Social Media and Crowdsourcing Assessment of Weather Impacts: The Dashboard

Who are we?

Project Lead: Renée Sieber
Technical Lead: Andrei Romascanu
Unsupervised classification Co Lead: Rosie Zhao
Unsupervised classification Co Lead: Mikael Brunila
Dashboard Lead: Lucia Berger
SNA Lead: Hannah Ker
Phase 1 Lead: Drew Bush
Phase 2 Lead: Sam Lumley

How do we make sense, in real time, of social media coming from the public about extreme weather events?

During extreme weather, social media data can be extremely useful.

Social media has the advantages and disadvantages of big data.

Natural Language Processing (NLP), with supervised and unsupervised classification can hopefully make sense of social media and make us more resilient in the face of storms.
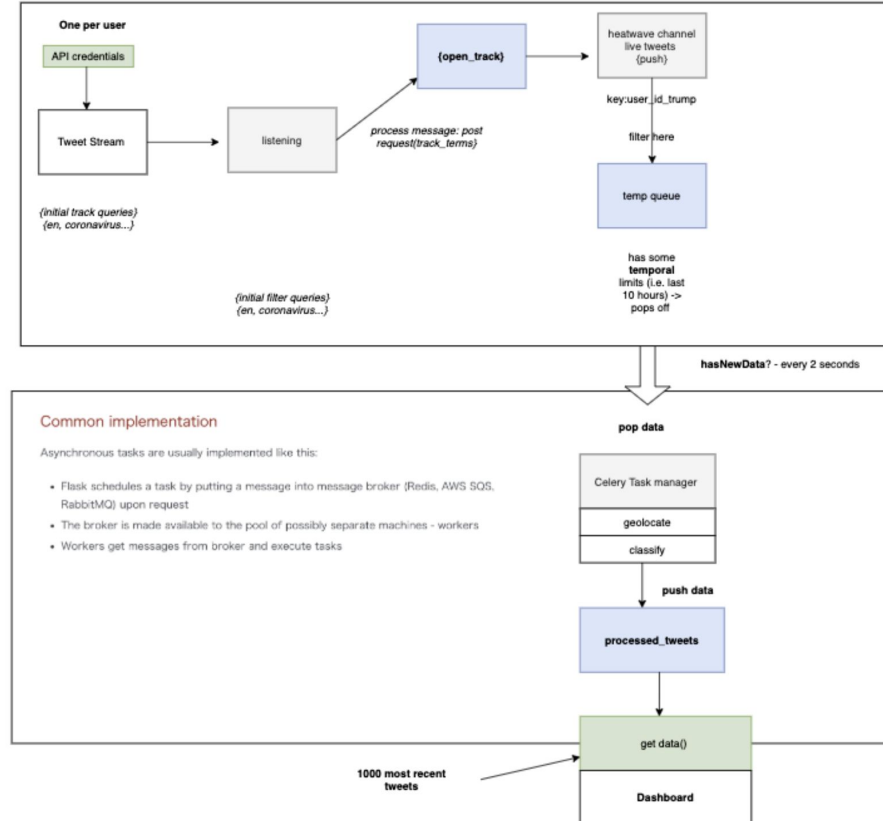
This session is about front end user interface

Dashboard Key Functionality

- **USER**:
  - User adds their twitter API keys on registration
  - Authentication
- **FILTER**: stream on **live key term**
  - **Live twitter data is**
    - Geolocated
    - Classified
    - Frequently occurring terms can be analyzed
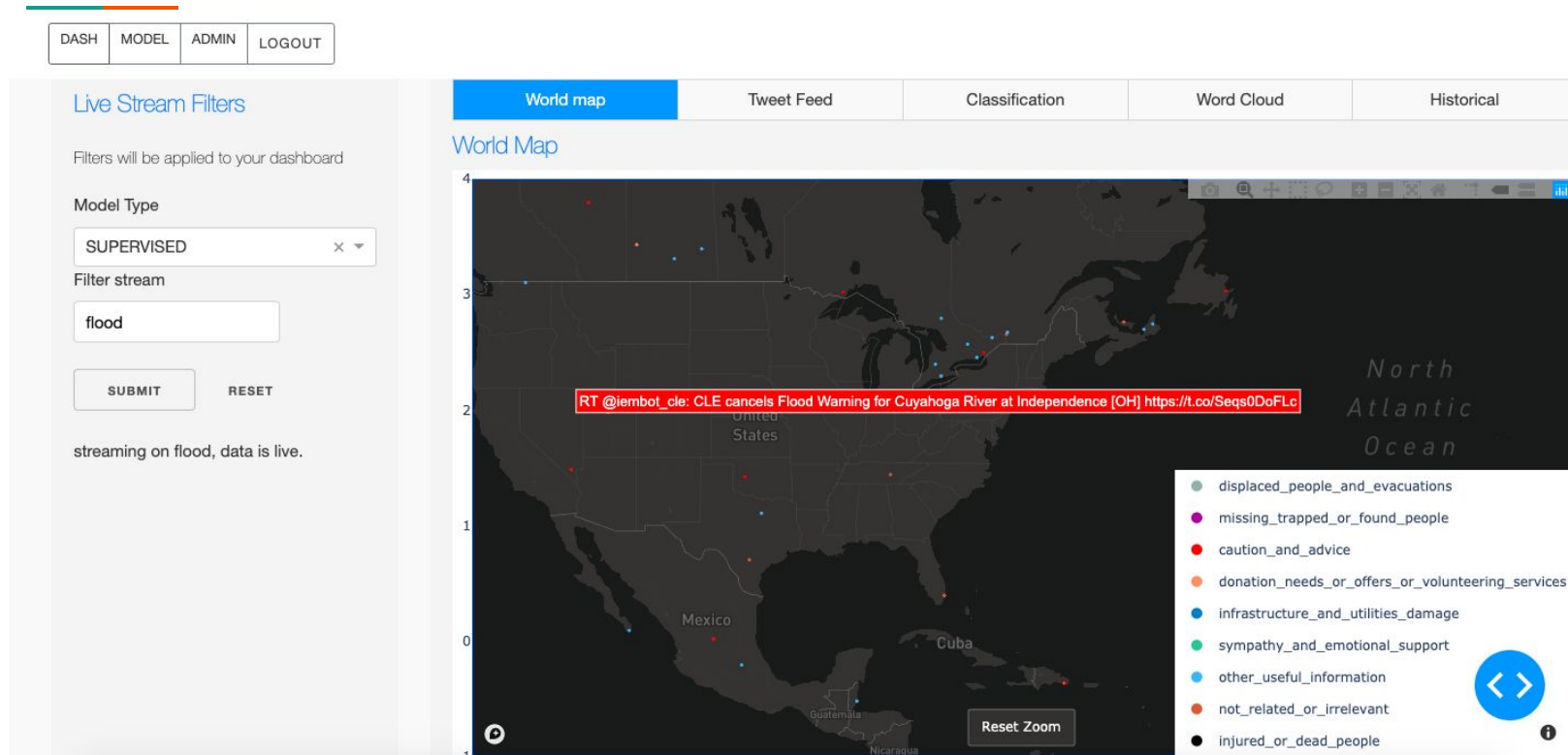 - Search historical data (7 days)

Dashboard diagram

Twitter Streaming API:

Ingest with Channels Lists/Sorted Sets

One per user

API credentials

Tweet Stream

listening

process message: post request{track_terms}

{open_track}

heatwave channel live tweets {push}

key:user_id_trump

filter here

temp queue

has some **temporal** limits (i.e. last 10 hours) -> pops off

{initial track queries} {en, coronavirus...}

{initial filter queries} {en, coronavirus...}

hasNewData? - every 2 seconds

Common implementation

Asynchronous tasks are usually implemented like this:

- Flask schedules a task by putting a message into message broker (Redis, AWS SQS, RabbitMQ) upon request
- The broker is made available to the pool of possibly separate machines - workers
- Workers get messages from broker and execute tasks

pop data

Celery Task manager

geolocate

classify

push data

processed_tweets

get data()

1000 most recent tweets

Dashboard

User filters stream on term *"flood"*

User filter 'flood' on "warning"

Incoming tweets are classified

"Flood" query returns 31 instances of *caution and advice*

A word cloud is generated
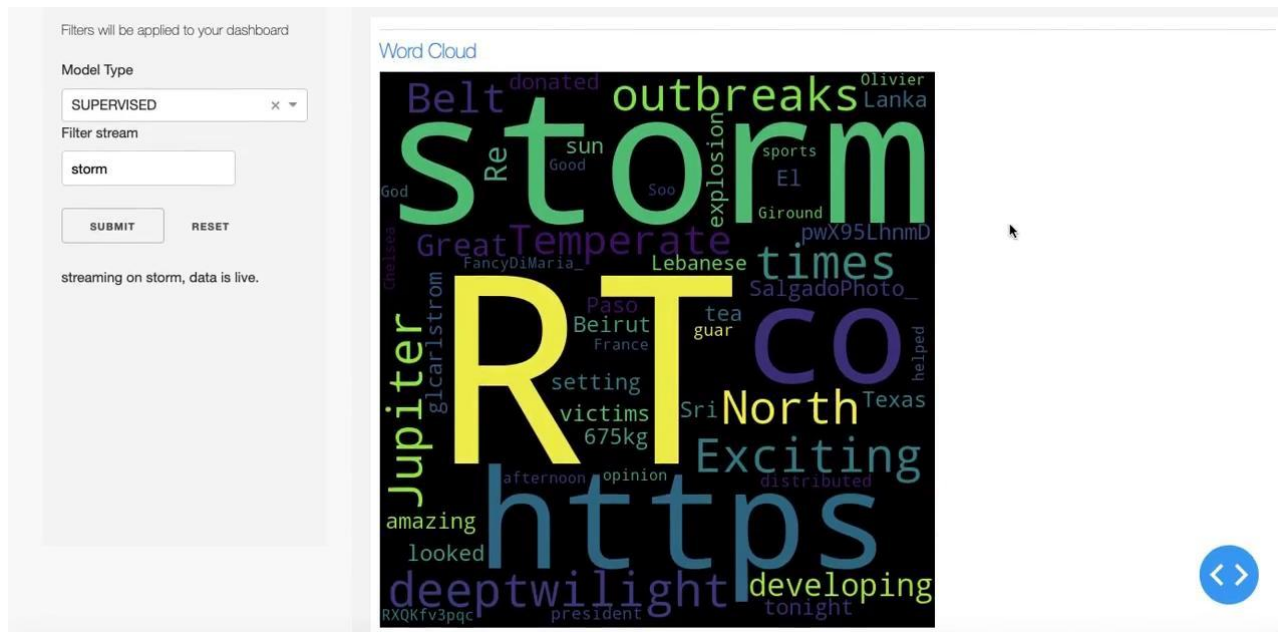

"Flood" query returns *prepared, warning, flood*
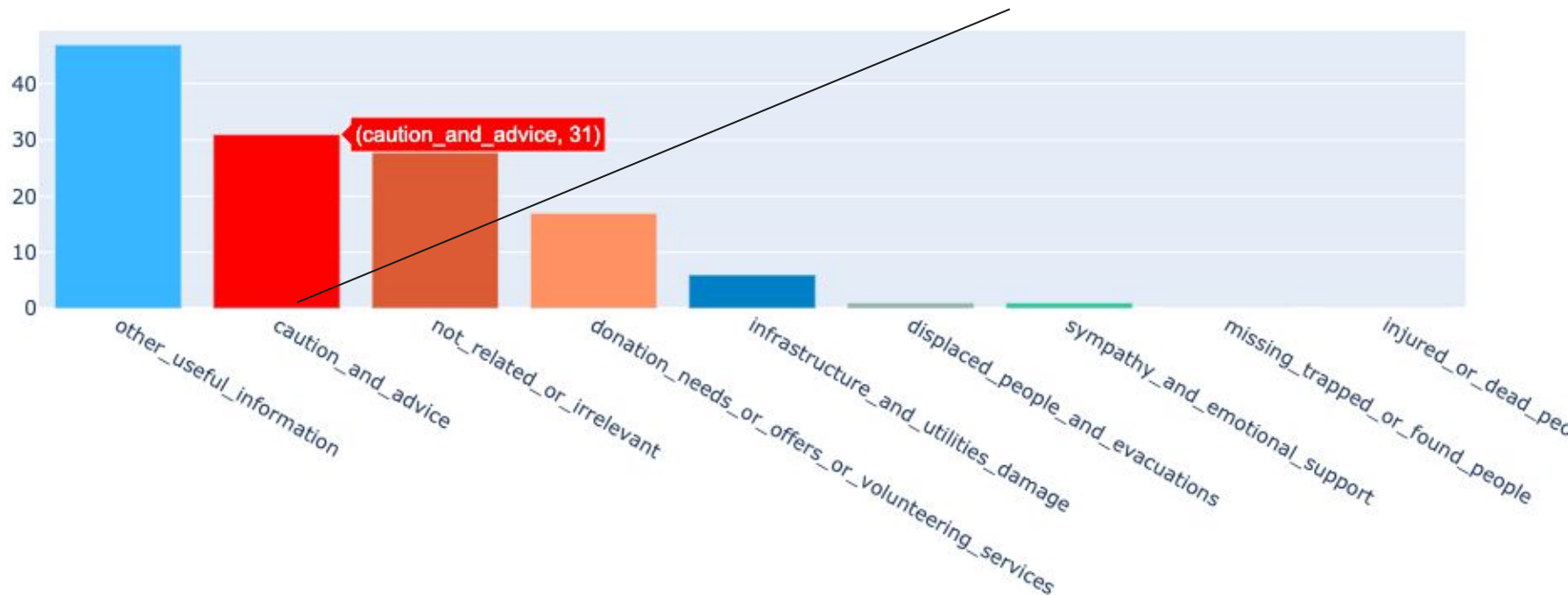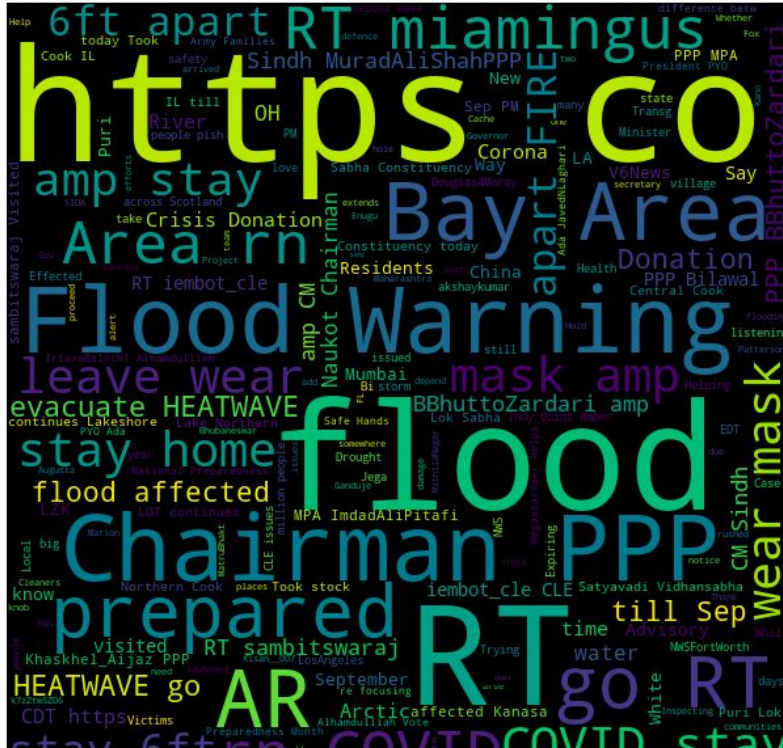
- Visualize most frequent terms
- These are dynamic and change as the data updates
- Changes with current events sublinks

A local/McGill server demo

- User logs in
- Filters on the word *storm*
- Analyze the classes and locations of incoming tweets
- On McGill network (http://132.206.251.68/)

Demo Video

What Skills/Positions You will need

GUI
yes

For your supervisor

For the end user (of the front end)

For the person doing the filtering*

Have skilled demonstrator who knows features

Have documentation describing features

Understand limits of filters, streams

GUI
no

For developer, who may wish to modify functions

For the data scientist, working on the classifications

For system administrator, who maintains h/s

Install via command line
Add to/modify to existing functionality (eg new viz)
Know plot.ly or comparable dashboard engine
Know Python, comparable coding language

Retrain model with new datasets/different configurations
Use new topic modelling, classification model

Monitor servers, update patches

*probably also needs skills to install

What hardware/software/data you will need

1. **<u>Hardware</u>**:
   a. Access to a web-server
   b. OR access to a cloud server or virtual machine

2. **<u>Software/Containers:</u>**
   a. Docker container
   b. Instructions for configuration are in the git repo
   c. Built in python
   d. Access to the Twitter API

3. **<u>Data</u>**
   a. Twitter data
   b. Training data
   c. Classification scheme
   d. Filters

Updates on Model Building

	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic_8
0	emergency	eminem	safe	cbcnl	people	like	nltraffic	closed	cm
1	state	photo	stay	today	today	storm	road	st	st
2	st	click	blizzard	thank	storm	snow	power	tomorrow	winds
3	city	learn	newfoundland	people	need	time	st	john	pm
4	cityofstjohns	saveng	storm	day	like	day	street	remain	today
5	says	michelleobama	canada	home	day	newfoundland	drive	january	km
6	declared	sexeducation	weather	work	grocery	going	pearl	update	airport
7	john	ken	nlstorm	help	food	blizzard	roads	emergency	yyt
8	mayor	starr	warm	storm	open	house	line	state	snowfall
9	roads	pin	snowstorm	nltraffic	know	today	mount	today	blizzard

Usefulness? Interpretability?

Live Stream Filters

Filters will be applied to your dashboard

Model Type

UNSUPERVISED   ✕ ▲

UNSUPERVISED

SUPERVISED

SUBMIT   RESET

streaming on arctic

Reminder: Supervised Classification

Supervised classification organizes content into predetermined categories. It can categorize in real time.

Before it's used, supervised classification requires large training datasets.

Our model was trained with a dataset of 16,000 tweets from crises (floods, hurricanes, earthquakes) labeled by **CrisisNLP** (Imran et al., 2016).

Reminder: Unsupervised classification

Unsupervised classification--we use topic modelling--categorizes a corpus when the labels aren't known (e.g., #nlwx).

Top relevant terms assigned to a cluster can be interpreted and an **emergent topic** can be labelled.

Preparation is **less labour-intensive** (i.e., no prior training of data required).