Real-Time Streaming Intelligence: Integrating Graph and NLP Analytics

David Ediger Scott Appling Erica Briscoe Rob McColl Jason Poovey
Georgia Tech Research Institute
Atlanta, Georgia

***Abstract*—With the growth of social media, embedded sensors, and "smart" devices, those responsible for managing resources during emergencies, such as weather-related disasters, are transitioning from an era of data scarcity to data deluge. During a crisis situation, emergency managers must aggregate various data to assess the situation on the ground, evaluate response plans, give advice to state and local agencies, and inform the public. We make the case that social graph analysis and natural language modeling in real time are paramount to distilling useful intelligence from the large volumes of data available to crisis response personnel. Using ground truth information from social media data surrounding the 2012 Hurricane Sandy in New York City, we test and evaluate our real-time analytics platform to identify immediate and critical information that increases situational awareness during disastrous events.**

I. Introduction

Though social media is primarily a source of entertainment value, during significant events, open, social communication platforms are likely to possess information vital to emergency managers responsible for crisis response. Often information providers are unaware of the civic value of their contributions, making it difficult for those searching for timely, localized reports. During a significant event, social media also becomes a critical source of information for citizens, who use these platforms as a means to gain understanding about the current status of an event, either from official or unofficial sources. Indeed, multiple studies across cultures [1], [2] have provided evidence that several kinds (e.g. situational, action-oriented, operational) of important communications take place in this new media.

While the timely dissemination of information during a crisis is an incredible benefit of new media, to fully realize its value, a host of technical issues must be overcome. These issues arise from many of the key characteristics of social media, specifically, the large quantity of information that must be quickly processed and the informal generation of the content contained therein. The grammar of online social media necessitates specialized language processing. The volume of social artifacts generated during a disaster requires automated methods to intelligently filter content. Detecting the messaging characteristics of large-scale information diffusion is important for emergency managers. It is these challenges that we discuss as being successfully addressed in our recently developed system for situational awareness during crises.

II. Related Work

In recent years the rapid adoption of social computing platforms has created massive amounts of data and has generated interest in several computing fields: distributed computing, high performance computing, and artificial intelligence to name a few. The goal of these platforms is to do efficient and large-scale computations over the massive data sets generated by sources such as social media. Traditional platforms have primarily focused on batch mode processing and forensic analysis; however with the volumes of data now being generated, and the need for real-time actionable intelligence, these platforms are now geared toward real-time streaming analytics.

The authors of [3] report on work conducted toward the design of a module called DataCell, as an extension to the open-source MonetDB column store project, to provide online analytics. The authors write that their goal has been to "fully exploit the generic storage and execution engine of the DBMS" so that "[s]tream processing becomes primarily a query *scheduling* task". These efforts center around the scenario where "one or more continous queries [are] waiting for incoming streams" and data from separate streams are combined and analytics performed in real-time, before sending their results to clients.

The authors of MOA (massive online analytics) [4] have focused their attention on the scenario involving online learning algorithms that are constantly updating their model parameters using large data streams in real time to provide the best classification and regression estimates. These frameworks represent a growing body of work on the effective utilization of streaming data towards problem solving.

Recently, with the growing interest of the humanitarian community of first responders and emergency managers, new computational frameworks have been proposed by researchers from both the computer science and humanitarian response communities. [5] recently introduced AIDR, a platform for utilizing social media data for disaster response. The platform allowed individuals to automatically classify social media texts as belonging to categories of interest that could be created by individuals running the platform.

III. Social Analytics Dashboard

The urgency with which crisis responders must operate necessitates an ability to readily comprehend complex and

978-1-4799-6233-4/14/$31.00 ©2014 IEEE

integrated information [6]. These personnel are required to make immediate decisions based on all data at hand, such as deciding the appropriate allocation of resources. The burden of integrating information can be lessened by the use of computer-based tools designed to reduce the cognitive load of users and by reducing the amount of time they must utilize to locate and assemble relevant information.

These cognitive demands are intensified with the staggering volume of information that arises from including social media data as an information source during crisis response. While some approaches to lessening this effort exist, such as crowd-sourcing social media search during crises, these processes are time-consuming and require sophisticated management. Our solution is to provide a user-friendly dashboard presentation that showcases the underlying powerful analytics that are capable of distilling this information into actionable intelligence. This interface allows users to readily comprehend only the critical information that is relevant to them at any given time. For example, a user may need to concentrate on the geographical location of tweets in the "incident" category (such as traffic accidents) at some point in time and later need to focus only on the dissemination of their public messaging (such as their instructions to the public).

IV. Natural Language Analytics

Our framework uses statistical natural language understanding models to provide situational awareness via topic and complex sentiment prediction. These models examine the language used in text samples to predict phenomena relevant to stakeholder interest in real-time. These predictions include classifications about topics of interest and the sentiment being expressed by individuals and groups.

A. *Deriving Meaning from Text During Disasters*

Effectively responding to a disaster requires successful mitigation of existing disaster-related challenges through planning and coordination to avoid anticipated as well as unknown future challenges related to unfolding circumstances. Weather-related disasters are one class of disaster that allow for citizen populations to utilize online social networks to communicate their own personal statuses, accidents, and observations about any number of facets related to a specific weather event. This means that if one were to log in to their favorite social media network, during a large weather event, they would likely see many artifacts related to what people were thinking and doing before, during and after the weather event, e.g. Hurricane Sandy in New York City.

The problem for individuals utilizing social media for disaster response is that while they can review current streams of social media artifacts, the sheer magnitude of individuals producing these artifacts, (whether or not relevant to the current event) drastically reduces traditional means of doing analyses, i.e. there is not enough time for an individual or group to read and make sense out of such a large number of artifacts during a disaster response. Current trends in social computing usage would only tend to support the case that more social media artifacts will be produced, amplifying the problem.

TABLE I
Stages of Disaster

Stage	Name	Information Description
1	Warning	That which relates to the potential effects of a perceived disaster, or potential response before event horizon.
2	Threat	That which reports or describes what people are doing or should do in order to survive.
3	Impact	That which reports or describes what is happening or did happen during the impact.
4	Inventory	Reports, descriptions, or assessments of how bad the effects of the disaster are affecting people or areas during or afterwards.
5	Rescue	Reports or descriptions of rescues planned e.g. first-aid, or individuals helping one another.
6	Remedy	Reports of government agencies taking large scale actions to remedy large issues.
7	Recovery	Reports or descriptions of an individuals recovery taking place or, the communitys property being recovered after the event e.g. power returning, finding shelter, assistance.

One approach to this big data problem has been to create tailored lists of keywords that can be used to search through and filter many of the non-relevant artifacts. This approach does successfully reduce the magnitude of artifacts exposed to an analyst, but fails to capture inherent qualities of social media:

- Artifacts collected from keyword searches may not be relevant to the event itself (e.g. rhetorical tweets, jokey language)
- Ambiguous word usage returns artifacts on other topics (e.g. Hurricane may also refer to a sports team name)
- Traditional lexical keyword searches miss semantically related content not captured by lexical tokens alone (e.g. searching for "storm" will return artifacts with the word storm in them but not hurricane or rainstorm, although highly relevant).

Our approach to making meaning out of these large texts and at the same time solve the filtering problem is to train supervised learning models to make meaningful domain-relevant classifications and consequently recognize what is not useful to the domain of interest. These models allow us to provide automated classifications about the domain in real-time to create decision support analytics and filter massive text streams.

Even still, solving the filtering problem alone does not meaningfully organize the large amounts of now relevant information. For the field of disaster response and emergency management we adopt two kinds of taxonomy systems. Using multiple taxonomies means that we can provide intelligence to multiple groups depending on their needs simultaneously. One useful taxonomy from the field of disaster and crisis response used to structure thinking about disaster-related events is Powell and Rayners Stages of Disaster taxonomy [2]. Table I provides concise descriptions of these stages as they relate to the status of populations facing and responding to specific disaster events.

Using the Stages of Disaster taxonomy, we take social media artifacts and classify them as they relate to particular stages. For example, Tweets before Hurricane Sandy about impending water damage likely fit the Warning stage while directions for how to waterproof doors due to flash flooding fit the Threat stage.

In addition to the Stages of Disaster, the United States Department of Homeland Security defines the term Essential Elements of Information as, "key information on the nature and scope of damages...such as: disaster boundaries, socio-economic impacts, and status of communications, transportation systems, and critical infrastructure" [7]. Social media artifacts created during disaster events tend to contain information that can be used to substantiate EEIs. We provide a classification model that provide multiple labels for social media artifacts according to EEIs they may likely support. For example, the social media artifact "Downed trees and power lines block Church Street in #Haledon #njsandy #hpsandy" receives the Infrastructure, Incident, and Transportation labels. Using the same training data we collected for the Stages of Disaster classification model, we trained four models reflecting different types of EEIs: Infrastructure, Incidents, Operations, and Transportation[1].

B. Complex Sentiment Analysis

Social media artifacts, employing large amounts of colloquial language, are often incompatible with existing off-the-shelf sentiment analysis tools and incapable of handling more than simple valence score assessments and averaging. Our sentiment analysis approach [8], allows us to better capture language complexities (e.g. "not too bad!" is a pretty positive sentiment) than off the shelf approaches by employing single word valence calculations and averaging. See [8] for evaluation results over a number of different corpora.

C. Evaluation

To understand the predictive quality of the natural language models on disaster stages and EEIs, we evaluate through precision and recall using a 10-Fold cross validation methodology; Figure 1 shows the results of our best classifier, a linear kernel SVM, using a training set of 5000 social media artifacts collected during the Hurricane Sandy weather event of 2012 that occured in and around the New York City area. Training data comes to us in the form of human-annotated social media artifacts with ground truth information about the appropriate categories each artifact. We use a majority vote approach to decide whether or not each artifact receives a particular label. When an artifact belongs to no relevant categories we label it off topic and use it for preparing an off topic model to initially filter future data removing irrelevant conversation.

[1]Due to a lack of identified training examples for the transportation EEI category we do not evaluate it here.

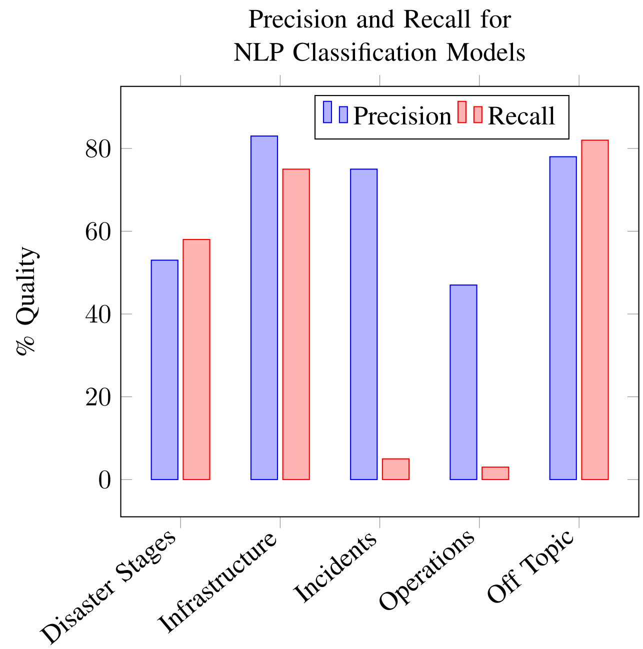

Fig. 1. Precision and Recall scores for 4 disaster-related NLP models using a Linear Kernel SVM. Training data comes from human annotated social media artifacts about each category. The "Off Topic" model provides principal support for whether or not artifacts are processed in later stages. Depending on the number of positive training examples, as high as 1k (Infrastructure), and as low as 260 (Operations), prediction quality can scale.

Text classification is the first step in the analytics pipeline. Positive and negative sentiment scores are calculated for each artifact. Artifacts are then automatically classified and tagged with the appropriate labels for Stages of Disaster and EEIs. We observe classification speed as fast as 40 milliseconds per artifact using a batch size of 1000 social media artifacts per classification.

V. Real-time Streaming

In a disaster response scenario, response time is a critical factor for effective understanding and amelioration of emerging situations. Dealing with massive streaming data sources requires computation to be fast and efficient. We handle these requirements with a real-time streaming analytics framework that takes in continuous streams of event information in the form of social media artifacts. Our framework performs real-time processing to: 1) extract social network graphs based on the interactions of the individuals originating each artifact (see Section VI) and 2) derive meaning from the textual content of each artifact through natural language understanding (see

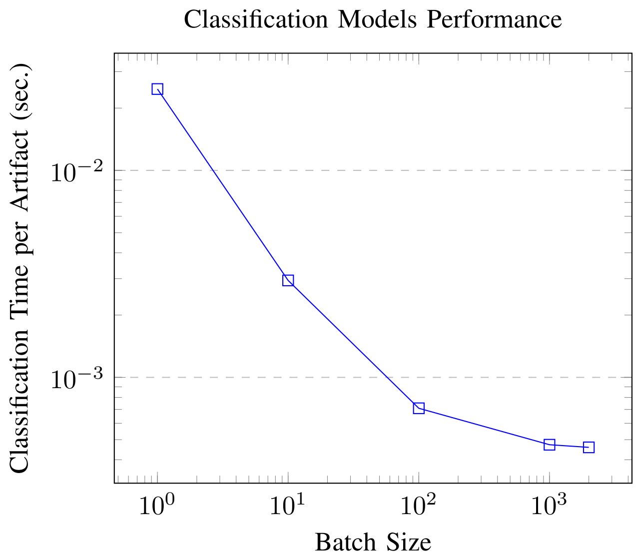

Fig. 2. Average classification time per social media artifact by increasing batch sizes. Classification time per artifact is drastically reduced from the individual case starting with a batch size of 100 artifacts and reaches a 2x improvement again at roughly a 2k batch size. These performance times are running over 6 models.

Section IV). These two methods are easily combined to isolate interactions based on subject of discussion.

Social network graphs allow for real-time analysis of community structure and influence and also provide a context that can be used to intrepret the the textual analytics. To build this capability an interaction network is derived from a Tweet data stream and is dynamically updated in a streaming graph database. This graph database then processes the real time analytic algorithms and publishes the result to the dashboard via a RESTful API.

The streaming graph database used for the Social Analytics Dashboard is STINGER [9]. STINGER is an open source software project that uses an in-memory graph storage and analysis platform for high performance [10] graph analytics. On commodity servers, STINGER is capable of more than one million updates to the graph structure per second where the possible size of the graph is dependent upon the main memory size of the machine. Current-generation servers now capable of terabytes of main memory are prime candidates for storing massive amounts of social data in real time using STINGER. In a two-terabyte system, a graph with 100 billion edges (representing 100 billion user interactions) can be represented and processed at real-time speed.

In the STINGER framework, there are several types of clients. Data “streams” are processes that parse semi-structured data and convert social artifacts into edge and vertex updates. Edge and vertex updates are sent to the server asynchronously over a defined protocol. Streams can be distributed across many front-end systems.

Client “algorithms” are processes that compute on the graph and the update stream. Algorithm processes memory map a read-only copy of the STINGER data structure and a batch of edge and vertex updates. Algorithms separate their incremental computation into pre-update and post-update phases. In between phases, the server applies the edge and vertex updates to the graph. Algorithm implementations are encouraged to perform the minimal set of incremental updates to their state to maintain performance and reduce latency.

“Monitor” processes are a special class of algorithms that have read-only access to all algorithm states and the graph in memory. These processes are used for HTTP endpoints as well as periodic backups. STINGER uses the JSON-RPC standard over HTTP. The JSON-RPC monitor process handles client requests for algorithm and graph data, performing sorting, sampling, and filtering.

The STINGER server coordinates all running processes and the mapping of shared data. In addition to maintaining the in-memory database, the server resolves physical to logical names, passes edge and vertex updates from the ingest streams to the client analytics, schedules client analytics, and resolve dependencies.

For an overview of the STINGER architecture, please refer to [11]. STINGER is free and open source software available at stingergraph.com.

VI. Graph Analytics

We construct a social network from Tweets expressing mentions and retweets as edges between entities. We use different edge types to indicate the “mentioned” action and the “mentioned by” relationship. Using natural language features discussed in Section IV, edges are also grouped by Tweet label (Stages of Disaster and EEI) and vertices are weighted by sentiment. Incorporating natural language features in this manner enables the graph features discussed in this section to be analyzed across NLP-defined dimensions in real time. For performance testing, we use an undirected, untyped graph.

A. *PageRank*

PageRank, or eigenvector centrality, is an algorithm for identifying influence in a social network [12]. The algorithm measures the likelihood of arriving at a vertex during a random walk about the graph. The solution is implemented using power iteration and converges at about 100 iterations. After a new batch of edges is applied to the graph, the computation is restarted using the previous PageRank scores as priors.

B. *Betweenness Centrality*

Betweenness centrality is a social network analysis metric that has been successfully applied in areas modeling disease spread, transportation, and bioinformatics [13], [14]. Betweenness centrality identifies entities that connect large portions of the network and form the backbone of the graph structure. Removing these individuals does significant structural damage to the network connectivity. We implement a sampling approximation [15] that chooses 256 vertices at random during each update phase. This algorithm has successfully identified major media outlets and local influencers in disaster-related social networks [16].

C. Degree Velocity

The degree of a vertex is the number of edges incident on that vertex, or the number of neighbors. At each time step, we can measure the degree of a vertex. Taking the first derivative of the degree with respect to time, we arrive at the degree velocity [17]. This statistic measures the rate at which the degree of vertex is changing, or how fast edges are being inserted or removed incident on this vertex. The second derivative measures degree acceleration. Degree velocity differentiates between vertices whose neighborhood is continuously growing over time versus those whose impact is due to a single event.

D. Performance

We evaluate performance of the graph analytics by measuring the time taken to update each part of the system on a batch of new edges. For the data structure, STINGER, the time taken to process the insertion of each of the new edges in the batch is measured. For the analytic algorithms, we measure the time to update the result value for each vertex in the graph. We run all components of the system for each trial, and we measure the time for all components. The STINGER graph data structure and each analysis algorithm is running as separate parallel process on the same machine. The machine used for testing is a dual-socket Intel Xeon E5-2670. It has a total of 16 hyperthreaded cores running at 2.60 GHz and 64 GiB of main memory. The results plotted in Figures 3 and 4 are the average of five runs. Edge updates are processed in batches of 5000 to increase parallelism and amortize the overhead of transmitting edges between processes.

A total of 53 batches of 5000 edges were ingested from the Hurricane Sandy Twitter dataset. The average time required to insert and update 5000 edges in the STINGER data structure was 2.89 milliseconds and the median was 2.85 milliseconds. This translates to an update rate of 1.73 million updates per second.

The betweenness centrality kernel had an average update time of 214 milliseconds per 5000 edges and a median update time of 193 milliseconds, and the update rate was 23,400 updates per second. The degree velocity kernel had an average update time of 24.2 milliseconds per 5000 edges and a median update time of 24.4 milliseconds, and the update rate was 207,000 updates per second. The PageRank kernel had an average update time of 132 milliseconds per 5000 edges and a median update time of 126 milliseconds, and the update rate was 38,000 updates per second.

The time taken to process the batch of updates in the STINGER data structure and the degree velocity kernel is independent of the graph size in this experiment, as seen in Figure 3. However, the time taken to process the update of the PageRank kernel grows with each batch. The complexity of the STINGER update is proportional to the average degree of the graph, and the complexity of the degree velocity is proportional to the number of vertices in the graph. For PageRank, the update time is related to the number of new vertices, the number of new edges, and the structural changes that both have on the graph. These structural changes cause "influence" to be iteratively spread around the network. If there is little change, the update will be much faster.

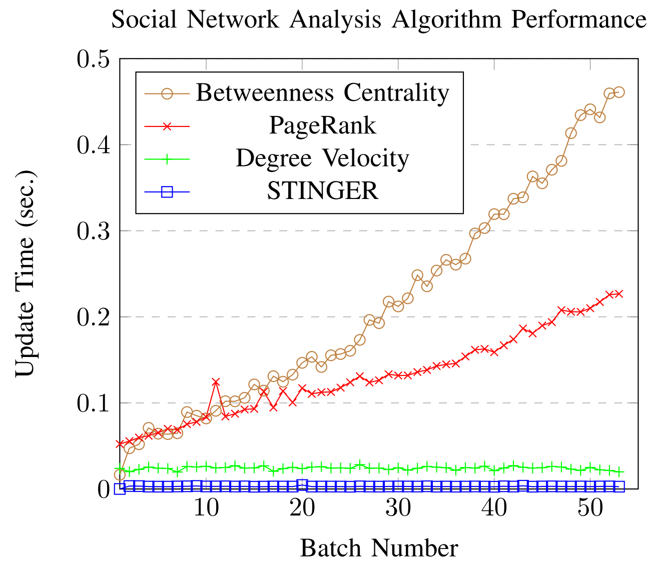

Fig. 3. Batches of 5000 edge updates. Average of five trials. Experiments conducted on an dual-socket Intel Xeon E5-2670 @ 2.60 GHz with 64 GiB main memory.

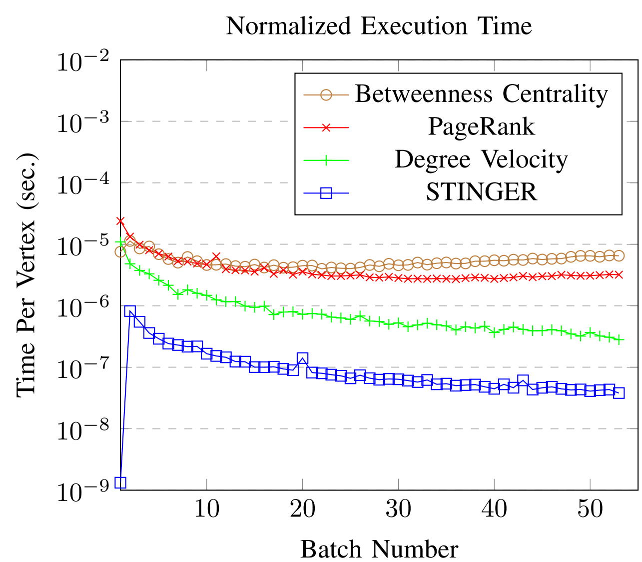

Fig. 4. Batches of 5000 edge updates. Average of five trials. Times normalized by the number of the vertices in the graph at that time step. Experiments conducted on an dual-socket Intel Xeon E5-2670 @ 2.60 GHz with 64 GiB main memory.

In Figure 4, the update time is normalized by the number of vertices in the graph at each time step. For STINGER and degree velocity, the time decreases as work is amortized by the larger graph. PageRank drops until batch 40 and begins a slow rise. In testing with larger artificial graphs, we see the update time for a 10,000 edge batch settles around 1 second even with a graph containing 18 million edges and 125,000 vertices.

VII. Conclusions and Future Work

SocialAI provides an integrated platform that integrates several aspects of real-time analytics of social data into actionable intelligence during crises. As far as our knowledge there is no other system that combines real-time streaming graph and natural language analytics in a manner that performs as well in terms of processing speed, memory usage, and classification quality. The framework, SocialAI, will continue to be expanded into other domains with a continued focus on computational social science to provide meaningful theory-supported analyses to explain and predict individual and group behavior.

References

[1] Y. Qu, C. Huang, P. Zhang, and J. Zhang, "Microblogging after a major disaster in china: a case study of the 2010 yushu earthquake," in *Proceedings of the ACM 2011 conference on Computer supported cooperative work*. ACM, 2011, pp. 25–34.

[2] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: Can we trust what we RT?" in *Proceedings of the First Workshop on Social Media Analytics*, ser. SOMA '10. New York, NY, USA: ACM, 2010, pp. 71–79.

[3] E. Liarou, S. Idreos, S. Manegold, and M. Kersten, "Monetdb/datacell: online analytics in a streaming column-store," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 1910–1913, 2012.

[4] A. Bifet, G. Holmes, B. Pfahringer, J. Read, P. Kranen, H. Kremer, T. Jansen, and T. Seidl, "Moa: a real-time analytics open source framework," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 617–620.

[5] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "AIDR: Artificial Intelligence for Disaster Response," in *Proceedings of the companion publication of the 23rd international conference on World wide web companion*. International World Wide Web Conferences Steering Committee, 2014, pp. 159–162.

[6] M. T. Gervasio and W. Iba, "Crisis response planning: A task analysis," in *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, vol. 929, 1997.

[7] U. S. D. of Homeland Security, "GeoCONOPS: Geospatial concept of operations," 2013.

[8] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. of the 8th International AAAI Conference on Weblogs and Social Media*, 2014.

[9] D. A. Bader, J. Berry, A. Amos-Binks, D. Chavarría-Miranda, C. Hastings, K. Madduri, and S. C. Poulos, "STINGER: Spatio-Temporal Interaction Networks and Graphs (STING) Extensible Representation," Georgia Institute of Technology, Tech. Rep., 2009.

[10] D. Ediger, R. McColl, J. Riedy, and D. A. Bader, "STINGER: High performance data structure for streaming graphs," in *The IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, MA, Sep. 2012.

[11] D. Ediger, R. McColl, J. Poovey, and D. Campbell, "Scalable infrastructures for data in motion," in *Scalable Computing for Real-time Big Data Applications (SCRAMBL), 1st International Workshop on*, May 2014, pp. 1–8.

[12] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 17, pp. 107–117, 1998, proceedings of the Seventh International World Wide Web Conference.

[13] H. Jeong, S. Mason, A.-L. Barabási, and Z. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, pp. 41–42, 2001.

[14] R. Guimer, S. Mossa, A. Turtschi, and L. A. N. Amaral, "The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 22, pp. 7794–7799, 2005.

[15] D. Bader, S. Kintali, K. Madduri, and M. Mihail, "Approximating betweenness centrality," in *Proc. 5th Workshop on Algorithms and Models for the Web-Graph (WAW2007)*, ser. lncs, vol. 4863. San Diego, CA: springer, December 2007, pp. 134–137.

[16] D. Ediger, K. Jiang, J. Riedy, D. A. Bader, C. Corley, R. Farber, and W. N. Reynolds, "Massive social network analysis: Mining Twitter for social good," *Parallel Processing, International Conference on*, pp. 583–593, 2010.

[17] J. Fairbanks, D. Ediger, R. McColl, D. A. Bader, and E. Gilbert, "A statistical framework for streaming graph analysis," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ser. ASONAM '13. New York, NY, USA: ACM, 2013, pp. 341–347.