DI-DAP is an information centric information platform aiming to provide convenient, interactive, and timely disaster information to the users in need. Generally, DI-DAP utilizes the latest advances in database, data mining, and information extraction techniques to accomplish the above tasks and deliver a user friendly, information-rich service in disaster management domain. In particular, document hierarchical clustering is

applied to automatically enrich and refine the existing concept hierarchy to address the iterative concept generation task. A focused crawler is developed to assign newly discovered web resources to different concepts and discover new concepts simultaneously to address the data crawling issue. To address disaster document summarization issue, we develop intelligent techniques to generate a storyline providing a sketch of the disaster evolution containing textual, temporal and spatial information. For template generation, we use sequential pattern mining to recommend query patterns for users.

1.3 Roadmap

The rest of the paper is organized as follows. Section 2 will give the system overview of DI-DAP, including the *Disaster Information Delivery Layer*, the *Data Manipulation Layer*, and the *Distributed Storage Layer*. Then, three major services provided by DI-DAP will be described in Section 3, Section 4, and Section 5, respectively. Having introduced the services, we will report the empirical study of DI-DAP in Section 6. Finally, the conclusion and future work of this paper will be introduced in Section 7.

2. SYSTEM OVERVIEW

From the architecture perspective, DI-DAP consists of three layers: *Disaster Information Service Layer*, *Data Manipulation Layer*, and *Distributed Storage Layer*. In the following, we will briefly introduce each of the layers.

2.1 Disaster Information Service Layer

As shown in Figure 1, *Disaster Information Service Layer* consists of three functional services to facilitate disaster information delivery: the *Disaster Information Retriever*, the *Disaster Storyline Visualizer*, and the *Geo-Spatial Data Analysis Portal*.

The *Disaster Information Retriever* is specifically designed for providing in-depth domain information about the disasters. Users can use it to quickly obtain relevant disaster information and the results will be plotted on the the map for reference.

The *Disaster Storyline Visualizer* is a visualization tool that helps the users to summarize a large number of disaster related documents into an interactive map-based summary. Due to large number of related documents, blogs, and news, it is prohibitive for users to read all the materials. Using the *Disaster Storyline Visualizer*, users can easily grasp the main idea of the disaster by reviewing the highly summarized events plotted on the map.

The *Geo-Spatial Data Analysis Portal* is an advanced portal that enables users to conduct complex SQL-like geographical queries to retrieve information relevant to the disaster and perform geo-spatial data analysis. In our system, the geo-spatial data analysis is powered by TerraFly Geocloud [33]. The portal is available at http://geocloud.cs.fiu.edu/. If a user wants to retrieve geo-spatial related information, they can leverage the *MapQL* to conveniently and interactively obtain the data needed.

2.2 Data Manipulation Layer

The *Data Manipulation Layer* is the logic centerpiece of DI-DAP. In this layer, there are four main components including *Vertical Search Engine*, *Storyline Generation Engine*, *MapQL Template Engine* and *FIU-Miner* to support the data manipulation requests from the top layer.

The *Vertical Search Engine* offers two functionalities to support the *Disaster Information Retriever* on the top layer for retrieving disaster information. One is the *Vertical Search Engine Crawler* and the other one is the *Taxonomy Generator*. The disaster taxonomy built by the *Taxonomy Generator* guides the *Vertical Search Engine Crawler* to crawl disaster specified information from Internet.

The *Storyline Generation Engine* generates a storyline related to a specified disaster for the *Disaster Storyline Visualizer*. The storyline generation is accomplished by two modules, i.e., the *Event Extractor* and the *Storyline Generator*. The *Event Extractor* extracts events from crawled documents by the *Vertical Search Engine Crawler*. The *Storyline Generator* summarizes and organizes the extracted events into a meaningful storyline.

The *MapQL Template Engine* is composed of the *MapQL Executor* and the *MapQL Query Template Generator*. The *MapQL Executor* is used to parse and execute the *MapQL* scripts and then retrieve the relevant data (including geospatial data and aerial photo data) to the *Geo-Spatial Data Analysis Portal*.

The *FIU-Miner* [30] is a Fast, Integrated, and User-friendly system to ease data analysis. It is a general computing platform that can be applied to various application scenarios [38, 32]. *FIU-Miner* allows users to rapidly configure a complex distributed computing task without much effort. In DI-DAP, the computing procedure of the vertical search engine indexing and retrieval, taxonomy generation, event extraction, summarization, query execution and storyline generation [39] are all configured as *FIU-Miner* computing tasks. As all the computation can be conducted offline, *FIU-Miner* treats them as batch-based distributed computing tasks. Besides these service oriented computing tasks, *FIU-Miner* is also used to conduct tasks like query log learning and pattern mining, which can be used to help improve the user experience for the usage of Geo-spatial analysis portal.

2.3 Data Storage Layer

Data Storage Layer is the lowest level of DI-DAP. It consists of several kinds of distributed data storage repositories, including *HDFS*, *Distributed Geo-spatial Database*, and *Distributed Aerial Photography Database*.

HDFS is in charge of storing all the documents, intermediate data, and mined results of the upper level modules. In *FIU-Miner*, most of the configured computing tasks would load and save the data to *HDFS*. *Distributed Geo-spatial Database* is the distributed repository that is specifically used to store the map related data, such as the latitude, longitude, and water level of all the point of interests (POIs). Also, the associated meta-data, such as the criminal rate, the house price of the POIs are also stored. *Distributed Aerial Photography Database* is mainly used to store the visual part of *Geo-Spatial Data Analysis Portal* service. As there are a large number of aerial photos, it is necessary to use a particular database for storage.

3. DISASTER VERTICAL SEARCH ENGINE

In DI-DAP, to effectively collect data from heterogenous sources, we develop an intelligent vertical search engine to crawl domain specific documents guided by a domain taxonomy.

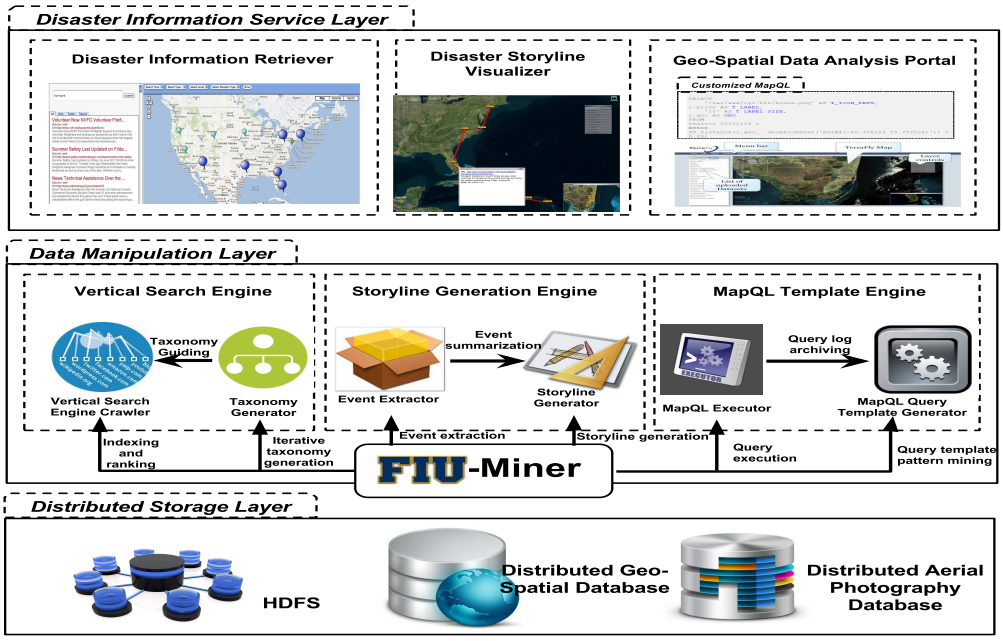


Figure 1: Overview architecture of DI-DAP

3.1 Taxonomy Hierarchy Generation

Based on our long cooperation with Miami Dade Emergency Operational Center (EOC), we extracted hundreds of frequent terms in its official announcements and situation reports over the past 5 years. We assume that those terms with high frequency indicate important concepts in disaster domain. Through careful filtering and organizing those terminologies by our staff and developers, our initial disaster taxonomy is obtained and then verified by our domain experts.

The disaster taxonomy generation process follows an interactive and iterative strategy. The focused crawler utilizes the taxonomy to classify accessed web pages and prioritizes those pages with highest relevance to disaster domain. From the repository of crawled data, high quality data will be analyzed and disaster-related concepts that have not been mentioned in the existing taxonomy, will be extracted. Those extracted concepts are considered as highly popular terms. They will be used to extend and enrich the existing taxonomy.

All concepts in the existing taxonomy are denoted as $T = \{t_1, t_2, ..., t_n\}$ and the newly-discovered concepts are denoted as $C = \{c_1, c_2, ..., c_m\}$. H is the existing concept hierarchy formed by terms from T. Our goal is to generate an updated concept hierarchy $H^{'}$ that is formed by all terms from both T and C. The integration of T and C is non-trivial. There are three important aspects that need to be pointed out: a) each concept in T or C is represented by a set of terms extracted from the web documents repository. So, essentially there is a subset of web documents under each concept; b) H is essentially a hierarchical clustering on all documents. The hierarchy of the concepts reflects the inclusion or exclusion of documents sets. There is no partial overlap between document sets under different concepts; c) there is a merging preference/order for each pair of concepts in both H and $H^{'}$ which indicates the level of closeness between two document sets. The new concepts in C should not change the relative merging order of existing concepts in T.

The merging preferences mentioned above are modeled as relative order constraints when performing hierarchical clustering on. Several types of constraints can be applied in hierarchical clustering and the must-link-before (MLB) constraint, proposed in Bade's algorithm, is used in our work [4, 35, 34].

3.2 Focused Crawler

Focused crawling techniques [5] are applied to build a focused crawler aiming to retrieve the disaster related documents from the Web. Besides, some local news feeds and announcements from government sites are also downloaded.

Standard focused crawlers use best-first approaches as the selection strategy to select the next page to be crawled out of all currently candidate page URLs, according to the score defined as

$$l^* = \arg\max_{l \in queue} score(l),$$

where $score(l)$ indicates the probability that the URL l belongs to the topic calculated by a classifier. However, due to the imbalance of subtopics and a limited initial training dataset, the classifier may bias toward some subtopics of general disaster topics. To enrich the diversity of crawled web pages for a specific disaster, we use the following selection strategy

$$l_C^* = \arg\max_{l \in queue} score(l, C). \quad (1)$$

For each disaster concept C, the crawler crawls next page from candidate page URLs ranked by the scores with respect to the concept.

Furthermore, to prioritize the URL l linked from page $page_l$, we calculate the *prioritization score* as follows:

$$score(l, C_d) = P(C_i^* \rightarrow C_d) * P(page_l = C_i^*),$$

where $P(page_l = C_i^*)$ is the output of our content classifier calculated by Equation 1 indicating the probability that the page $page_l$ belongs to its optimal concept C_i^*. $P(C_l \rightarrow C_d)$

is the probability that a page of concept C_i links to a page of concept C_d. It can be calculated as

$$P(C_i \to C_d) = \frac{\sum_{p \in C_i} |L_{p,fetched} \cap C_d| + \lambda}{\sum_{p \in C_i} |L_{p,fetched}| + \lambda \sum_{p \in C_i} |L_{p,unfeteched}|},$$

i.e., the ratio of the number of links classified as C_d from pages of C_i to the totally fetched links from page of C_i, with a Dirichlet smoothing using un-fetched links. $P(C_i \to C_d)$ is updated along with the process of crawling.

Although a page is disaster related, the links of the page may lead to other pages irrelevant to disaster. We observe that for a pair of links which are in the sibling nodes of the HTML DOM tree (e.g., in a list of page), they tend to relate to similar topics. Thus, we build a link classifier following the work [2] based on Naive Bayes. After applying link prediction, the prioritization score is extended to

$$score(l, C_d) = P(C_i^* \to C_d) * P(page_l = C_i^*) * P(C_d|l),$$

where $P(C_d|l)$ is the output of the link classifier (i.e., the probability that link l leads to a page under concept C_d).

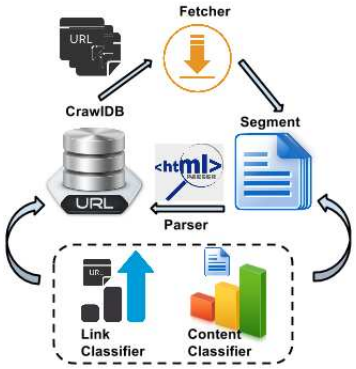


Figure 2: Architecture of the focused crawler

Figure 2 shows the architecture of a focused crawler. The *fetcher* fetches page content of a list of URLs and stores them as a segment. The *Parser* parses them and updates CrawlDB, where the crawled data is associated with a URL. The scoring module assigns a prioritization score to each URL indicating the importance of the URL. Based on the scores, a set of URL, covering all disaster concepts in the concept hierarchy, is delivered to *Fetcher* for fetching.

4. DISASTER STORYLINE GENERATOR

A disaster storyline describes how the disaster evolves over time along the location attribute of the *events* and how the disaster affects different areas indicated by the description attributes. *Event* here denotes a tuple (t, l, s) where t is the time that an event occurs, l is the event location and s is the textual description about the event. For example, (08/27/2011, New York City, "The five main New York City-area airports will be closed to arriving flights") represents an event in Hurricane Sandy.

In disaster management, people are interested to know the development of the events as well as the changes/differences of different phases as the events evolve over time. However, traditional research efforts on document summarization focus on generating a compressed summary delivering the major (or query-relevant) information of the original documents while news topic detection and tracking systems usually aim at grouping news articles into a cluster to present an event in a topic and monitor future events related to the topic. As a result, their systems mainly focus on highlighting and summarizing events in a topic and lack of the theme structure to capture the event evolution [21, 7, 24, 12, 13, 27]. In our system, we propose to generate a storyline containing textual, geo-spatial and structural information to provide a sketch of the event evolution. Different from existing work on temporal summarization and timeline generation, we aim to integrate text, geo-location and temporal information, and generate storyline-based summaries to reflect the evolution of the given topic.

In particular, the problem of generating a storyline can be defined as follows:

Input: A collection of documents related to a disaster.

Output: A storyline consists of the most representative events summarizing the evolution of disaster-relevant topics. It is a chain of events $(o_1, \ldots, o_n)$, as the global temporal and spatial evolution of a disaster.

The chain structure is used under the assumption that a disaster at any time should have only one geo-spatial center, which should move continuously over time. Such an assumption is valid for most of the natural disasters like hurricanes, storms, and blizzards, but not for the man-made disasters like cyber attacks.

Here, we briefly show how to extract *events* given disaster-related corpus and them propose a new approach based on graph algorithms and linear programming.

4.1 Event Extraction and Event Graph Construction

An event extracted by our *Event Extractor* simply using rule-based entity recognition techniques, in which we select those text snippets from disaster specific documents containing an entity type {*date, location*}. Although each text snippet can be considered as an event, many of those are redundant. To remove the redundancy and obtain a set of representative events, we construct a graph $G = (V, E)$ with the given text snippets as the vertex set V, and add an edge between each pair of snippets which are likely to refer to the same event. Specifically, for obtaining the similarity of two nodes $v_i, v_j \in_{i \neq j} V$, we first calculate the lexical similarity between two text snippets, and then convert them into two feature vectors for computing the context similarity. By combining two parts, we use $sim(v_i, v_j)$ to represent the similarity between two nodes $v_i, v_j \in V$ shown below:

$$sim(v_i, v_j) = \underbrace{lex\_sim(v_i, v_j)}_{lexical\ similarity} \times \underbrace{cont\_sim(v_i, v_j)}_{context\ similarity},$$

where $lex\_sim(\cdot)$ is defined as a binary function representing whether a text snippet v_i is an abbreviation, acronym, part of another text snippet v_j, or if the character edit distance between the two nodes is less than a threshold θ[1]:

$$lex\_sim(v_i, v_j) = \begin{cases} 1 & v_i(v_j)\ is\ part\ of\ v_j(v_i), \\ 1 & EditDist(v_i, v_j) < \theta, \\ 0 & Otherwise. \end{cases}$$

We define $cont\_sim(\cdot)$ of two nodes as the cosine similarity between their context vectors $\vec{c_i}$ and $\vec{c_j}$. Note that on the text-snippet stream, two temporally distant snippets can be very different even though they are lexically similar.

[1] θ was empirically set as $0.2 \times min\{|v_i|, |v_j|\}$

We therefore restrain the context to a segment of the text-snippet stream $|S_k|$, where $\max_{\forall v_{ik}, v_{jk} \in S_k} \{|v_{ik}.date - v_{jk}.date|\}$ is less than a threshold δ [25], and then take the weighted average of the segment-based similarity as the final context similarity. To build the context vector, we use term frequency (TF) as the term weight and remove all the stopwords.

$$cont\_sim_{|S_k|}(v_i, v_j) = cos(\vec{c_i}, \vec{c_j}),$$

$$cont\_sim(v_i, v_j) = \sum_k \frac{|S_k|}{|V|} \times cont\_sim_{|S_k|}(v_i, v_j).$$

Here, $e_{ij} = (v_i, v_j) \in E$ if and only if both the similarity of v_i and v_j is greater than a similarity threshold parameter α, and their distance calculated by their geocodes is less than a distance threshold parameter *radius*. Note that the latter constraint takes the spatial smoothness of events into consideration.

4.2 Storyline Generation via Linear Programming (LP)

We identify the set of representative events in the original snippets with minimum redundancy by solving the minimum dominating set problem. A vertex u of a graph dominates another vertex v of the graph, if u and v are joined by an edge in the graph. A subset of S of the vertex set of an undirected graph is a dominating set if for each vertex u, either u is in S or a vertex in S dominates u. The *Minimum Dominating Set* (MDS) problem is to find a dominating set with minimum size. MDS has been previously used to model multi-document summarization problem [24]. In our case, we use the MDS of text snippets to capture the representative events from the text snippets of disaster event descriptions.

Having the dominating set of $G(V, E)$, m text snippets $d_1, \ldots, d_m$, as the representative events. Without loss of generality, the set of events are assumed to be in chronological order. To generate a storyline capturing the major location change of the disaster, we select a sequence of nodes $o_1, o_2, \ldots, o_l$ from the representative events in chronological order. Intuitively, the generated storyline should also be in spatial coherence, reflecting the continuous location change of the disaster over time. Since a disaster is likely to affect adjacent areas in a similar fashion, the storyline should be coherent in content as well.

Based on the above discussions, we model the storyline generation problem using integer linear programming. To select a chain of nodes from $d_1, \ldots, d_m$, we use variables $node\text{-}active_i \in \{0, 1\}, i = 1 \ldots m$ to indicate whether d_i is included in the selected chain, and $next\text{-}node_{i,j} \in \{0, 1\}, i, j = 1 \ldots m \text{ and } i < j$ to indicate that d_i and d_j are two successive nodes (i.e., a transition) in the chain. The objective function aims to maximize the storyline's content coherence which is defined as the minimal similarity between two successive nodes along the storyline as shown below:

$$Coherence(o_1, o_2, \ldots, o_n) = \min_{i=1,2,\ldots,n-1} sim(o_i, o_{i+1}).$$

We further impose the following set of constraints to model storyline's spatial coherence.

Chain Constraints: The consistency of variables $node\text{-}active_i$ and $next\text{-}node_{i,j}$ should be guaranteed by Formula 2 and 3, and the selected nodes should compose a chain in chronological order constrained by Formula 4 and 5.

$$\sum_i node\text{-}active_i - \sum_{i,j} next\text{-}node_{i,j} = 1. \quad (2)$$

$$\forall_{i<k<j} : next\text{-}node_{i,j} \leq 1 - node\text{-}active_k. \quad (3)$$

Clearly, Formula 2 indicates the number of *active transitions* is equal to the number of *active nodes* minus 1, and Formula 3 constrains a transition of two nodes cannot be active if there exists an active node between them.

$$\forall_{j(i)} : \sum_{i(j)} next\text{-}node_{i,j} \leq node\text{-}active_{j(i)}. \quad (4)$$

$$\forall_{i>j} : next\text{-}node_{i,j} = 0. \quad (5)$$

Length Constraints: The selected chain should be in a reasonable length ranged between pre-defined minimum length threshold $\mathcal{L}_{min}$ and maximum length threshold $\mathcal{L}_{max}$.

$$\mathcal{L}_{min} \leq \sum_i node\text{-}active_i \leq \mathcal{L}_{max}.$$

Location Smoothness Constraints: We require both pairwise and triple-wise smoothness to location change on the selected chain by satisfying Formula 6 and 7, respectively. Let $\mathcal{D}_{i,j}, i, j = 1, \ldots, m$ be the distance based pairwise location relationship between d_i and d_j, and $\mathcal{D}_{i,j} = 1$ if distance between d_i and d_j is less than a pre-defined distance parameter, $\mathcal{D}_{i,j} = 0$ otherwise. For triple-wise smoothness, let $\mathcal{A}_{i,j,k}$ be the angle based triple-wise location relationship, and $\mathcal{A}_{i,j,k} = 1$ indicates the angle constructed by three successive nodes d_i, d_j and event k is not an acute one, otherwise $\mathcal{A}_{i,j,k} = 0$. By not including in the chain three successive nodes of which the angle is acute, we excludes the back-and-forth events from the storyline and smooth the location change.

$$\forall_i : \sum_j (1 - \mathcal{D}_{i,j}) \cdot next\text{-}node_{i,j} \leq 0. \quad (6)$$

$$\forall_{i,j,k} : next\text{-}node_{i,j} + next\text{-}node_{j,k} \leq 1 + \mathcal{A}_{i,j,k}. \quad (7)$$

Formula 6 defines the pairwise smoothness constraint (i.e., the distance of two successive nodes should be within some range) and Formula 7 defines the triple-wise smoothness constraint (i.e., three successive nodes cannot construct an acute angle).

Minimal Similarity Constraints: Let $\mathcal{S}_{ij}, i, j = 1 \ldots, m$ be the cosine similarity between d_i and d_j. we can use the following constraints to find the similarity of the minimum similar transition *min-edge* among active transitions.

$$\forall_{i,j} : min\text{-}edge \leq 1 - (1 - \mathcal{S}_{i,j}) \cdot next\text{-}node_{i,j}$$

The Objective Function: Besides to maximize minimal similarity between two successive nodes along the storyline, we also try to make storyline as long as possible, so the objective function has the following form:

$$\text{Maximize: } min\text{-}edge + \lambda \cdot l,$$

where λ is a coefficient parameter. Although integer linear programming is an NP-hard problem, there are efficient approximation algorithms and implementations such as IBM CPLEX[2], which is used for optimization in this paper.

[2]http://www.ibm.com/software/commerce/optimization/cplex-optimizer/

5. DISASTER ANALYSIS OPTIMIZATION

As discussed in the system overview, Geo-spatial data analysis portal enables users to conduct complex SQL-like geographical queries expressed in MapQL statements to retrieve disaster information. Although MapQL is powerful and flexible to satisfy the disaster analysis requirement for the users, it requires the users to compose the statements, typically from scratch. As a consequence, it poses a big challenge for the end users, especially for those without any SQL knowledge background, to accomplish their disaster analysis tasks. To bridge the gap between our system and the end users, sequential query patterns are extracted from the user query logs and used for both query template construction and disaster analysis optimization [31].

5.1 Sequential Query Pattern

5.1.1 Mining Sequential Query Pattern

A typical disaster analysis task involves a sequence of MapQL statements which are produced during the end users' interaction with the system. A snippet of MapQL query logs is displayed in Figure 3. Each MapQL statement is associated with a user session ID and a time stamp. All the statements are organized in temporal order. Those MapQL statements sharing the same session ID are those issued by a user within a session. Our goal is to discover interesting patterns from the query logs. For example, according to the log data in the Figure 3, an interesting pattern is that users who viewed a particular street are more likely to look for the hotels along that street.

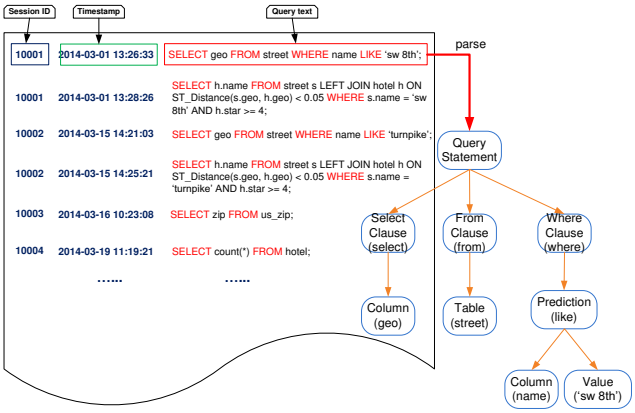


Figure 3: A snippet of MapQL query logs and an example of a syntax tree for a MapQL statement

However, it is difficult to discover meaningful sequential query patterns by directly applying existing sequential pattern mining algorithms to the raw logs of MapQL statements since representing a query item by the text in an MapQL statement is often too specific. Therefore, the original MapQL text is parsed into a syntax tree. As shown in Figure 3, each node in the example syntax tree has two labels. One describes the type of the syntax node, such as "Column", while the other label denotes the text content, such as "geo". Provided with the syntax tree, the MapQL query can be generalized by representing any nodes with their types instead of their actual contents.

Formally, let D be a collection of sequences of queries, denoted as $D = \{S_1, S_2, ..., S_n\}$, where S_i is a sequence of queries occurring within a session, ordered according to their timestamps. Therefore, $S_i =< q_1, q_2, ..., q_i, ..., q_m >$ is a sequence including m queries in temporal order. If q_i is a compound query composed of two sub-queries q_{i0} and q_{i1}, then $S_i =< q_1, q_2, ..., (q_{i0}, q_{i1}), ..., q_m >$. Sub-queries in a parenthesis are from a compound query occurring at the same timestamp.

A k-subsequence of S_i is a sequence of queries with length k denoted as $T =< t_1, t_2, ..., t_k >$, where each $t \in T$ corresponds to only one query $q \in S_i$, and all the queries in T are kept in temporal order. $T \sqsubseteq S_i$ is used to indicate that T is a subsequence of S_i.

Given the query sequence data collection D, a sequential query pattern is a query sequence whose occurrence frequency in the query log D is no less than a user-specified threshold $min\_support$. Formally, the support of sequence T is defined as

$$support(T) = |\{S_i | S_i \in D \wedge T \sqsubseteq S_i\}|.$$

A sequence T is a sequential query pattern, iff $support(T) \geq min\_support$.

The process of discovering all the sequential query patterns from the MapQL logs generally consists of two stages. The first stage is to generalize the representation of MapQL statements by parsing the MapQL text into syntax units. Based on the syntax representation of MapQL statements, the second stage is to mine the sequential query patterns from the sequences of MapQL statements.

According to the properly generalized representation of a MapQL query, the PrefixSpan algorithm [18] is applied to efficiently discover all the sequential query patterns from the MapQL query log data.

5.1.2 Query Template

Query template is generated by MapQL Query Template Engine in the system. This function alleviates the burden of users since MapQL queries can be composed by rewriting query templates. A query template is generated based on the discovered sequential query patterns. The syntax trees in the sequential query pattern are scanned and all the specific table, column and constant values are replaced with their corresponding template parameters. It guarantees that the same table, column or constant value appearing at multiple places, even multiple queries of a sequence acquires the same template parameter. Users can easily convert the template to executable queries by assigning the template parameters with specific values.

```
Template(#arg1#, #arg2#, #arg3#):

SELECT      geo
FROM        street
WHERE
            name LIKE #arg1#;

SELECT      h.name
FROM        street s
LEFT JOIN   hotel h
ON          ST_Distance(s.geo,h.geo) < #arg2#
WHERE       s.name LIKE #arg1#
AND         h.star >= #arg3#;
```

Figure 4: Example of a generated template

Given a sequential query pattern that contains the two queries with session ID 10001 in Figure 3, It can generate the template for the sequential query pattern. The generated template is shown in Figure 4. This template has

three parameters (i.e., #arg1#, #arg2#, #arg3#). Provided with values of these parameters, the executable sequence of queries can be easily derived from the template.

5.1.3 *Analysis Workflow*

All the MapQL queries in a sequential pattern are organized in a workflow, where the template parameters indicate the data transmission between queries. A sequence of queries constitutes a disaster analysis task and a typical disaster analysis task often involves a few sub-tasks. The dependencies among those sub-tasks make spatial data analysis very complicated. The complexity of disaster analysis dictates the support of workflow.

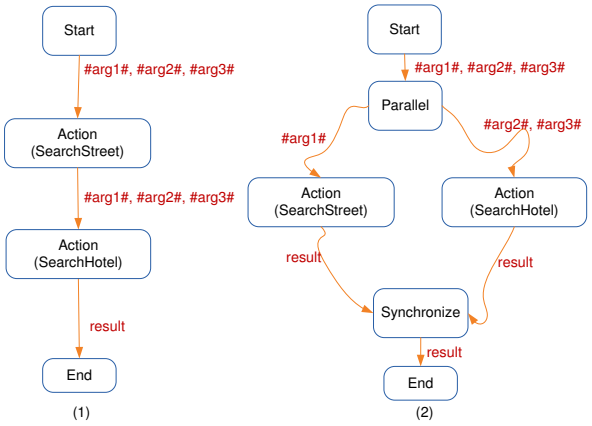


Figure 5: Workflow examples

Based on the generated template in Figure 4, two simple workflows can be constructed in Figure 5. These two workflows accomplish the same spatial data analysis task described in Figure 4. In the subfigure (1) of Figure 5, the two sub-tasks (i.e., SearchStreet and SearchHotel) are executed sequentially. However, SearchStreet needs the template parameter #arg1# as its input, while SearchHotel needs all three parameters. Provided with the three parameters, both sub-tasks can be executed independently. Thus, in the subfigure(2) of Figure 5, a parallel workflow is introduced to complete the spatial data analysis task. Since our data analysis tasks are scheduled by FIU-Miner, which takes full advantage of the distributed environment, the parallel workflow is more preferable to our system in terms of efficiency.

6. EMPIRICAL STUDY

In this section, the empirical study is conducted to demonstrate the effectiveness and efficiency of our system.

6.1 Data Collection

In addition to the available data sources from the Miami-Dade Emergency Operation Center (EOC), South Florida Emergency Management and industry partners, we also came up with a list of websites as the seeds for our disaster crawler. The websites are organized in a hierarchy based on our domain expertise. In general, there are four types of resources: jurisdictional, media, non-governmental organizations (NGOs), and private sectors. For each type, the resources are organized with several levels of administrative division and different categories according to their types of businesses. A snapshot of our site list is shown in Table 1 where a set of refined url seeds are used to cover the areas that hurricane Irene passing-through in late August, 2011. With respect to geo-spatial dataset, TerraFly Geocloud has a rich collection of GIS datasets [33], which includes the US and Canada roads data, the US Census demographical and socioeconomic data, the property lines and ownership data of 110 million parcels, various public place datasets, Wikipedia, extensive global environmental data, etc. In addition, users can also upload customized datasets for spatial data analysis and visualization by manipulating MapQL language seen in Figure 6.

Name	URL
1	http://www.fema.gov
2	http://www.fema.gov/hazard/index.shtm
3	http://www.fema.gov/news/recentnews.fema
4	http://www.fema.gov/photolibrary/index.jsp
5	http://www.disasterassistance.gov/
6	http://www.sba.gov/
7	http://www.redcross.org/
8	http://www.noaa.gov/
9	http://www.csc.noaa.gov/csp/
10	http://www.nhc.noaa.gov/
11	http://www.nccrimecontrol.org/
12	http://www.ct.gov/demhs/site/default.asp

Table 1: Crawling seeds used for the focused

Figure 6: A query example using MapQL in Terrafly.

6.2 System Evaluation

This section consists of two parts. The first part describes practical exercises conducted on our developed platform and the other part showcases the case study on our platform.

6.2.1 *Practical Exercise*

Our system evaluation process consists of presenting the system to our community of emergency managers, business continuity professionals and other stakeholders for feedback and performing community exercise. The community exercises involve a real time simulation of a disaster event and are integrated into an existing exercise that the community conducts for readiness each year. This evaluation exposes information at different time intervals and asks the community to resolve different scenarios by using the tool developed. The evaluation, conducted by Miami-Dade Emergency Management with the 2014 Hurricane Exercise, takes on the form of "table-top" exercise in which information injects provide the details about the current disaster situation and specify potential goals and course of action. In return, the participant uses the system to gather information to best access the situation and provide details about the actions they will take. We gather information from the users about what information they found to derive their conclusions or lack thereof. These information allows us to better understand how those techniques overall improve the information effectiveness.

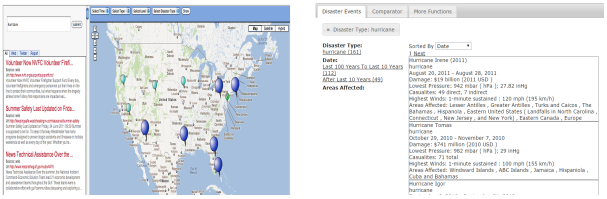
Feedbacks from our users are overwhelming positive and suggest that our system can be used not only to share the valuable actionable information but to perform more complex tasks like business planning and decision making. There

are also many collaborative missions that can be undertaken on our system which allow public and private sector entities to leverage their local capacity to serve the recovery of the community. Our initial work has been recognized by FEMA (Federal Emergency Management Agency) Private Sector Office as a model in assistance of Public-Private Partnerships [1]. Also it is worth mentioning that during the Miami's hurricane season from Jun. 1st to Nov. 30th in 2015, DI-DAP has successively applied by Miami-Dade EOC (Emergency Operation Center) to help more than 1.75 million people, who are within one of Miami-Dade County's five Storm Surge Planning Zones, better understand what storm surge could mean to them and their homes[3].

6.2.2 A Case Study

In general, DI-DAP can guide users toward an efficient and comprehensive understanding on disasters by effectively a) crawling and retrieving disaster-related information using disaster vertical search engine; b) presenting an overview on the disaster evolution using the storyline visualizer; and c) conducting advanced geo-spatial data analysis using TerraFly GeoCloud. In this section, we present a case study to better illustrate how our platform can benefit disaster information management.

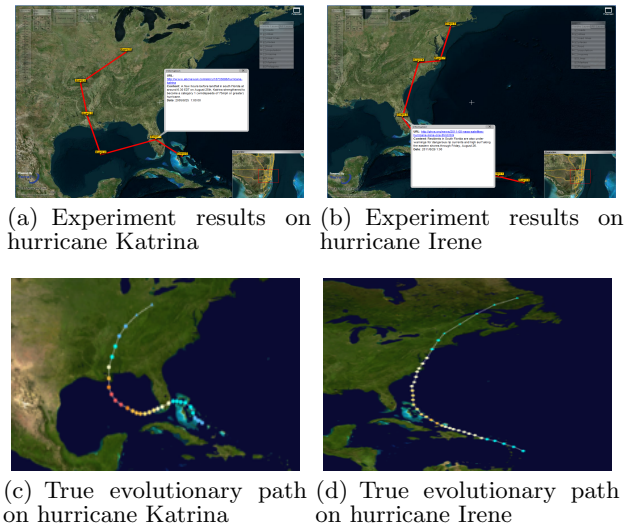
A lay user, Laura wants to find out the realtime social and economic impacts (e.g., housing prices and criminal rates) of Hurricanes Katrina and Irene. She first needs to retrieve relevant documents about these events. As shown in Figure 7(a), our vertical search engine is able to provide the most relevant information. Furthermore, our advanced information presentation, including both abstract and map mashup, allows her to quickly know where and when an event occurred and glean the event information at a quick glance.



(a) Vertical Search Engine Portal and Visualization (b) List of disaster events extracted from Wikipedia

Figure 7: Search Portal and Disaster Events

Our system also extracts the summarized information about historical disasters from Wikipedia [4] as a knowledge database to understand users' input from the search portal. While our vertical search engine provides Laura with the relevant information, she may still need to review lots of documents to get an overall picture of the disaster events as the documents may be scattered on the map and not chronologically and spatially organized. To overcome this disadvantage, disaster storyline visualizer is utilized. In particular, if the search keywords include a disaster name like *Irene* [5], disaster storyline visualizer will be triggered automatically. Figure 7(b) gives an brief overview on those disasters. Figures 8(a), 8(b), 8(c), 8(d) present the disaster storylines generated by our storyline generator and those manually labeled by human about disasters *Katria* and *Irene*. Those figures show that our system generates almost the identical paths as the true evolutionary ones.

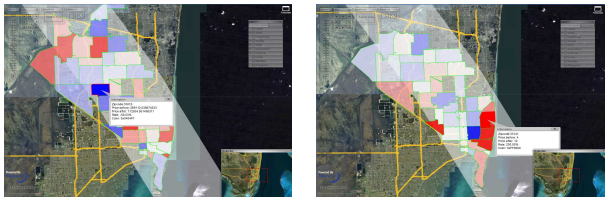


(a) Experiment results on hurricane Katrina (b) Experiment results on hurricane Irene

(c) True evolutionary path on hurricane Katrina (d) True evolutionary path on hurricane Irene

Figure 8: Experimental results of Hurricane Katrina and Irene

Once Laura knows the evolution path of the hurricane events, she is interested to find out how the house property price and criminal rate change in the affected region after disasters for further business planning and decision making. Then she performs geo-spatial data analysis using TerraFly GeoCloud. Figures 9(a) and 9(b) show the results of average house property price and average criminal rate variation before and after the disaster Irene in the region along the evolutionary path of disaster Irene. The colors indicate the variation ratios: the red color for great increase and the blue color for decrease.



(a) Average house property price variation during hurricane Irene (b) Average criminal rate variation during hurricane Irene

Figure 9: TerraFly geo-spatial data analysis

In this case study, we've well demonstrated how users can benefit from DI-DAP platform. Moreover, the role of DI-DAP in the success of businesses is immeasurable. Studies [22] show that about 40% of the companies that closed for three or more days as a result of a hurricane failed within 36 months. If DI-DAP helped 5% of the companies in South Florida to speed up their hurricane recovery by one week, it would prevent $220 million of non-property economic losses that would result from that week's closure [36, 37, 38, 39].

7. CONCLUSIONS

This paper presents the design and implementation of DI-DAP, an efficient and effective disaster information delivery and analysis platform to address the information explosion problem during the disaster period. DI-DAP supports three important services: *Disaster Vertical Search Engine*, *Disaster Storyline Generation*, and *Geo-Spatial Data Analysis Portal* and provide convenient, interactive, and timely disaster information to the users in need.

[3]http://www.miamidade.gov/hurricane/

[4]Wikipedia pages are semi-structured, brief disaster information can be easily extracted by analyzing the web pages' DOM tree.

[5]Fuzzy match is used here

With further development and refinement, it can be a powerful tool in disaster management, especially for improving situation awareness. We hope this work provides the community with a fresh perspective on the "practical" aspects of building and running a user-friendly and powerful disaster management tool.

8. ACKNOWLEDGMENTS

The work was supported in part by the National Science Foundation under Grant Nos. HRD-0833093, CNS-1126619, IIS-1213026, CNS-1461926, the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0001, Scientific and Technological Support Project (Society) of Jiangsu Province (No. BE2016776), Natural Science Foundation of China under Grant NO.61503313, and an FIU Dissertation Year Fellowship.

9. REFERENCES

[1] F. E. M. Agency. https://www.fema.gov/public-private-partnership-models. 2002.

[2] D. Ahlers and S. Boll. Adaptive geospatially focused crawling. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 445–454. ACM, 2009.

[3] M. Avvenuti, S. Cresci, A. Marchetti, C. Meletti, and M. Tesconi. Ears (earthquake alert and report system): a real time decision support system for earthquake crisis management. In *Proceedings of the 20th ACM SIGKDD*, pages 1749–1758. ACM, 2014.

[4] K. Bade and A. Nürnberger. Creating a cluster hierarchy under constraints of a partially known hierarchy. In *SDM*, volume 8, pages 13–24, 2008.

[5] S. Chakrabarti, M. Van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11):1623–1640, 1999.

[6] S. Cresci, A. Cimino, F. Dell'Orletta, and M. Tesconi. Crisis mapping during natural disasters via text analysis of social media messages. In *Web Information Systems Engineering*, pages 250–258. Springer, 2015.

[7] G. Erkan and D. R. Radev. Lexpagerank: Prestige in multi-document text summarization. In *EMNLP*, volume 4, pages 365–371, 2004.

[8] GeoVISTA. http://www.geovista.psu.edu.

[9] V. Hristidis, S.-C. Chen, T. Li, S. Luis, and Y. Deng. Survey of data management and analysis in disaster situations. *Journal of Systems and Software*, 83(10):1701–1714, 2010.

[10] M. Imran, C. Castillo, F. Diaz, and S. Vieweg. Processing social media messages in mass emergency: a survey. *ACM Computing Surveys (CSUR)*, 47(4):67, 2015.

[11] E. A. Inc. Webeoc. http://www.esi911.com/home.

[12] L. Li and T. Li. An empirical study of ontology-based multi-document summarization in disaster management. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 44(2), 2014.

[13] L. Li, D. Wang, C. Shen, and T. Li. Ontology-enriched multi-document summarization in disaster management. In *Proceedings of the 33rd international ACM SIGIR*, pages 819–820. ACM, 2010.

[14] S. Luis, F. C. Fleites, Y. Yang, H.-Y. Ha, and S.-C. Chen. A visual analytics multimedia mobile system for emergency response. In *Multimedia (ISM), 2011 IEEE International Symposium on*, pages 337–338. IEEE, 2011.

[15] I. Mani. Automatic summarization. *Computational Linguistics*, 28(2), 2001.

[16] S. E. Middleton, L. Middleton, and S. Modafferi. Real-time crisis mapping of natural disasters using social media. *Intelligent Systems, IEEE*, 29(2):9–17, 2014.

[17] NC4. E-teams. http://www.nc4.us/ETeam.php.

[18] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *icccn*, page 0215. IEEE, 2001.

[19] H. Purohit and A. P. Sheth. Twitris v3: From citizen sensing to analysis, coordination and action. In *ICWSM*, 2013.

[20] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408, 2002.

[21] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, pages 21–30, 2000.

[22] N. A. O. REALTORS. http://www.realtor.org/sites/default/files/hurricanes-impact-on-housing-and-economic-activity-case-study-florida-2006-04.pdf.

[23] K. Saleem, S. Luis, Y. Deng, S.-C. Chen, V. Hristidis, and T. Li. Towards a business continuity information network for rapid disaster recovery. In *Proceedings of the international conference on Digital government research*, pages 107–116. 2008.

[24] C. Shen and T. Li. Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 984–992. 2010.

[25] C. Shen, F. Liu, F. Weng, and T. Li. A participant-based approach for event summarization using twitter streams. In *HLT-NAACL*, pages 1152–1162, 2013.

[26] Ushahidi. http://www.ushahidi.com/, 2012.

[27] D. Wang, T. Li, S. Zhu, and C. Ding. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st international ACM SIGIR*, pages 307–314. ACM, 2008.

[28] Y. Yang, H.-Y. Ha, F. Fleites, S.-C. Chen, and S. Luis. Hierarchical disaster image classification for situation report enhancement. In *IRI, 2011 IEEE International Conference on*, pages 181–186. IEEE, 2011.

[29] J. Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power. Using social media to enhance emergency situation awareness. *IEEE Intelligent Systems*, 27(6):52–59, 2012.

[30] C. Zeng, Y. Jiang, L. Zheng, J. Li, L. Li, H. Li, C. Shen, W. Zhou, T. Li, B. Duan, et al. Fiu-miner: a fast, integrated, and user-friendly system for data mining in distributed environment. In *Proceedings of the 19th ACM SIGKDD*, pages 1506–1509. ACM, 2013.

[31] C. Zeng, H. Li, H. Wang, Y. Guang, C. Liu, T. Li, M. Zhang, S.-C. Chen, and N. Rishe. Optimizing online spatial data analysis with sequential query patterns. In *Proceedings of the 15th IEEE International Conference on Information Reuse and Integration (IRI)*, pages 253–260. IEEE, 2014.

[32] C. Zeng, L. Tang, W. Zhou, T. Li, L. Shwartz, and G. Grabarnik. An integrated framework for mining temporal logs from fluctuating events. *IEEE Transactions on Services Computing*, PP(99), 2016.

[33] M. Zhang, H. Wang, Y. Lu, T. Li, Y. Guang, C. Liu, E. Edrosa, H. Li, and N. Rishe. Terrafly geocloud: an online spatial data analysis and visualization system. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):34, 2015.

[34] H. Zhao and Z. Qi. Hierarchical agglomerative clustering with ordering constraints. In *Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on*, pages 195–199. IEEE, 2010.

[35] L. Zheng and T. Li. Semi-supervised hierarchical clustering. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 982–991. IEEE, 2011.

[36] L. Zheng, C. Shen, L. Tang, T. Li, S. Luis, and S.-C. Chen. Applying data mining techniques to address disaster information management challenges on mobile devices. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 283–291. ACM, 2011.

[37] L. Zheng, C. Shen, L. Tang, C. Zeng, T. Li, S. Luis, S.-C. Chen, and J. K. Navlakha. Disaster sitrep-a vertical search engine and information analysis tool in disaster management domain. In *IRI, 2012 IEEE 13th International Conference on*, pages 457–465. IEEE, 2012.

[38] L. Zheng, C. Zeng, L. Li, Y. Jiang, W. Xue, J. Li, C. Shen, W. Zhou, H. Li, L. Tang, T. Li, B. Duan, M. Lei, and P. Wang. Applying data mining techniques to address critical process optimization needs in advanced manufacturing. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014.

[39] W. Zhou, C. Shen, T. Li, S. Chen, N. Xie, and J. Wei. Generating textual storyline to improve situation awareness in disaster management. In *Proceedings of the 15th IEEE International Conference on Information Reuse and Integration (IRI)*, pages 585–592. IEEE, 2014.