Tracking recent events through recent Wikipedia changes using Storm

by Gustaf Helgesson
Aim

● Correlate # of article changes *within* a language to recent events.
  ○ For English, German, Spanish and Japanese.

● Correlate article changes *between* languages to recent events.
  ○ By using Wikipedia’s “in another language: English” feature.
Data collection

- #Recent changes per article per language
  - For: English, Spanish, German and Japanese

- Use streaming windows of 2-6 hours and see how event changes for the top 100 events

- Depending on necessity I may make use of approximate counting in the counting phases.
Input stream - JSON data!
Storm Intro/Recap

- Stream Processing Engine
- Programmers create explicit DAGs (topologies) of custom or built in functions
- External inputs (spouts), external outputs (sinks), processing elements (bolts)
Storm topology
Storm topology

Spouts

- English
- Deutsch
- Español

Bolts

- (Approximate) counter #1
- (Approximate) counter #n
Storm topology

Spouts

- English
- Deutsch
- Español
- 日本語

Bolts

- (Approximate) counter #1
- (Approximate) counter #n

Local ranker #1
- 
- 
- 

Local ranker #4
Storm topology

Spouts
- English
- Deutsch
- Español
- 日本語

Bolts
- (Approximate) counter #1
- ...
- ...
- (Approximate) counter #n

Local ranker #1
- ...
- ...

Local ranker #4

Global trender
Storm topology

Spouts:
- English
- Deutsch
- Español
- 日本語

Bolts:
- (Approximate) counter #1
- ...
- (Approximate) counter #n

Local rankers:
- Local ranker #1
- ...
- Local ranker #4

Sink:
- MySQL
- Global trender
Storm topology

Spouts
- English
- Deutsch
- Español
- 日本語

Bolts
- (Approximate) counter #1

Local ranker #1
- .
- .
- .

Local ranker #4

Global trender

MySQL

Sink

Apache/flask
Expected Results

● Recent news locally and globally between the languages visible in trending topics and related people
  ○ E.g. Sotji medal count, Canada hockey team, Sidney Crosby.

● To a smaller degree article propagation
  ○ Minor changes in an English article being picked up and added to other languages.
Potential pitfalls

● Missed events
  ○ One person making a single, large change to a topic
  ○ May be solvable by comparing against similar pages which should hopefully be edited too!

● Potential noise
  ○ Spammers may trigger many changes and community undos will add to the number of changes!
Deployment

- Rent 4-5 Amazon EC2 instances for a two day period

- m3.large instances
  - Dual core Intel Xeon E5-2680 @2.6GHz, 32GB SSD 7.5GB RAM

- Use the Storm-deploy tool to deploy the Storm program over a
Current Progress

- Design plan
- Got the sample Storm program and a development environment locally
- Set up an EC2 account
- Able to scrape recent changes from Wikipedia in JSON format
Plan

● Create a Storm program with the proposed topology

● Setup a simple web interface to easily observe recent trends between languages

● Deploy the program on EC2

● Try to see how different topologies can make the program more efficient

● Look into page view counts as opposed to edits and see if these correspond better with recent events
Questions / Suggestions?