

Real-time understanding of humanitarian crises via targeted information retrieval

K. T. Pham
P. Sattigeri
A. Dhurandhar
A. C. Jacob
M. Vukovic
P. Chataigner
J. Freire
A. Mojsilović
K. R. Varshney

Humanitarian relief agencies must assess humanitarian crises occurring in the world to prioritize the aid that can be offered. While the rapidly growing availability of relevant information enables better decisions to be made, it also creates an important challenge: How to find, collect, and categorize this information in a timely manner. To address the problem, we propose a targeted retrieval system that automates these tasks. The system uses historical data collected and labeled by subject matter experts to train a classifier that identifies relevant content. Using this classifier, it deploys a focused crawler to locate and retrieve data at scale. The system also incorporates feedback from subject matter experts to adapt to new concepts and information sources. A novel component of the system is an algorithm for re-crawling that improves the crawler efficiency in retrieving recent data. Our preliminary result shows that the algorithm can increase the freshness of collected data while simultaneously decreasing crawling effort. Furthermore, we show that focused crawling outperforms general crawling in this domain. Our initial prototype has received positive feedback from analysts at the Assessment Capacities Project, a humanitarian response agency.

1. Introduction

International humanitarian assistance reached a record high of U.S. \$28 billion in 2015 [1]. Even with the level of resources this affords, it is not possible to address all crises in the world. Therefore, prioritization is required. Over the last few years, there has been a notable increase in information collection, analysis, and dissemination to assess the nature and magnitude of major humanitarian crises around the world and prioritize the responses to them. In the early stages of an emergency, when collecting primary data (i.e., surveys, interviews, and direct observations) is limited by human resources, time, and access constraints, humanitarian crisis analysis relies crucially on secondary data (i.e., primary data that is processed by local and international public institutions, non-governmental organizations, and news media) [2]. Secondary data is deemed the key information source during the initial days and weeks after a disaster, compared with primary data [3]. Therefore, the systematic use of

secondary data to inform and provide context for emergency programming has become a norm. Example projects include Secondary Data Review [3], Syria Needs Analysis Project [4], Ebola Needs Analysis Project [5] by Assessment Capacities Project (ACAPS), Multi-Cluster/Sector Initial Rapid Assessment (MIRA) framework [6], and Human Needs Overview [7] by the United Nations Office for the Coordination of Humanitarian Affairs (OCHA).

However, the volume and variety of secondary information are rapidly becoming too unwieldy to process manually. Collating and organizing this information is time consuming and resource consuming, especially at the onset of disasters. As a result, too much time is spent collecting data and not enough time is spent making sense of it. For example, more than 15 people were dedicated to collecting and analyzing information for the recent Nepal earthquake crisis according to ACAPS. Five years ago, two people could follow 40 crises weekly, but today, eight people are required, and they are leaving increasing amounts of

Digital Object Identifier: 10.1147/JRD.2017.2722799

© Copyright 2017 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied by any means or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of this paper must be obtained from the Editor.

0018-8646/17 © 2017 IBM

information unprocessed. Information and communication technology have the potential to greatly improve humanitarian analysis by reducing much of the manual burden of collection and collation. However, developing these advancements requires different skills and work modes than those used in the past by the humanitarian community. Major investment in the last few years on solutions that augment the ability to collect data (i.e., mobile data collection systems [8]) has not been very useful because the existing automatic data capture platforms/systems and business software (see Section 3) are designed for general-purpose applications, not for collecting humanitarian crisis-related information; hence they return a high level of noise and are not used extensively.

In this work, we aim to develop a tailored solution that is usable and effective for humanitarian assessment. With guidance from ACAPS subject matter experts (SMEs) (in Section 2 of this paper, we introduce ACAPS) to understand the current workflows and requirements at their organization and other humanitarian agencies, we propose a targeted retrieval system to streamline the process of collecting, filtering, and classifying secondary data for humanitarian crisis analysis. The target secondary data includes media reports as well as published research from newspapers, response agencies, and humanitarian organizations, such as unocha.org, reliefweb.int, and fews.net. The system operates as a data pipeline that consists of four components: a domain-specific crawler, a metadata extractor, a content classifier, and an indexer. Of these, the first and third components utilize machine learning techniques bootstrapped with labeled data that ACAPS has collected since 2012.

Although various systems and tools have been proposed to study humanitarian crises and enhance situational awareness, social media services (e.g., Twitter^{**}, Facebook^{**}, and Reddit^{**}) are the primary sources of data [9]. Our work is an attempt to build a data pipeline system that targets secondary data for humanitarian crisis analysis. Compared with social media messages, secondary data requires different methods of acquisition and processing. In this paper, we discuss the requirements for the problem and present our initial prototype. The system uses historical data to fuel the learning components, including the focused crawler and content classifiers. Furthermore, its design includes the ability to use expert feedback to adapt the focused crawler over time, allowing it to be robust with respect to concept drift. Finally, we propose a new re-crawling technique for focused crawling and present a preliminary evaluation that demonstrates its effectiveness for this scenario. In our prototype, we use the open-source focused crawler ACHE (adaptive crawler for hidden-web entries) [10], an implementation of the focused crawler described elsewhere [11], as well as IBM Watson^{*}

services, i.e., IBM Retrieval and Rank (IBM-RR). We present empirical results on a real-world operation of the system to crawl for humanitarian crisis data.

The rest of the paper is organized as follows. In Section 2, we briefly introduce ACAPS and its secondary data collecting process. Then, we discuss related work in Section 3. In Section 4, we describe the system and the implementation of its components. In Section 5, we show our solution for achieving near-real-time crawling and the corresponding evaluation metric. Experimental results are presented in Section 6. Finally, we discuss limitations and future work in Section 7 and conclude in Section 8.

2. Assessment Capacities Project

ACAPS is a non-profit, non-governmental project of a consortium of three humanitarian non-governmental organizations: Action Against Hunger, Norwegian Refugee Council and Save The Children. At ACAPS, humanitarian specialists or SMEs perform data analysis to inform operational, strategic, and policy decision-makers. This requires collecting reliable and timely secondary data.

SMEs from ACAPS have been collecting secondary data from the web since 2012 for analyzing humanitarian crises following the guideline described in Secondary Data Review [3]. The data is available from different sources such as non-governmental organizations (e.g., hrw.org and crisisgroup.org), United Nation agencies (e.g., unhrc.org, fao.org, reliefweb.int, and unocha.org), and news media (e.g., bbc.com, aljazeera.com, and trust.org). Some sources provide general information and thus only contain a small portion of documents relevant to the humanitarian domain, whereas others run by humanitarian agencies and organizations do have more. A simplified process of data collection by SMEs starts with navigating the main humanitarian information sources. If a document is determined as relevant, it will be annotated with regard to geographical area, publication date, type of source, crisis group, crisis type, and crisis subtype. The content of each document is also summarized into a more coherent piece of text before being saved to Excel spreadsheets for further analysis. Dropbox^{**}, an information-sharing platform, is used for sharing the collected information between SMEs.

3. Related work

In this section, we discuss similarities and differences of our proposed system with existing approaches to humanitarian analysis and general information retrieval pipelines. First, we note that a significant number of projects from humanitarian agencies have been deployed to collect and annotate secondary data to support humanitarian crisis analysis, including Secondary Data Review, Syria Needs Analysis Project, Ebola Needs Analysis Project, and Human Needs Overview. However, the data collection and filtering processes of all of these projects rely fully on the manual

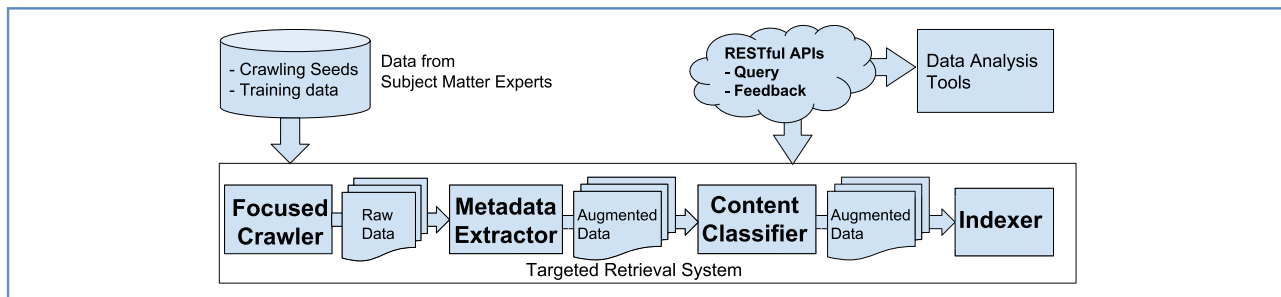


Figure 1

Targeted retrieval system.

efforts of SMEs. To our knowledge, no available automatic system has been built to improve these tasks.

The humanitarian agencies have experimented with existing commercial automatic data capture platforms and business software such as Crimson Hexagon** [12], GDELT (Global Database of Events, Language, and Tone) [13], and Mention** [14]. However, as mentioned earlier, these tools collect media data for a general purpose and hence lack the advantages of the focused crawler, which is employed in our system.

Since social media services (i.e., micro-blogging, social networking sites, and social media sharing platforms) have become a common means for affected populations and concerned individuals to communicate and spread information during crisis events, systems proposed by academics and practitioners to enhance situational awareness considered social media data as a primary source [9]. Example systems that use Twitter messages (tweets) include CrisisTracker [15], AIDR (artificial intelligence for disaster response) [16], and Twitcident [17]. CrisisTracker is a semi-automatic system that incorporates crowdsourcing-based processing of tweet streams and automatic clustering to detect new events for disaster awareness. Similarly, AIDR also employs crowdsourcing workers to annotate crisis-related tweets and automatically classify them based on user-defined categories. Twitcident “listens” to emergency broadcasting services to trigger the tweet monitoring process. To allow effective filtering with a faceted search interface, the collected tweets are semantically enriched using name entity recognition, classification, linkage to external web sources, and metadata extraction. Both of these systems access tweets from Twitter APIs (application program interfaces), as crawling is against the terms of service of most social media platforms. Compared with news articles and reports, tweets are fundamentally different in term of size and type of content, therefore requiring different processing techniques.

Alonso [18] noted opportunities to design information retrieval pipelines from a data perspective, with such considerations as how data is retrieved, stored, cleaned,

extracted, classified, and indexed. This work also presents five parts of the data stack in information retrieval: raw data ingestion, data augmentation with annotation, content indexing and ranking, behavior data capturing and analysis infrastructure. The design of our data processing pipeline follows these principles focusing on the first three parts. Database researchers also study end-to-end data processing [19], but despite the overlap in work between the database and information retrieval communities, there is a key difference. The database community focuses on data storage and analysis, whereas the information retrieval community focuses on data collection and retrieval.

Research on focused crawling has a long history, since the rapid growth of the world wide web, leading to unprecedented scaling challenges [11, 20–22]; however, few pieces of work have been deployed for social good domains and integrated into fully automatic pipelines. Realizing that the gap exists, the Memex Project [23] from the Defense Advanced Research Projects Agency (DARPA) uses a focused crawler for indexing and curating data to counter human trafficking (as opposed to analyzing humanitarian crises). The open-source focused crawler ACHE, employed in our system, has been developed under the Memex Project.

We reserve discussion about related work for re-crawling (which is one of the contributions of our system) to Section 5.

4. System architecture

As we discussed in the previous section, the current process of humanitarian crisis analysis using secondary data lacks automation, which leads to a large amount of manual time and effort spent on collecting, filtering, and annotating reports. In this section, we describe our proposed automated system to ameliorate this problem; we discuss all of the components of the system, as well as how we build them using data from ACAPS.

As shown in **Figure 1**, the system consists of four components in a pipeline. The first component is the focused crawler that is designed to return only

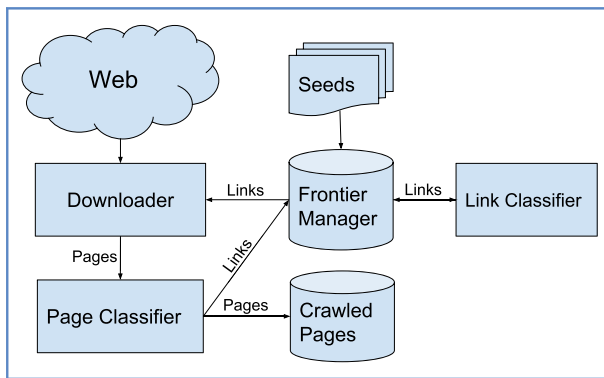


Figure 2

Block diagram of a focused crawler, showing its main components and the data flow.

humanitarian-crisis-related information from the Web. This is the main component that distinguishes our system from other solutions using social media monitoring services. The next component, the metadata extractor, extracts information from the crawled documents that can be useful for certain analyses. Extracting comprehensive types of information is beyond the scope of this work, but in our prototype, we extract some basic properties of the documents such as textual content, publication date and country. These are also the main pieces of information that ACAPS SMEs extract from the collected documents in their current process. After the extraction, the content classifier determines the type of crisis described in the text, one of the essential pieces of information that SMEs seek from the documents [3]. These components are bootstrapped (initialized) using historical data collected by ACAPS. After all the processes, the crawled documents and their metadata are indexed to be retrievable via RESTful APIs (representational state transfer, application programming interfaces). The APIs also allow the system to receive feedback from users. The following subsections describe each component in detail.

4.1. Focused crawler

A focused crawler (some refer to this as a topical crawler, domain-specific crawler, or scoped crawler) is a web crawler that is optimized to seek web pages that are relevant to predefined topics [19]. In other words, a focused crawler “tries” to visit only small subsets of the web where relevant pages reside and avoid fetching irrelevant ones. **Figure 2** depicts a simplified block diagram of a focused crawler, similar to the one described in [21]. In this minimal form, the crawler performs a best-first search to explore a web graph, as opposed to a breadth-first search in a general crawler that aims to crawl as many pages as possible.

Its processes can be briefly outlined as follows. The downloader component requests unvisited universal resource locators (URLs) from the frontier manager (i.e., a component that stores and ranks unvisited URLs, as well as determines the next URLs to be visited). The downloader then fetches the URLs and sends their content to the page classifier for computing the relevance score with respect to the topic of interest. URLs of the relevant or irrelevant pages are used as positive or negative examples for training the link classifier, which is used by the frontier manager in ranking unvisited URLs. The out-links extracted from the fetched content are sent to the frontier manager to be ranked and stored.

A focused crawler requires the construction of a page classifier in advance, while the link classifier may either be initialized empty or be pre-trained and updated during operation. We use the ACHE focused crawler in our implementation of a humanitarian crisis crawler. ACHE allows users to train a page classifier simply by providing a set of relevant and irrelevant pages of the domain. Beyond the built-in functionality of ACHE, we made some slight modifications in feature extraction to improve page classification accuracy for the domain. Below, we present how the page classifier and link classifier are built for working with ACHE.

4.1.1. Page classifier for humanitarian crisis documents

In our focused crawler, the goal of the page classifier is to identify whether a web page is relevant to humanitarian crises; therefore, we use binary labels in training the classifier (i.e., a relevant/irrelevant document to a humanitarian crisis is considered positive/negative). Training data comes from both ACAPS and our own manual annotation.

Irrelevant documents were not initially annotated by SMEs, leaving two options: either applying one-class classification methods or creating a set of negative examples. Although there exist methods for one-class classification such as the one-class SVM (support vector machine), this paradigm has been shown ineffective in classifying web documents [24, 25]. Therefore, we collected negative documents as follows. We randomly crawled pages using breadth-first search starting from the homepages of seed websites (i.e., the websites where ACAPS SMEs most frequently found relevant documents) to discover in-site links, resulting in 11,000 candidate negative examples to then be manually annotated. In order to reduce the annotation effort, we applied a heuristic to remove positive examples from the candidate set. Specifically, we removed pages from the candidate set whose URLs contain crisis-related keywords (e.g., earthquake, flood, fire, and volcano), as they are unlikely to be negative examples. We then manually discarded positive examples remaining in the candidate set by examining their

URLs, titles, and main content if necessary, leaving a set of 2,600 negative documents. We also obtained an additional 100 irrelevant documents from ACAPS SMEs, giving us a final set of 2,700 negative examples to train the page classifier.

We randomly selected 2,700 documents out of 17,828 reports about humanitarian crises provided by ACAPS SMEs as positive samples. The down-sampling step is to balance the positive and negative classes. Details of this collection and how we process the documents are presented in Section 6.1.

Our feature extraction builds upon ACHE's bag-of-words-based feature extraction module. It extracts the textual content from the HTML source and generates word features using term frequency-inverse document frequency scores. From our observation of the relevant humanitarian pages, their URLs and titles can be as informative as the main content to determine the relevance of a page; hence we add them to the feature set to make the classifier more robust. Note that tokens generated from URLs and titles are concatenated with distinguishable prefixes—URL and TITLE, respectively—to distinguish them from tokens from the main content.

The page classifier we use is the SVM, which has been empirically successful in text classification tasks especially with standard benchmarks [26]. ACHE provides a wrapper of Weka's SVM, implemented using sequential minimal optimization (SMO).

4.1.2. Link classifier

The link classifier ranks the URLs in the crawler frontier so that the crawler can perform a best-first search by prioritizing the highest scores to visit. It computes the score of an unvisited URL by examining its anchor text, the surrounding words, the URL string and the parent page. We use the link classifier of ACHE without any modification. Its training and feature generation processes are similar to those of the page classifier described above. If the link classifier is initialized empty, it will be updated in an online fashion based on the examples labeled by the page classifier. For a more elaborate explanation of the link classifier, see [21]. However, if the link classifier is initialized with pre-training, the crawler has better performance, as shown in the crawler evaluation later. As part of this work, discussed in the section on re-crawling, we improved ACHE so that it can reuse the link classifier directly from the results of previous crawls.

4.1.3. Crawler configuration

There are two main configurations: crawling scope and follow rule. The first configuration determines whether the crawler should crawl beyond the provided seed websites, and the second one determines whether the crawler should follow all out-links extracted from the given pages.

The first configuration is crawling scope. We seed the crawler with 100 websites that ACAPS SMEs have found to contain the most humanitarian crisis-related documents. For the crawler evaluation and prototype, we restrict the crawler to perform inside those websites since they contain representative numbers of relevant documents according to ACAPS. However, we note that it is "effortless" to configure the crawler to crawl beyond the seeds to search for more relevant documents. Nevertheless, it requires more representative set of training data so that the page classifier can perform well on documents outside the seed websites.

The second configuration is follow rule. Following out-links from relevant pages is obviously necessary, since relevant pages tend to link to other relevant pages [22], but it is not obvious whether links from irrelevant pages should be followed. Doing so can improve the recall by discovering more relevant links that are not directly linked to the relevant pages but might reduce the harvest rate, the number of relevant pages divided by a total number of retrieved pages. In our crawler evaluation later, we configure the crawler to follow out-links from all crawled pages, which leaves all of the ranking work to the link classifier. This helps evaluate the link classifier more accurately.

4.2. Cleaning, extraction, and classification

4.2.1. Cleaning

Although data cleaning is not depicted in our system diagram, it happens at all stages of the pipeline, from preparing data provided by SMEs to indexing processed data. Indeed, each stage unveils document properties that can be used to remove unnecessary documents. For example, at the crawling stage, the system filters out the pages that were already processed in the previous crawls before passing to the extraction stage. At the extraction stage, the system removes malformed documents or ones that are not likely to be in the scope of interest.

4.2.2. Metadata extractor

The extraction component augments the crawled data by extracting the title, the textual content, the publication date, and the mentioned country from the crawler documents. The title is extracted using regular expressions. To extract the textual content, we use the boilerpipe library [27], an extractor that implements and extends concepts described in [28]. To extract the publication date, we look for the value of common HTML tags that store this information such as "date", "published_date", "date_created", "published_time", etc. Only if none is found do we use a regular expression to search for strings with date format in both the URL and the extracted textual content; if multiple matches are found, we choose the one that is most recent. To extract the mentioned country, we maintain a list of country names and check if any tokened

Table 1 Crisis types and evaluation results of the content classifier using cross-validation.

<i>Crisis type (label)</i>	<i>No. of documents</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 score</i>
Accident	338	0.40	0.01	0.02
Drought	822	0.78	0.63	0.70
Earthquake	190	0.94	0.70	0.80
Epidemic	1,355	0.91	0.78	0.84
Extreme temperature	86	0.89	0.36	0.51
Flood/rain	1,821	0.84	0.81	0.83
Food insecurity	625	0.80	0.43	0.56
Insect infestation	36	0.93	0.72	0.81
Mass movement	77	0.88	0.19	0.32
Refugees	346	0.84	0.11	0.19
Storm/tropical/cyclone	404	0.84	0.65	0.73
Tension/dispute/conflict	11,589	0.86	0.99	0.92
Volcano	104	0.94	0.69	0.80
Wildfire	20	1.00	0.15	0.26

word from the articles falls into this list. We accept more than one country to be associated with a document as a crisis event can happen across multiple countries. These extractions are mostly primitive but sufficient to demonstrate the functionalities of the proposed system. They can be improved in future work by state-of-the-art learning methods if labeled data is available.

4.2.3. Content classifier

In the secondary review process, SMEs at ACAPS categorized the collected documents into crisis type and subtype. There are 17 types of crises and 76 subtypes according to the ACAPS categorization. We automate the document annotating process by using a content classifier (multiclass classifier) with the crisis types as labels. We decide not to use subtypes as labels since the corresponding class distribution is highly imbalanced and the number of training examples is quite small with respect to the number of labels. We merge some crisis types that have too few documents and belong to the same group of crisis type. Specifically, industrial accident, miscellaneous accident, and transport accident become *accident*; mass movement (dry) and mass movement (wet) become *mass movement*. A final list of 14 crisis types and the number of corresponding documents for training are shown in **Table 1**. The collection

contains 17,828 documents that are processed from the collection provided by ACAPS (see Section 6.1 for more detail of the collection). We apply standard preprocessing to construct the bag-of-words feature space, including removing stop words, non-alphabet symbols, and non-digit tokens, and we also apply bi-gram tokenization. The classifier is then trained using the SVM algorithm.

4.3. Indexer and RESTful API

We need the last component in the pipeline to store the augmented documents in a predefined schema and provide a standard mechanism for other applications to easily interact with. Specifically, after the extraction and classification processes, each document object contains HTML source, URL, extracted textual content, top-level domain, mentioned country, publish date, crawl date and crisis type. These properties form the schema of the document in the indexer. Having said that, our ideal system should allow extending the schema when there are more extractions and classifications applied. We also want the component to support different kinds of applications to query and update the data. Furthermore, since our data contains unstructured text, a full-text search feature should be supported. With these requirements in mind, we use IBM-RR [29] to index

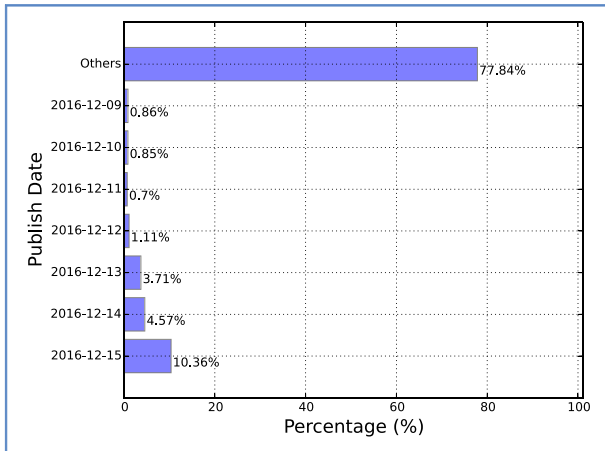


Figure 3

The percentages of relevant documents according to their publish dates.

the processed data. It also implements a HTTP-based RESTful API, a widespread and efficient standard in designing APIs for Internet services.

In order to demonstrate the functionalities and evaluate the usability of the system, we build a faceted search interface as a front-end application. It accesses the documents from the indexer through a RESTful API and enables users to browse through documents by choosing from a set of categories that are defined by the metadata associated with the documents.

4.4. Feedback cycle

One important and distinctive feature that we design into the system is a mechanism that allows it to receive feedback from users to improve the page classifier. In the previous section, we have discussed that the page classifier determines the efficiency of the focused crawler via the link classifier. A simple way to improve the page classifier is to acquire a reasonable amount of representative training examples. This especially increases the robustness of the page classifier as well as the adaptivity of the crawler.

This feedback mechanism can be facilitated in our system through the RESTful API. However, for this feature to be fully operational, there must be implementation on both the back end (our system) and the front end (analytic tools and applications). When users explore documents from front-end applications, they should be able to annotate incorrectly labeled documents. These are deemed as user feedback and sent back to the system through the RESTful API. At the back end, we use the indexer to hold both the original training data and the feedback set, which simplifies the implementation. Indeed, the indexer already provides a RESTful API; hence, the

other components can easily receive user feedback without further implementation.

5. Real-time re-crawling for the focused crawler

5.1. Problem description

Humanitarian crisis information changes rapidly as the nature of emergency situations change. As a result, the data collection should be performed as frequently as data changes in order to produce a timely analysis. This is another problem in crawling. Indeed, we have demonstrated the focused crawler that collects documents in a cold-start setting, for all documents classified by the page classifier as positive. However, when we re-run the crawler, all documents published before the last crawl become irrelevant. **Figure 3** shows the distribution of relevant documents by their publish dates. The crawl spans, in two days, from 2016/12/14 to 2016/12/15 and retrieves 55,000 relevant documents. From these, we were able to extract publish dates in 45,000 documents. We can see in the figure only 14.93% out of the relevant documents are published at the crawl time. This ratio can be higher than that obtained by starting the crawl from random seeds, since newly published documents tend to be listed in the homepages, where our crawl starts. If we perform the crawl daily, then all the documents before the crawl can be considered irrelevant. This observation can help us to optimize the crawling strategy in the re-crawling setting.

In re-crawling, the goal is to avoid revisiting pages crawled in previous crawls while still discovering new relevant pages, i.e., pages published after the last crawl. That being said, since the newly published pages are usually linked to existing pages [30], revisiting is unavoidable. When time and computing resources are constrained, we need to solve the trade-off. The time constraint comes from the demand of discovering the relevant pages as soon as they are published. Ideally, with a sufficiently large number of distributed machines, and computing power, this can be thoroughly solved. However, this level of computing resources is often not available, especially for humanitarian organizations.

Re-crawling is not a new problem: it has been studied for web crawlers [30–32] and has been widely deployed in successful web search engines. However, in domain-specific crawling, the problem requires a different solution because of the concept of relevance and limited computing resources. A notable approach introduced in [30] is to mine the historical crawls to identify a small set of pages that are more likely to link to new pages. However, this requires multiple crawl snapshots to identify the content changes. In our scenario, multiple snapshots are not necessary as we can extract the publish dates from the crawled pages. As a result, we can identify the change of a page based on the publish dates of its out-links.

Algorithm 1 Greedy algorithm to find the top K nodes that maximize $Score(K)$.

```

TopK( $K, G = (V, E)$ )
1:  $covered\_nodes, selected\_nodes = \emptyset$ 
2:  $V = V_p \cup V_r$  //  $V_p$  is the list of all parent nodes, and  $V_r$  is the list of all relevant nodes.
3: while there are less than  $K$  nodes in  $selected\_nodes$ 
4:   select  $v$  in  $V_p$  such that  $Score(v) = \sum_{p \in Children(v) \setminus covered\_nodes} A_p \times r_p$  is maximum
5:   add  $v$  to  $selected\_nodes$ : remove  $v$  from  $V_p$ 
6:   add  $Children(v)$  to  $covered\_nodes$ 
7: end while
8: return  $selected\_nodes$ 

```

To solve the problem in our setting using a single crawl snapshot, we can redefine the problem formally as follows. Given the directed graph $G = (V, E)$, where G is discovered from the previous crawl, V is the set of all nodes, and E is the set of all edges in the graph. Each node represents an individual web page and is associated with two values. The first value (r_p) is the relevance of the corresponding page; it has value 1 if the page is relevant and 0 if it is irrelevant. The second value (t_p) indicates the time the page was published. Each directed edge in E represents a directed link from a parent node to a child node. Assume that we can only crawl a maximum of K parent pages; hence, K is the crawl budget. The goal is to identify K parent pages so that the numbers of their relevant children that are published recently are maximized. Let the reward A of a relevant page p be determined by a decay function f of its publish time t_p :

$$A_p = f(t_p). \quad (1)$$

Then, the score of K pages $V_k = \{v_1, v_2, \dots, v_k\}$ is computed by the total reward of the union (U) of all their child pages. It is noted that multiple pages can link to the same child nodes; however, all nodes in U are unique.

$$Score(K) = \sum_{p \in U} A_p \times r_p. \quad (2)$$

The goal is to find K pages so that $Score(K)$ is maximized. The procedure $TopK()$ in **Algorithm 1** presents a greedy algorithm to approximately solve the problem.

5.2. Evaluation metric

The focused crawler performance has been measured by the harvest rate metric [11], i.e., the fraction of pages visited that are relevant:

$$HR = N_{RP}/N_{VP} \quad (3)$$

Here, HR , N_{RP} , and N_{VP} are the harvest rate, number of relevant pages, and number of visited pages, respectively.

This is the metric for measuring the one-time crawler. In re-crawling, we are only interested in the relevant pages that are published after the previous crawls. Therefore, the metric for re-crawling should be refined as follows:

$$THR = N_{TRP}/N_{VP}. \quad (4)$$

THR , N_{TRP} , and N_{VP} are the temporal harvest rate, number of relevant pages published at crawl time, and number of visited pages, respectively.

6. Experimental results

We implemented a fully operational prototype of the proposed system. Since the system's design is inspired by the current data-collecting process at ACAPS and is bootstrapped by data collected from this process, in this section we first detail the manual process. While evaluation of the data pipeline as a whole is not available, which requires a comprehensive study by SMEs, we evaluate some components individually. We also sought feedback from SMEs at ACAPS with respect to the quality and usability of the collected data via a demonstration of a faceted search interface built on top of the system APIs. The feedback is positive, in a sense that the data and its augmented information can be directly useful for further analysis. In this section, we present the evaluation results of the classifiers and the performance comparisons of the focused crawler component for the humanitarian crisis domain in different settings.

6.1. Understanding data from ACAPS

We received data from ACAPS, containing 34,700 documents in Microsoft Excel** format. These come from 433 distinct websites. We select the top 100 websites that yield the greatest number of collected documents as seeds for crawling. Of these 34,700 documents, we perform several cleaning steps before they can be used for training the classifiers described above. First, we remove the documents whose crisis type fields are missing, because we need these values as class labels for training the content classifier. Then, we remove the duplicated documents based

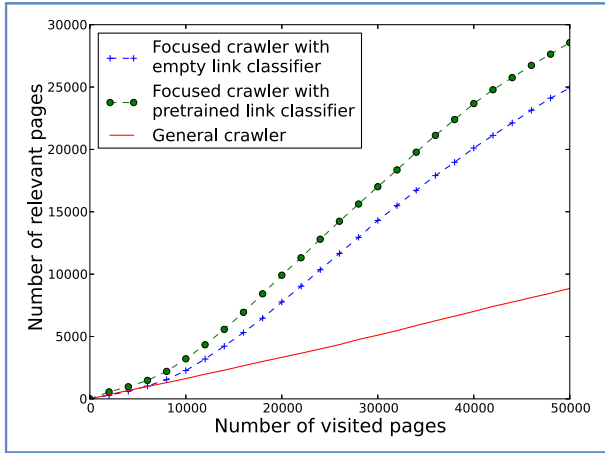


Figure 4

Comparison of the harvest rate of the general crawler with the focused crawler (with the link classifier initialized empty and with the pre-trained link classifier).

on the content summarization field. Next, as original HTML source of the documents is not saved, we retrieve them again via their URLs. The HTML source is necessary for training the classifiers since they feature inputs of the classifiers. Also, we discard the documents whose formats are PDF (Portable Document Format), as we only process HTML source in the scope of this work. From the HTML source, we extract the textual content and use this for language detection. We then filter the documents that are not in English, which consists of about 9% of the total collection. After all of these steps, we obtain a final set of 17,828 documents that are ready for training the classifiers.

6.2. Classifier evaluation

6.2.1. Page classifier evaluation

We obtain a 0.94 F1 score (the precision and recall are 0.95 and 0.94, respectively) in cross-validation with our set of positives and negatives of humanitarian crisis documents described in Section 4.1.1. It is noted that the high score does not guarantee that the classifier would perform as well on documents outside the seed websites, where the training set was collected. A more representative collection of documents would be important to improve the robustness of the classifier.

6.2.2. Content classifier evaluation

We evaluate the classifier with fivefold cross-validation and obtain the 0.84 weighted macro-average F1 score. Note that the predicted label for each document in the training set was obtained for that document when it was in the test set. Table 1 shows the precision, recall, F1 score, and number of corresponding documents for each label. Although the overall F1 score is quite high, given the fact that there are

14 different labels, the F1 score for some labels is extremely poor partly due to the dominant representation of documents from other labels (e.g., tension/dispute/conflict and flood/rain). We examine the documents from the three labels that have the lowest F1 scores (accident, wildfire, and refugees). Most of them are predicted as tension/dispute/conflict, which is the label that has the largest number of documents. The main reason is that the label wildfire has only 20 documents. Also, documents associated with accidents and refugees might share similar features with those from tension/dispute/conflict. For better understanding of this situation, experimenting with an interpretable classifier (e.g., a decision tree) would help.

6.3. Focused crawler evaluation

In this experiment, we restrict the crawler to the selected 100 seed websites for more control of the experiment of comparison between different crawls. Nevertheless, the crawler can perform outside these seeds to seek for more relevant sources of humanitarian document. In this section, we compare the crawler performance on the humanitarian domain with different crawling settings.

6.3.1. Crawler performance over time

To compare performance of the crawler with different configurations, it is useful to analyze how the corresponding harvest rates change over time. In Figure 4, we show these results for the general crawler, focused crawler with empty link classifier, and focused crawler with pre-trained link classifier. We disable the link classifier and page classifier in ACHE to obtain the result for the general crawler. We enable the link classifier and page classifier in ACHE to run it in the focused mode. In the first time run, the link classifier is initialized as empty and is trained while the crawler is running. In the second run, we reuse the link classifier trained from the first run and disable the online learning for the link classifier.

It is easy to observe that the focused crawler in both settings outperforms the general crawler in terms of harvest rate. Specifically, the focused crawler with pre-trained link classifier performs the best and retrieves approximately three times more relevant pages than the general crawler does. We note that one should reinitialize the link classifier once the page classifier is updated, since the link classifier is built on the results of the page classifier.

6.3.2. Re-crawling performance over time

We show below a linear function to model the change of the reward of a given relevant page with respect to its publish time. If a page was published more than d days before the crawl time, it has a minimal reward, which is 1:

$$A_p = f(t_p) = \max(1, d - (t_{\text{crawl}} - t_p)). \quad (5)$$

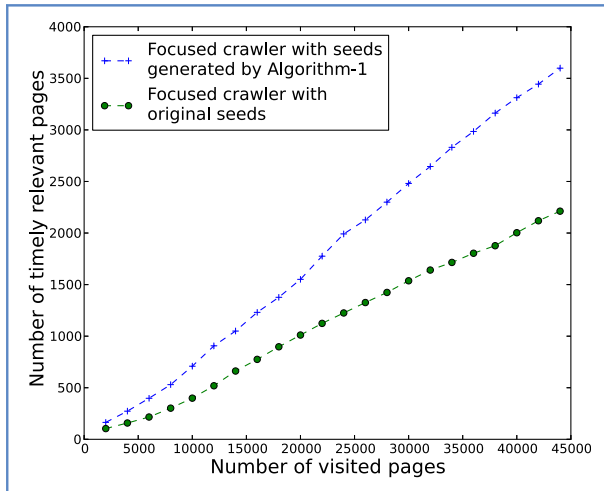


Figure 5

Comparing the temporal harvest rate between focused crawlers using original seeds and seeds generated by Algorithm 1.

Here, t_{crawl} is the time when starting the crawler, and we use “day” as a time unit. We set d to 7 in the experiment.

We obtain 2,000 URLs by running Algorithm 1 with $K = 2,000$ with the crawled data obtained by running the focused crawler with the pre-trained link classifier. Two weeks after that crawl (2016/12/27), we start two other crawls concurrently. One crawl uses the original seeds of 100 URLs, and the other uses the 2,000 URLs generated by Algorithm 1. Both crawls use the same pre-trained link classifier and page classifier as in the crawler experiment presented in the previous section. We show the performance of these two settings with respect to the change of temporal harvest rate over time in **Figure 5**. Here, we define a page as relevant if its publishing date is the day we ran the crawl and it is classified as relevant by the page classifier. We can see that using the seeds generated by Algorithm 1, the temporal harvest rate (THR) is improved by about 50% in retrieving the relevant pages. Despite the encouraging improvement, this preliminary result leaves space for the future work. Specifically, in this experiment, we restrict the crawler to traverse only within 100 websites; however, expanding the experiment to a larger corpus would definitely yield more insights (e.g., publishing patterns and frequency of relevant documents) of the domain. Furthermore, there are several parameters in the formalizations that can affect the outcome, i.e., $f(t_p)$, d , and K ; therefore, it is also important to measuring this dependency. Lastly, our method relies on an assumption that hub pages that link to more recent relevant pages are likely the good sources in the future crawls; however, linking pattern might change. Indeed, web pages do have different changing patterns; existing work shows that

understanding the change pattern would lead to better crawling strategies [33]. It is noted that in our work, we consider the change of a page with regards to its out-links rather than its content.

7. Limitations and future work

In this work, we focused on automating parts of the workflow of experts at ACAPS for searching and cataloging humanitarian crisis secondary data. The system was bootstrapped and evaluated only on the humanitarian data sources commonly used by the subject matter experts at ACAPS. Although this limits the system to perform well on the new data sources, this helped us simplify the process of gathering feedback, as they were already familiar with the data sources and the domain. Another limitation of the study is that it is lacking an evaluation of the system as a whole; we hope that the deployment of the system at humanitarian agencies in the future could make the study more complete.

A major avenue of improvement to the system involves the end-user experience. In order to make the system appeal to the end-users, which are humanitarian agencies, it requires more contributions, namely advanced domain-specific extraction (number of people affected, time, level of emergency), data cleaning (e.g., de-duplication) and a visual analytics/search interface (that can leverage the extracted information and allow an efficient feedback mechanism).

Enriching the data sources using URLs extracted from social media streams could also make the information collected by the system more useful and timely. Further efforts are also needed to evaluate the effectiveness of the system with more diverse data sources and also extended use cases that involve non-humanitarian domains.

8. Conclusion

Collecting and annotating secondary data is one of the main precursors to support the strategic and operational decision-making processes of humanitarian crisis response. These tasks currently involve high levels of manual work from SMEs at humanitarian agencies, thereby consuming a large portion of the effort of the entire analysis process. Also, with the rapid growth of information from the Web, it is impossible to navigate and capture the relevant information in a timely fashion. In this work, we presented a targeted retrieval system that streamlines the process of collecting, filtering and classifying secondary data for humanitarian crisis analysis. The system employs a focused crawler as its core and uses historically collected data by ACAPS to train its classification components. Its capacity of retrieving timely relevant documents is also improved by our proposed algorithm. The system is fully implemented as a prototype whose results can be accessed via RESTful APIs.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of Twitter, Facebook, Reddit, Dropbox, Crimson Hexagon, Mention Solutions, or Microsoft Corporation in the United States, other countries, or both.

Acknowledgments

This project was conducted under the auspices of the IBM Science for Social Good initiative.

References

1. Global Humanitarian Assistance Report, 2016. [Online]. Available: <http://www.globalhumanitarianassistance.org/wp-content/uploads/2016/07/GHA-report-2016-full-report.pdf>
2. United Nations Disaster Assessment and Coordination, 2013. [Online]. Available: https://docs.unocha.org/sites/dms/Documents/UNDAC%20Handbook_interactive.pdf
3. "Second data review—Sudden onset disasters." Tech. Brief, May 2014. [Online]. Available: https://www.acaps.org/sites/acaps/files/resources/files/secondary_data_review-sudden_onset_natural_disasters_may_2014.pdf
4. Syria Needs Analysis Project, Jun. 2015. [Online]. Available: https://www.acaps.org/sites/acaps/files/products/files/1_s-snap-summary-of-work-dec-2012-june-2015.pdf
5. Ebola Needs Analysis Project, Apr. 2015. [Online]. Available: https://www.acaps.org/sites/acaps/files/products/files/k_sierra_leone_multi-sector_needs_assessment_report_april_2015.pdf
6. "Multi-sector initial rapid assessment guidance," Jul. 2015. [Online]. Available: https://www.humanitarianresponse.info/system/files/documents/files/mira_revised_2015_en_1.pdf
7. "Human needs overview guidance," 2014. [Online]. Available: <https://docs.unocha.org/sites/dms/ROWCA/Coordination/SRP/HNO%20guidance%202014.PDF>
8. Humanitarian Operations Mobile Acquisition of Data, Jan. 2012. [Online]. Available: https://www.acaps.org/sites/acaps/files/resources/files/nomad_mobile_data_collection_systems-a_review_of_the_current_state_of_the_field_january_2012.pdf
9. M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: A survey," *ACM Comput. Surveys*, vol. 47, 2015, Art. no. 67.
10. ACHE, an open source focused crawler. [Online]. Available: <https://github.com/ViDA-nyu/ache>
11. L. Barbosa and J. Freire, "An adaptive crawler for locating hidden-web entry points," in *Proc. 16th Int. Conf. World Wide Web*, Banff, AB, Canada, 2007, pp. 441–450.
12. Crimson Hexagon, a social media analytics platform. [Online]. Available: <https://www.crimsonhexagon.com/>
13. Global Database of Events, Language, and Tone (GDELT). [Online]. Available: <http://www.gdeltproject.org/>
14. Mention, a social media and web monitoring tool. [Online]. Available: <https://mention.com/en/>
15. J. Rogstadius, M. Vukovic, C. Teixeira, V. Kostakos, E. Karapanos, and J. Laredo, "CrisisTracker: Crowdsourced social media curation for disaster awareness," *IBM J. Res. Develop.*, vol. 57, no. 5, pp. 4:1–4:13, 2013.
16. M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "AIDR: Artificial intelligence for disaster response," in *Proc. Int. World Wide Web Conf.*, 2014, pp. 159–162.
17. F. Abel, C. Hauff, G. J. Houben, R. Stronkman, and K. Tao, "Semantics + Filtering + Search = Twitcident. Exploring information in social web streams," in *Proc. Int. Conf. Hypertext Social Media*, Milwaukee, WI, USA, 2012, pp. 285–294.
18. O. Alonso, "The data stack in information retrieval," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Pisa, Italy, 2016, pp. 597–597.
19. D. Abadi, R. Agrawal, A. Ailamaki, M. Balazinska, P. Bernstein, M. Carey, S. Chaudhuri, J. Dean, A. Doan, M. Franklin, J. Gehrke, L. Haas, A. Halevy, J. Hellerstein, Y. Ioannidis, H. Jagadish, D. Kossmann, S. Madden, S. Mehrotra, T. Milo, J. Naughton, R. Ramakrishnan, Volker Markl, C. Olston, B. Ooi, C. Ré, D. Suci, M. Stonebraker, T. Walter, and J. Widom, "The Beckman report on database research," *Commun. ACM*, vol. 59, no. 2, pp. 92–99, 2016.
20. S. Chakrabarti, M. Berg, and B. Dom, "Focused crawling: A new approach to topic-specific web resource discovery," in *Proc. 8th Int. World Wide Web Conf.*, Toronto, ON, Canada, 1999, pp. 1623–1640.
21. S. Chakrabarti, K. Punera, and M. Subramanyam, "Accelerated focused crawling through online relevance feedback," in *Proc. 11th Int. Conf. World Wide Web*, Honolulu, HI, USA, 2002, pp. 148–159.
22. C. Olston and M. Najork, "Web crawling," *Found. Trends Inf. Retrieval*, vol. 4, no. 3, pp. 175–246, 2010.
23. DARPA's Memex Project. [Online]. Available: <http://www.darpa.mil/program/memex>
24. X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in *Proc. Int. Joint Conf. Artif. Intell.*, Acapulco, Mexico, 2003, pp. 587–592.
25. H. Yu, J. Han, and K. Chang, "PEBL: Web page classification without negative examples," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 70–81, Jan. 2004.
26. F. Li and Y. Yang, "A loss function analysis for classification methods in text categorization," in *Proc. 20th Int. Conf. Mach. Learn.*, Washington, DC, USA, 2003, pp. 472–479.
27. Boilerpipe Library. [Online]. Available: <https://code.google.com/archive/p/boilerpipe/>
28. C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proc. 3rd ACM Int. Conf. Web Search Data Min.*, New York, NY, USA, 2010, pp. 441–450.
29. IBM Watson Retrieval and Rank Service. [Online]. Available: <https://www.ibm.com/watson/developercloud/retrieve-rank.html>
30. A. Dasgupta, A. Ghosh, R. Kumar, C. Olston, S. Pandey, and A. Tomkins, "The discoverability of the web," in *Proc. 16th Int. Conf. World Wide Web*, Banff, AB, Canada, 2007, pp. 421–430.
31. A. Santos, B. Pasini, and J. Freire, "A first study on temporal dynamics of topics on the web," in *Proc. 25th Int. Conf. Companion World Wide Web*, Montréal, QC, Canada, 2016, pp. 849–854.
32. J. Cho and H. Garcia-Molina, "Effective page refresh policies for web crawlers," *ACM Trans. Database Syst.*, vol. 28, no. 4, pp. 390–426, 2003.
33. J. Cho and H. Garcia-Molina, "Estimating frequency of change," *ACM Trans. Internet Technol.*, vol. 3, no. 3, pp. 256–290, 2003.

Received January 10, 2017; accepted for publication February 4, 2017

Kien T. Pham Department of Computer Science and Engineering, NYU Tandon School of Engineering, NY 11201 USA (kien.pham@nyu.edu). Mr. Pham is a Ph.D. candidate in the Computer Science and Engineering Department at the NYU Tandon School of Engineering. He received a B.Sc. degree in computer science from Hanoi University of Science and Engineering (Vietnam) in 2010. Prior to joining the graduate school, he worked as a research and development engineer, developing infrastructure for an online social network and an Internet startup. Mr. Pham's current research focuses on information retrieval and machine learning, in particular working with web search and web mining.

Prasanna Sattigeri IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (psattig@us.ibm.com). Dr. Sattigeri is a Research Staff Member in the Data Science Department at the IBM T. J. Watson Research Center. He received

his Ph.D. degree in electrical engineering from Arizona State University. His broad research interests include developing theory and algorithms for representation learning of high-dimensional data and building systems for semantic inferences. He is also interested in developing scalable solutions and has experience shipping machine-learning products to consumers.

Amit Dhurandhar *IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (adhuran@us.ibm.com).* Dr. Dhurandhar is a Research Staff Member in the Mathematical Sciences Department at the IBM T. J. Watson Research Center. He received his B.E. degree in computer engineering from Pune University in 2004. He then received his masters and Ph.D. degrees in computer engineering from the University of Florida in 2005 and 2009, respectively. Dr. Dhurandhar's research interests primarily span the areas of machine learning, data science, and computational neuroscience. He has authored several papers in top machine-learning and data-mining venues, receiving a Deployed Application Award from the Association for the Advancement of Artificial Intelligence. He also had an IEEE International Conference on Data Mining best paper candidate submission. He also has multiple granted patents and has served on National Science Foundation grant panels.

Arpith C. Jacob *IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (acjacob@us.ibm.com).* Dr. Jacob is a Research Staff Member in the Advanced Compiler Technologies Group at the IBM T. J. Watson Research Center. His interests include parallelizing compilers and special-purpose accelerators.

Maja Vukovic *IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (maja@us.ibm.com).* Dr. Vukovic is a Research Manager of the Cloud Services Automation organization at IBM T. J. Watson Research Center. She received her Ph.D. degree from the University of Cambridge, UK, for her work on context-aware service composition using AI planning. Her research expertise is in service automation, cloud transformation, crowdsourcing technologies, and API ecosystems. She has received four IBM Outstanding Technical Achievement Awards and three IBM Research awards for her technical leadership and contributions to enterprise crowdsourcing and innovations in IT service management. Dr. Vukovic is an IBM Master Inventor and a Member of the IBM Academy of Technology. She has more than 80 publications in top international venues. She is a Senior Member of the IEEE.

Patrice Chataigner *ACAPS (Assessment Capacities Project), CH-1202 Geneva, Switzerland (pc@acaps.org).* Mr. Chataigner has 19 years of relevant emergency and preparedness experience with complex emergencies and for a wide range of clients, including governments, the United Nations System, and international non-governmental organizations. His recent assignments include the head of methodology and innovation for the ACAPS project (for 7 years), the emergency desk for Action Against Hunger Spain, the emergency desk for Solidarités Internationale, and assessment training facilitator for ACAPS. His work also involves multiple software developments, including secondary data review and questionnaire design for rapid assessments and training on coordinated assessment, analytical thinking, data analysis, and data visualization.

Juliana Freire *New York University, New York, NY 10003 USA (juliana.freire@nyu.edu).* Dr. Freire is a Professor of Computer Science and Engineering and Data Science at New York University. Dr. Freire

holds an appointment at the Courant Institute for Mathematical Science and is a faculty member at the NYU Center for Urban Science and at the NYU Center of Data Science, where she is also the Director of Graduate Studies. She is the executive director of the NYU Moore-Sloan Data Science Environment. Her recent research has focused on big-data analysis and visualization, large-scale information integration, web crawling and domain discovery, provenance management, and computational reproducibility. Dr. Freire is an active member of the database and web research communities, with over 170 technical papers, several open-source systems, and 11 U.S. patents. She is an ACM Fellow and a recipient of an NSF CAREER award, two IBM Faculty awards, and a Google Faculty Research award. She has chaired or co-chaired workshops and conferences, and participated as a program committee member in more than 70 events. Her research grants are from the National Science Foundation, DARPA, Department of Energy, National Institutes of Health, Sloan Foundation, Gordon and Betty Moore Foundation, W. M. Keck Foundation, Google, Amazon, AT&T, the University of Utah, New York University, Microsoft Research, Yahoo! and IBM.

Aleksandra Mojsilović *IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (aleksand@us.ibm.com).* Dr. Mojsilović is an IBM Fellow and scientist at the IBM T. J. Watson Research Center. She received her Ph.D. degree in electrical engineering from the University of Belgrade, Serbia, in 1997. She was a Member of Technical Staff at the Bell Laboratories from 1998 to 2000, and then joined IBM Research, where she currently leads the Data Science Department. Dr. Mojsilović is a founder and co-director of the IBM Science for Social Good initiative. Her research interests include multidimensional signal processing, predictive modeling, and pattern recognition. She has applied her skills to problems in computer vision, healthcare, multimedia, finance, human resources, public affairs and economics. She is one of the pioneers of business analytics at IBM and in the industry; throughout her career she championed innovative uses of analytics for business decision support. Dr. Mojsilović was appointed an IBM Fellow, the company's highest technical honor. She is the author of over 100 publications and holds 16 patents. Her work has been recognized with several awards including IEEE Signal Processing Society Young Author Best Paper Award, Institute for Operations Research and the Management Sciences (INFORMS) Wagner Prize, IBM Extraordinary Accomplishment Award, IBM Gerstner Prize, and Best Paper awards at the European Conference on Computer Vision (ECCV) and the Conference on Service Operations and Logistics, and Informatics (SOLI). She is an IEEE Fellow and a member of INFORMS and Society of Women Engineers.

Kush R. Varshney *IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (krvarshn@us.ibm.com).* Dr. Varshney is a Research Staff Member and manager in the Data Science Department at the IBM T. J. Watson Research Center. He received his Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology in 2010. He applies data science and predictive analytics to human capital management, healthcare, olfaction, public affairs, and international development. He conducts academic research on the theory and methods of statistical signal processing and machine learning. His work has been recognized through best paper awards at the 2009 International Conference on Information Fusion, 2013 IEEE Conference on Service Operations and Logistics, and Informatics (SOLI), 2014 ACM SIGKDD Conference on Knowledge Discovery and Data Mining, and 2015 SIAM (Society for Industrial and Applied Mathematics) Conference on Data Mining (SDM). Dr. Varshney is codirector of the IBM Science for Social Good initiative.