CS6365 - Introduction to Enterprise Computing

News Source Validation for LITMUS

Final Report

Version 1.0
April 24th, 2015

Presented by:
Uy Nguyen, Nakul Patel

Motivation and Objective

It is difficult to detect landslides effectively. Current research into this area is focused on targeting data from many different sources including seismic activity and even extends to social media. In particular, a system being developed at Georgia Tech named LITMUS (Landslide Detection by Integrating Multiple Sources) aggregates and analyzes real-time social media data to determine an instant verification of a landslide occurrence. However, the problems facing landslide detection utilizing these sources lies in the sources' reliability and also the "noise" contained within the information. For instance, a search for "landslide" on Twitter can lead to results containing irrelevant information that is needed. An example would include a "landslide victory" for a political election. Filtering of these sources is needed to extract the desired data relating to the type of landslides associated with disasters. Furthermore, the main problem with this system is that there are no real time authoritative sources that could help detect a landslide or help validate an analysis done on social media data.

We proposed a system that extracts a set of reliable news articles for use in evaluation in the LITMUS system. To begin, the team was tasked in evaluating news sources on the web today. The USGS (U.S. Geological Survey) webpage provides a good starting point for discovering authoritative and trusted sources. Source evaluation is based on criteria including frequency of news items, noise of news items, availability of an API, web ranking, etc. From these news sources, we then filter out news items pertaining to landslide data which we hope to use as an extra source for validation for LITMUS.

Related Work

For many people today, Twitter/Facebook is their primary source of news. News stories usually circulate on social media much earlier than it does on official news sites. Due to its popularity, disaster detection using social media has drawn a lot of attention recently. There are many research studies going on for detection of various disasters. However, most focus is on a single social network, usually Twitter. In addition, there are also studies which use weather data and sensors in selected regions to detect landslides and other disasters.

Listed here are a few notable projects:

- *LITMUS (Landslide Detection by Integrating Multiple Sources)* [1] - A system being developed at Georgia Tech which aggregates and analyzes real-time social media data to determine an instant verification of a landslide occurrence. LITMUS integrates multiple sources to detect landslides including social sensors (Twitter, Instagram, and YouTube) and physical sensors (USGS seismometers and TRMM satellite). The data from social sensors is filtered various ways (key words, smart geo-tagging, machine learning, blacklist URLs, etc) and processed by LITMUS along with physical sensor data to produce a list of detected landslides.

- *Twitter Earthquake Detector* (TED) [2] - This is an effort by USGS in developing a system that gathers real-time, earthquake-related messages from the social networking site Twitter and applies place, time, and keyword filtering to gather geo-located accounts of shaking. This approach provides rapid first-impression narratives and, potentially, images from people at the hazard's location. With this research, they are also investigating the potential for earthquake detection in populated but sparsely seismically-instrumented regions.

- *Real-time Event Detection by Social Sensors* [3] - Research being done at the University of Tokyo similarly are investigating the real-time interaction of events such as earthquakes, in Twitter, and propose an algorithm to monitor tweets and to detect a target event.

- *Real-time Wireless Sensor Network for Landslide Detection* [4] - This paper discusses the development of a wireless sensor network(WSN) to detect landslides, which includes the design, development and implementation of a WSN for real time monitoring, and the development of the algorithms needed that will enable efficient data collection and data aggregation.

- *Detection of Landslide Using Wireless Sensor Networks* [5] - Similar to above, this research is being done at Central Institute of Mining and Fuel Research also looks into the development of a wireless sensor network (WSN) to detect landslides, which includes design and development of WSN for real time monitoring system.

Project Scope

The challenge with all above projects is validating the final result. The purpose and scope of our project is to provide real-time authoritative new sources to LITMUS for validation of its social media findings.

The work for this project begins with gaining an understanding of how and where the USGS obtains their information when they report a landslide. Apart from the sources used by USGS, we also looked at top 50 news sites rated by Alexa, which is a global traffic ranking website.

It is important to gain insight into what keywords and events contribute to a landslide news story. Thus, we examine the above sources and composed a ranking of the sources based on these criterias:

- RSS Feed Availability
- Frequency of update
- Geographic scope coverage
- Site Ranking (via Alexa)
- Amount of Noise (irrelevant information)
- API Availability

Implementation

From the above criteria, RSS feed availability dictated what sources to include in our system. We noticed that very few sources provided APIs, and if they did, they weren't free. Hence, we decided to obtain news via RSS Feed. An advantage of RSS Feeds is that it is an internet standard for monitoring up-to-date and published content. This allows us to make scaling our application up easier by being able to add many more news sources in the future.

This takes care of the top news sites in the world. But how do we capture news from the lesser known sites around the world? To fill in this gap we search Google and Bing Alerts/News for recent and then relevant articles. Currently our system takes in feeds from total 76 sources.

Our application is written in Java, the front-end is written using HTML/Javascript and we utilize the following libraries:

JDOM - a solution for accessing, manipulating, and outputting XML data from Java code.

JSOUP - a Java library for working with real-world HTML. It provides a very convenient API for extracting and manipulating data, using the best of DOM, CSS, and jquery-like methods.

ROME - a set of RSS and Atom Utilities for Java that is open source under the Apache 2.0 license.

Logback - generic, fast and flexible logging library for Java.

SLF4J - a simple facade or abstraction for various logging frameworks (e.g. java.util.logging, logback, log4j) allowing the end user to plug in the desired logging framework at deployment time.

OpenCSV - a very simple csv (comma-separated values) parser library for Java.

GSON - a Java library that can be used to convert Java Objects into their JSON representation.

Google Maps API - a wide array of APIs that let you embed the robust functionality and everyday usefulness of Google Maps into your own website.

SIMILE Timeline API - an open-source library to visualize temporal information on an interactive drag-able timeline.

Alchemy API - A natural language processing API that processes text content and adds high level-semantic information.

Workflow

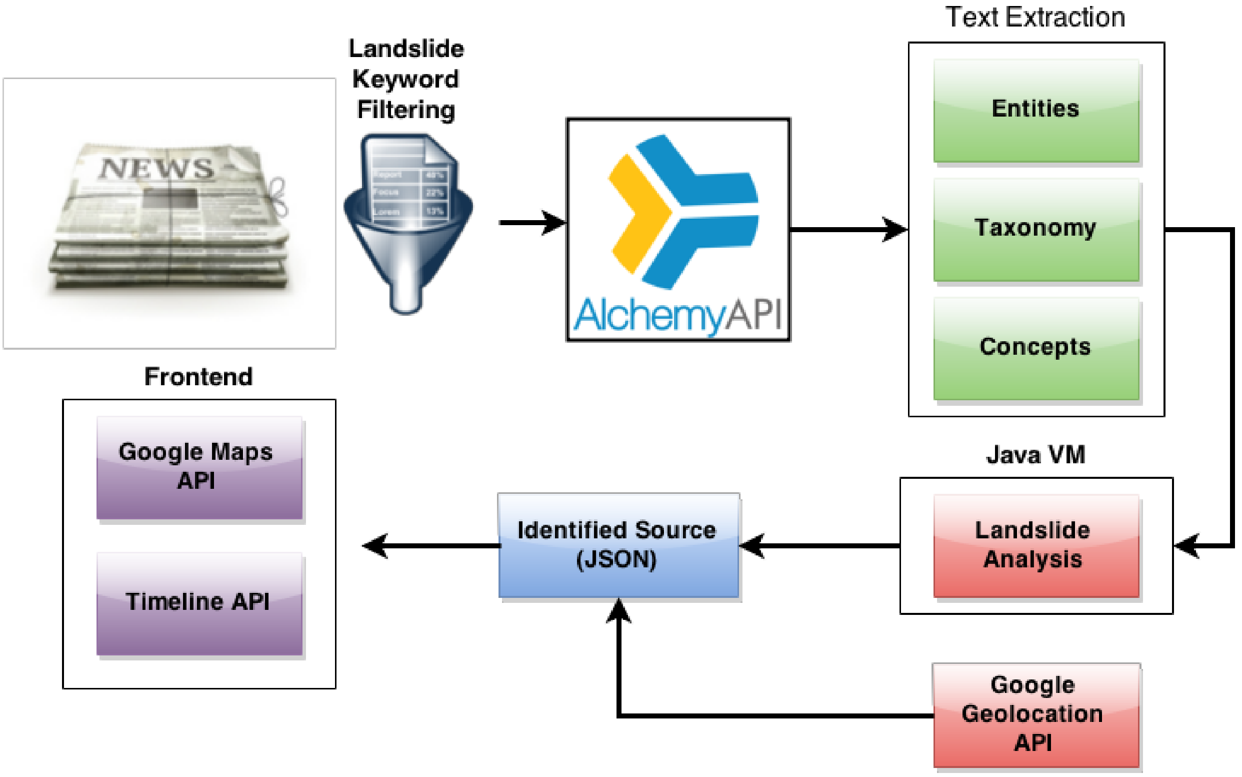


Figure 1. High level view of LITMUS News system

To begin, our application first retrieves latest news articles from the sources using RSS Feed. Next we extract the title, description, link to the article and the published date from the feed message. Each feed message is then examined to check if the article was published after the last time we pulled from the source or is it an old article. The recent articles then go through keyword filtering where an article is checked for common keywords relating to a landslide disaster. The keywords that we have chosen to check for are mudslide, rockslide, and landslide. If the article passes the keyword filter, it is then passed on to AlchemyAPI for text extraction.

AlchemyAPI uses deep machine learning to do natural language processing specifically, semantic text analysis, classification and sentiment analysis. Most importantly we are interested in taxonomy. AlchemyAPI can categorize the articles in classes. For example the article "Ubinas Volcano Causes Massive Landslides in Peru" would be categorized as

"/science/geology/seismology/earthquakes" and "/science/weather/meteorological disaster". We see that it is related to geology and natural disaster so we can be confident that this article is a legitimate landslide article. Once we have determined a particular article is a landslide article, we once again utilize AlchemyAPI to retrieve any city names and number of time the city name was mentioned in the article. We use this data along with Google Maps API to determine where the landslide occurred. The article (source, title, description, url, published date), taxonomy confidence, and cities (name, relevance, count) are saved as output in JSON format which is used by front-end to display landslide occurrences on a timeline map. This entire process repeats at user specified interval.

The following figures show the frontend of our system. Here we take the JSON output before and compile it into a TimeMap interface that utilizes Google Maps API to show landslides as they occur over time. As currently implemented, we only show landslide occurrences in a week period to allow the user to more easily identify landslides from a specified date. Clicking on the pins on the map gives more information containing the location of the landslide, a link to the news article, and a short description of what the article contains. This allows a user to easily confirm whether an article was in fact a landslide and further match the article to the location described in the article.

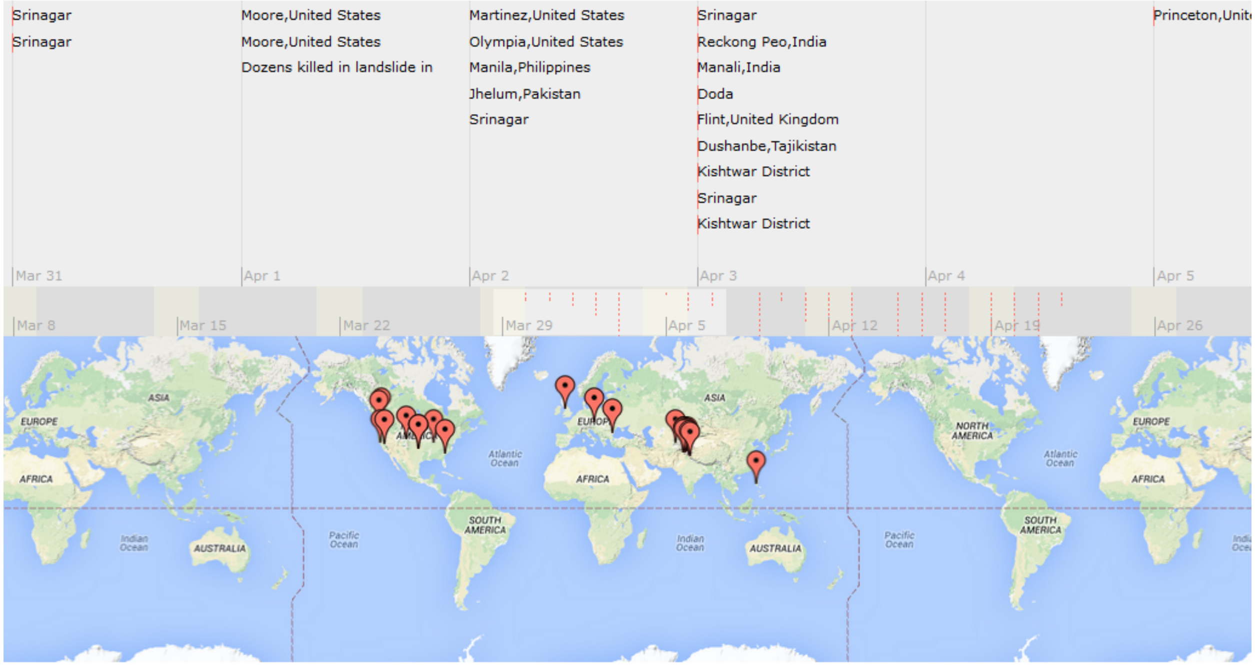
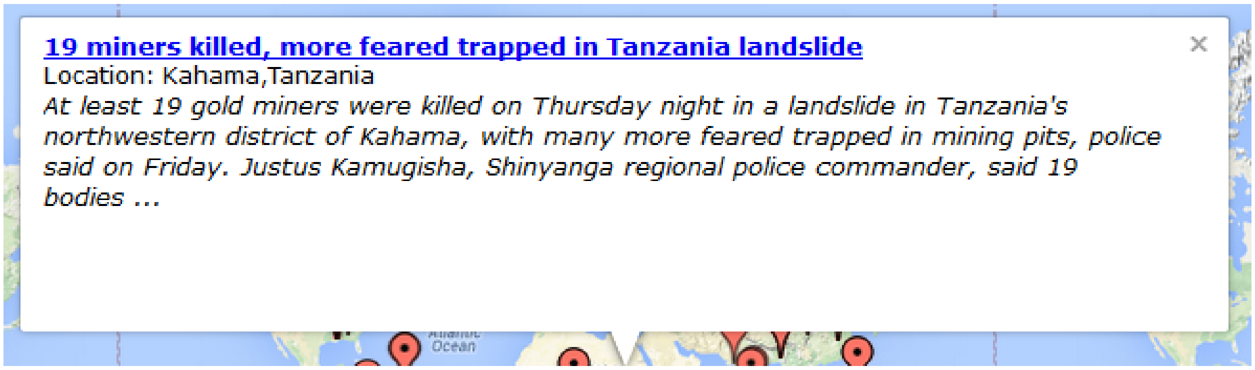


Figure 2. Timemap of Landslides



19 miners killed, more feared trapped in Tanzania landslide
Location: Kahama,Tanzania
At least 19 gold miners were killed on Thursday night in a landslide in Tanzania's northwestern district of Kahama, with many more feared trapped in mining pits, police said on Friday. Justus Kamugisha, Shinyanga regional police commander, said 19 bodies ...

Figure 3. Pin Description of specific landslide

Results

In evaluation, we let our system run for 10 days and appropriately collected results as given below.

In 10 days, the application analyzed 17,044 unique articles. From this, 825 articles passed the initial keyword filtering and were passed onto AlchemyAPI for taxonomy check. From all, only 139 articles passed all checks and included in the output as legitimate landslide articles. The bar graph in Figure 4 emphasizes the level of filtering done to give final output. Here we show that we do not process every candidate news source and look for certain keywords in order to limit amount of items we process over time.

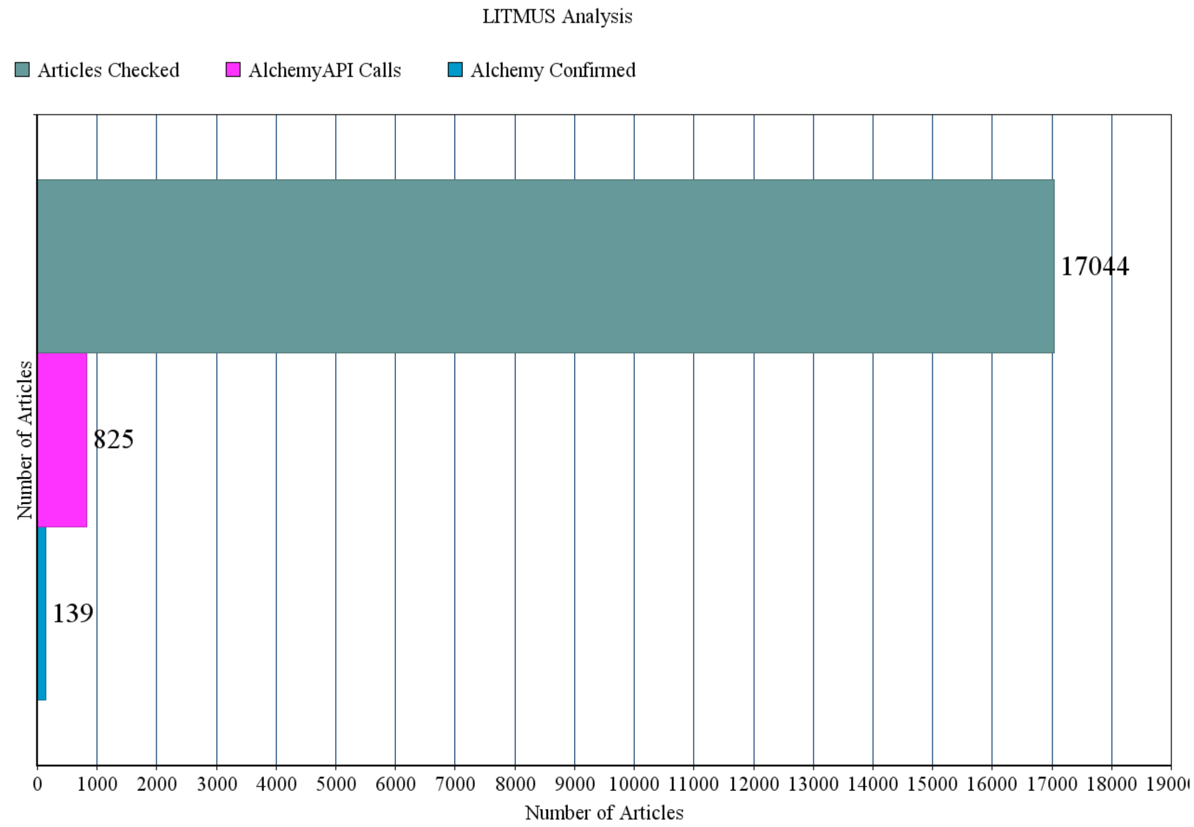


Figure 4. Bar graph depicting number of articles processed

Using strict analysis, where we consider articles such as “Road opens up first time after 1 week of landslide” as a false positive, we see that 73.9% of the articles in the output were legitimate landslide disaster articles, 26.1% were false-positives, 0.4% false-negatives, 99.6% true-negatives giving us overall F-Score of 84.9%. This is visualized in Figure 5 below. To further explain strict analysis, we only considered news articles that were indicative of a landslide that occurred on the published date. Also, we excluded articles that focused scientific discussions of landslides themselves and not necessarily of their occurrence on a particular date.

With lenient analysis, we extended our analysis to include articles in which events were a result of a landslide that had occurred in the recent past. By changing this analysis, we observed that false-positives go down to 8.9%, false-negative to 0.4%, true-positive to 91.1% and true-negative to 99.6% giving us overall F-Score of 95.16%. For rest of the section we will use results of strict analysis.

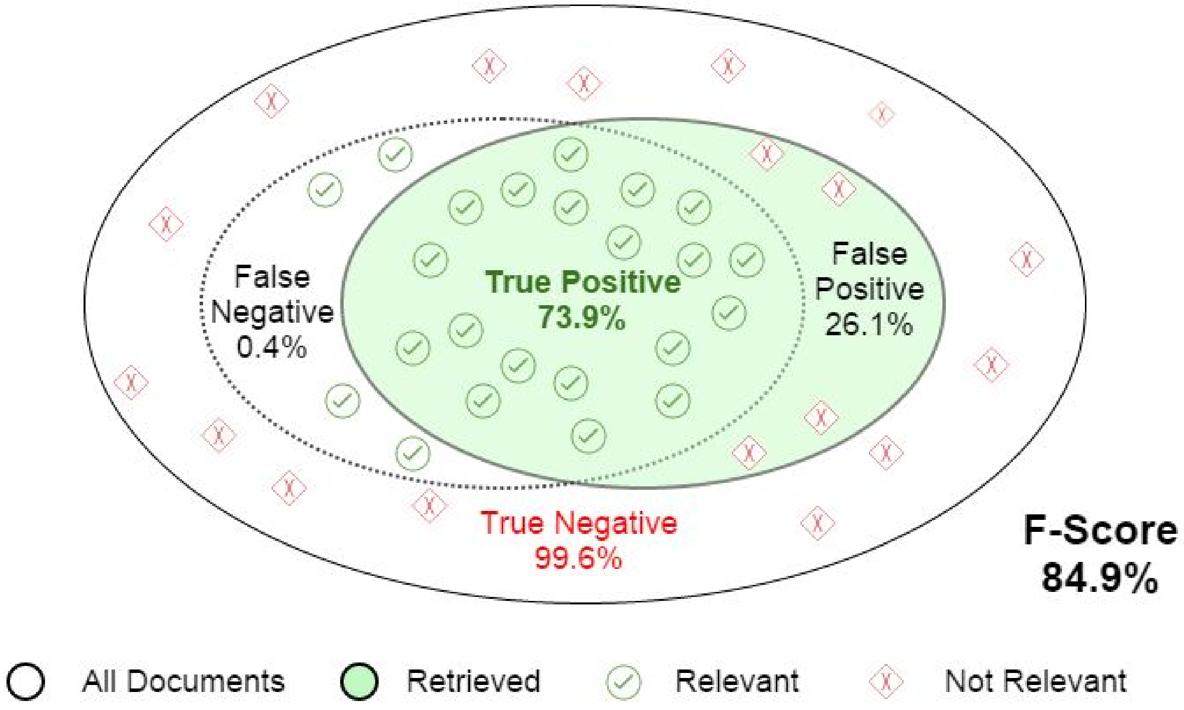
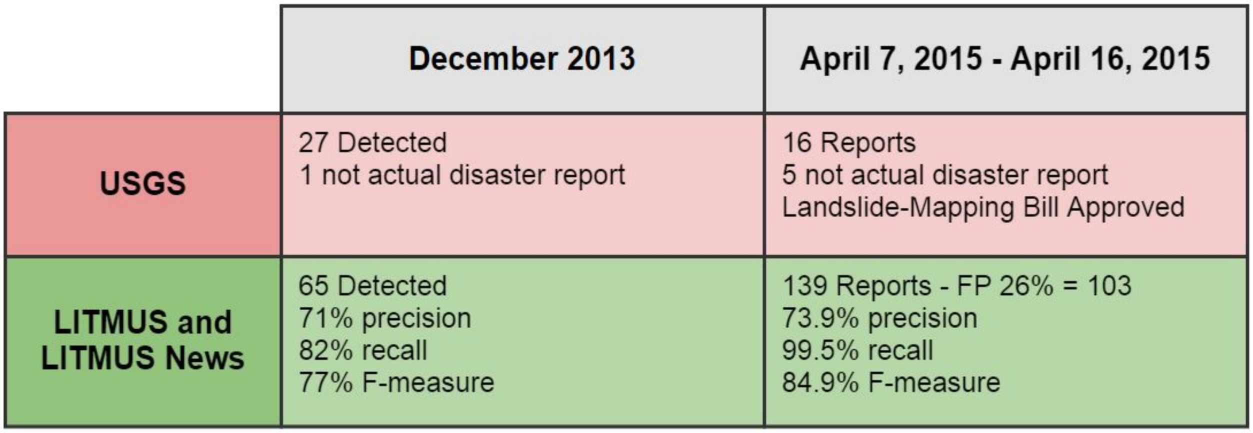


Figure 5. F-Score chart of LITMUS News results

Our application compares well with LITMUS and in the 10 day period where our application was able to discover 139 landslide news articles, USGS published only 16 and 5 of those were not actual disaster reports. Figure 6 provides a direct comparison of our system to LITMUS system findings in December 2013. Also important to note, is that of 139 landslides news articles, they also contained multiple sources of the same landslide occurrence but are also legitimate landslide occurrences.



	December 2013	April 7, 2015 - April 16, 2015
USGS	27 Detected 1 not actual disaster report	16 Reports 5 not actual disaster report Landslide-Mapping Bill Approved
LITMUS and LITMUS News	65 Detected 71% precision 82% recall 77% F-measure	139 Reports - FP 26% = 103 73.9% precision 99.5% recall 84.9% F-measure

Figure 6. LITMUS vs LITMUS News vs USGS comparison

Future Work

We summarize future work in the following points as described below while also considering lessons learned from class.

- Extend work to include detection of a natural phenomenon such as a tornado or an earthquake. The system will place heavier emphasis in targeting those specific areas in news sources.
- Extend project to determine likelihood of landslide in case of natural disaster. The amount of data collected can act as a training set and utilized in conjunction with terrain data to help predict and verify landslide activity.
- With our frontend interface, we can produce a way to track epidemics as they spread throughout the world. Recently, ebola was known to be of such an occurrence and our system could be used to track ebola spread from a news standpoint.
- Multi-threaded processing of news source to increase throughput. Currently, we only process news stories in a sequential order. Adding multi-thread will speed processing time and allow us to include much more new sources into the system. This is important if our system becomes integrated as a smaller component to a large system.
- Add non-English news sources to the system. Because all of the sources that are used are in English format, the majority of articles are pinpointed in the United States. Having more variety of language sources helps increase our landslide coverage more thoroughly.

Lessons Learned

The lessons learned from this project are taken twofold. While trying to decipher the credibility of a landslide report, we were also evaluating the reliability of the Alchemy API and the usage of natural language processing. From this, the team was able to observe the usefulness of such a component and how it is probably better for us to develop a component like this to maintain control over the output. Given the time constraints of the project, we were not able to accomplish to build our own language processing API.

Another lesson that we've taken from this project is figuring out compatibility between system components. In every step of the development process, we considered how the output of one component affected another. For instance, we decided to output our data in JSON format because of its widespread usage today. Also, our intentions was to try to make the application as lightweight as possible. In doing so, we discussed various design features to limit resource usage. An example of this includes tracking of duplicate landslides articles. We debated if we wanted to track this within the program execution itself or via tracking this externally. We decided it was best to do it externally so that the size of the program did not dramatically increase over time.

Conclusion

By adding an authoritative news source as a way to detect landslides, we show that it can help improve landslide detection over the current system which mainly utilizes social media data to confirm landslide occurrence. We must also take into account the usage of the Alchemy API since the system depends on the natural language processing capabilities to produce desired outputs. Given the combination of these two items, it is feasible to improve the current system without creating too much overhead. The system is lightweight enough to be implemented on any platform offering support for Java.

In summary, the three most interesting contributions of the project include the improvements over the LITMUS system, the visual TimeMap frontend, and the usage of Natural Language Processing to categorize text data.

References

[1] Musaev, Aibek, De Wang, and Calton Pu. "LITMUS: Landslide detection by integrating multiple sources." (2014).

[2] Guy, Michelle, et al. "Integration and dissemination of citizen reported and seismically derived earthquake information via social network technologies."*Advances in intelligent data analysis IX*. Springer Berlin Heidelberg, 2010. 42-53.

[3] Sakaki, Takeshi, Makoto Okazaki, and Yutaka Matsuo. "Earthquake shakes Twitter users: real-time event detection by social sensors." *Proceedings of the 19th international conference on World wide web*. ACM, 2010.

[4] Ramesh, Maneesha V. "Real-time wireless sensor network for landslide detection." *Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on*. IEEE, 2009.

[5] Mishra, P. K., Shukla, S. K., Dutta, S., Chaulya, S. K., & Prasad, G. M. (2011). Detection of Landslide Using Wireless Sensor Networks. *IEEE*.

[6] Alchemy API. http://www.alchemyapi.com/products/alchemylanguage.