The Eric & Wendy Schmidt
Data Science for Social Good
Summer Fellowship 2013

Twitter-Informed Disaster Relief

Zahra Ashktorab* (UMD), Christopher Brown* (UT), Manojit Nandi* (CMU), Aron Culotta (IIT)

*These authors contributed equally.

THE UNIVERSITY OF CHICAGO

QCRI معهد قطر لبحوث الحوسبة Qatar Computing Research Institute

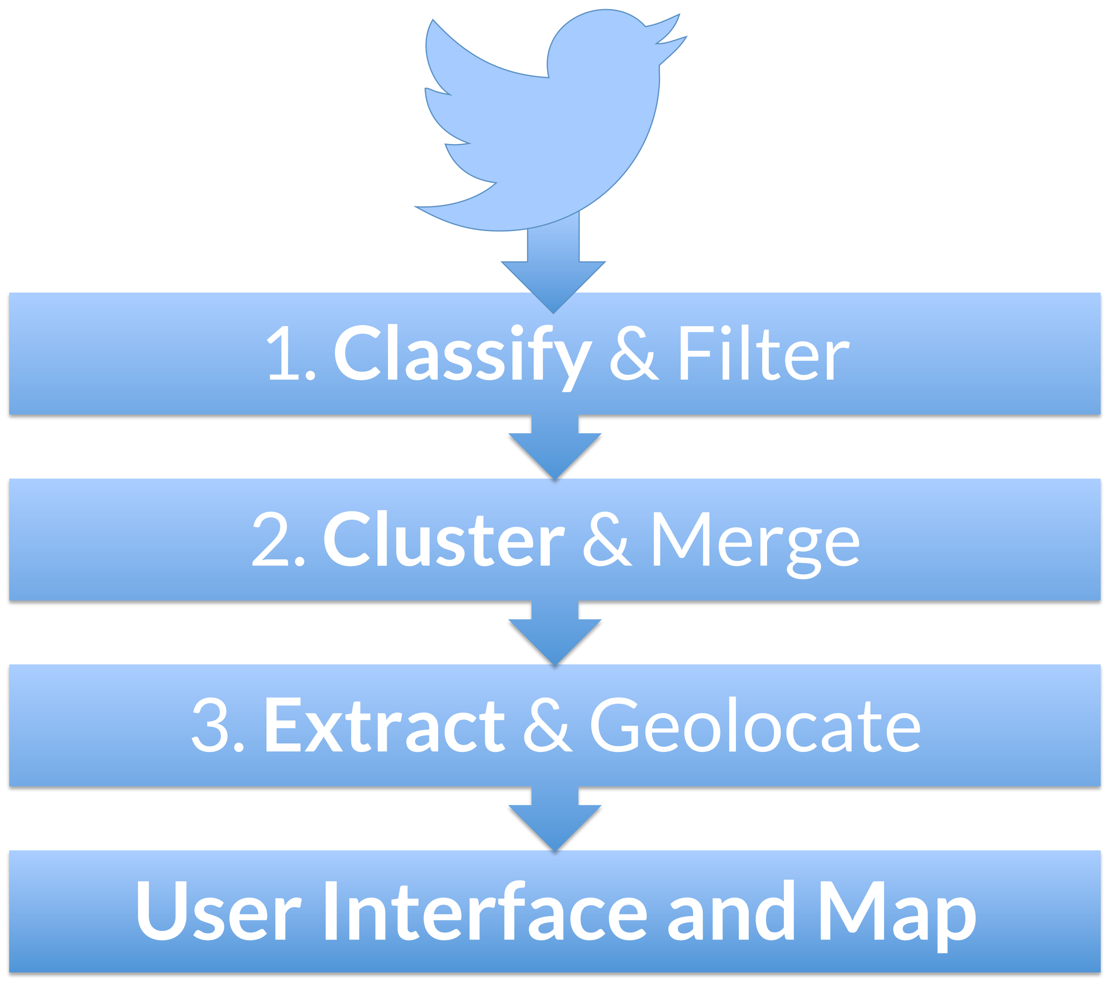
Overview

Tweedr filters social media, identifying and extracting information that can help direct disaster relief efforts. Using various machine learning techniques, we can aggregate and triage status updates, making the sparse information on Twitter more accessible to disaster relief workers.

The Problem

Too little information, too many tweets

- Disasters cause damage and create chaos precisely when disaster relief workers need the **most accurate** and **up-to-date** information.
- **Twitter** is a critical channel of communication **during disasters**, as landlines and electricity can be temporarily disabled or destroyed.
- Twitter is used for many other things; **the signal-to-noise ratio is very low**. Relief workers may have to sift through thousands of tweets before they find anything that can help them expedite or direct their efforts.



The Solution

Tweedr, an application (API and UI) to filter, aggregate, and consume actionable information from tweets

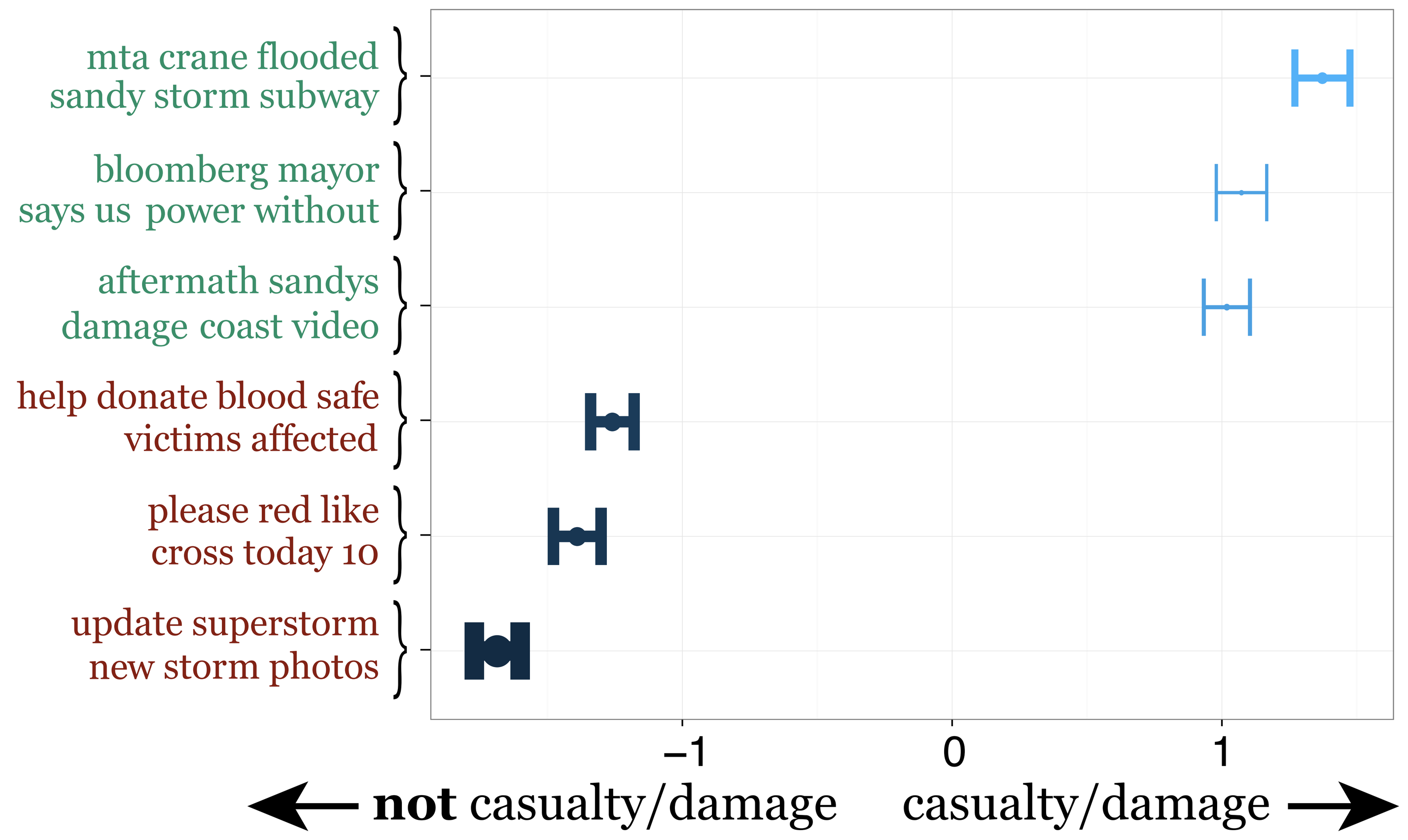
- We can do much of this **filtering** and **aggregation** using machine learning algorithms to train models on **past disasters** and apply these to new tweets from **new disasters**.
- **Tweedr** brings together approaches from several established open source machine learning libraries to sort through the **deluge of text** and deliver information that **disaster relief workers can act on immediately**.

Data

- Our data consists of **500 million tweets** from twelve disasters in English speaking countries, including **Hurricane** Sandy, the Joplin tornado, Hurricane Ike, the Christchurch (NZ) **earthquake**, Hurricane Irene, Moore (OK) tornadoes, Super Tuesday **tornados**, and the Samoa earthquake.

Tweedr identifies and extracts useful information from social media during disasters: tweedr.dssg.io

sLDA: Confidence in casualty/damage classification task, top six indicators



1. Classification

Supervised topic modeling

- We train a topic model, using supervised **Latent Dirichlet Allocation**, on labeled tweets from a wide sample of past disasters, learning tokens and collocations that indicate different types of tweets.
- Each incoming tweet is classified using that model, and if any **relevant** category exceeds a threshold of significance, we assign it to one or more **categories** and send it to the next step in the pipeline.

RT @DaphneUn: Awesome. Go NYC. RT @pourmecoffee: Empire State Building shines in the dark like a boss. http://t.co/HLuLBW05 #sandy	Wider shot of scaffolding toppling car on CPW and 92nd across from Central ParkNYC @nowthisnews #sandy http://t.co/ivkExinW
sLDA → Building: 0.97 **Damage: 0.07** Injury: 0.01 ✗ toss it	sLDA → Building: 0.34 **Damage: 0.29** Injury: 0.04 ✓ keep it

2. Clustering

Bloom filters

- Bloom filters can identify duplicate messages efficiently, helping to cluster **retweets**.

Similarity hashing

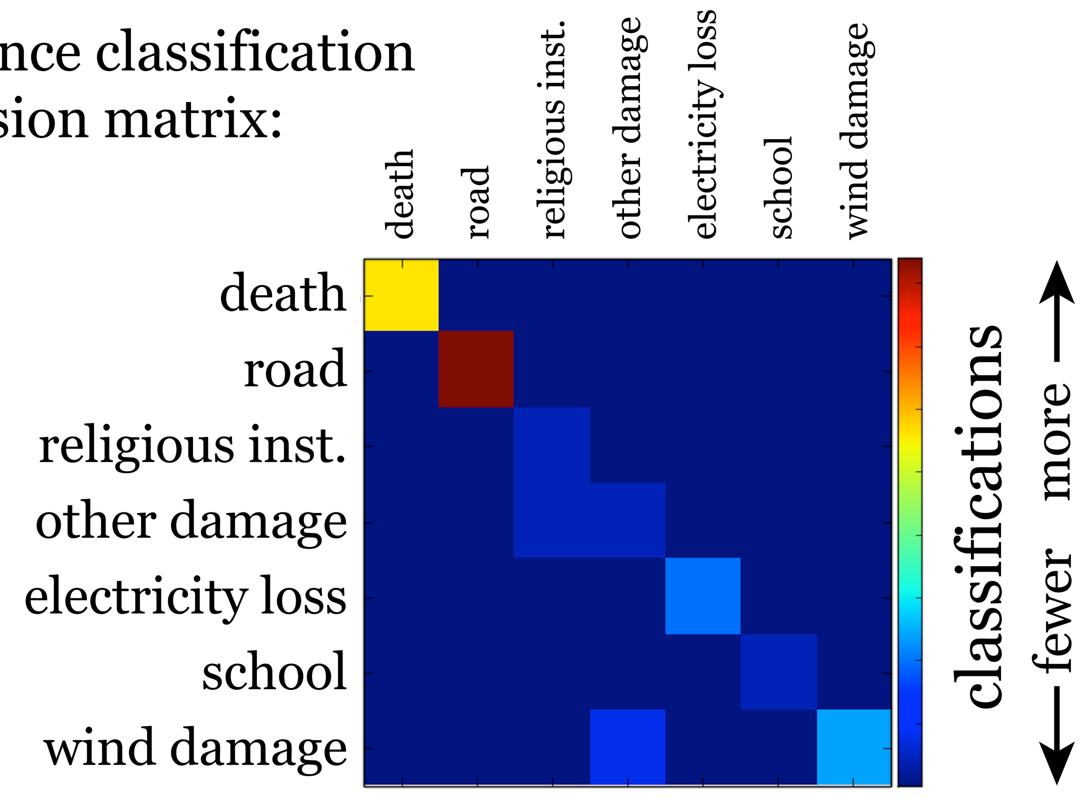
- On a smaller scale, we can use **approximate similarity** matching to compare new tweets to a smaller window of potential matches, such as all tweets produced in the last 24 hours, or tweets with exact duplicates.

3. Information Extraction

Conditional Random Fields

- We train a CRF on sequence-annotated tweets with labels, such as "road", "flood", " health service" and about twenty others types of interesting text.
- Tagging incoming tweets enables aggregation and filtering at a **higher level than token searches**.

Sequence classification confusion matrix:



The semantic web

- DBpedia is a structured representation of 3.7 million entities from the Wikipedia project.
- Using AWS to host an instance of DBpedia Spotlight, we identify known entities in incoming tweets, and use geolocation data to map mentions.

Goal Product

- **API** to handle streams of incoming tweets, apply knowledge gathered from past disasters, and **enhance raw tweets** with additional information.
- **UI** to consume streams from the **API**, allowing relief workers from the **Red Cross** or **FEMA** to query the data for information that they need.

References

- Moses S. Charikar. Similarity estimation techniques from rounding algorithms. *STOC* '02, 2002.
- Naoaki Okazaki. CRFsuite: a fast implementation of conditional random fields (CRFs), 2007.
- David Blei and Jon McAuliffe. Supervised topic models. *NIPS*, 2007.

This work was done during the Eric & Wendy Schmidt Data Science for Social Good Fellowship at the University of Chicago.