Feature-rich event detection in social media streams

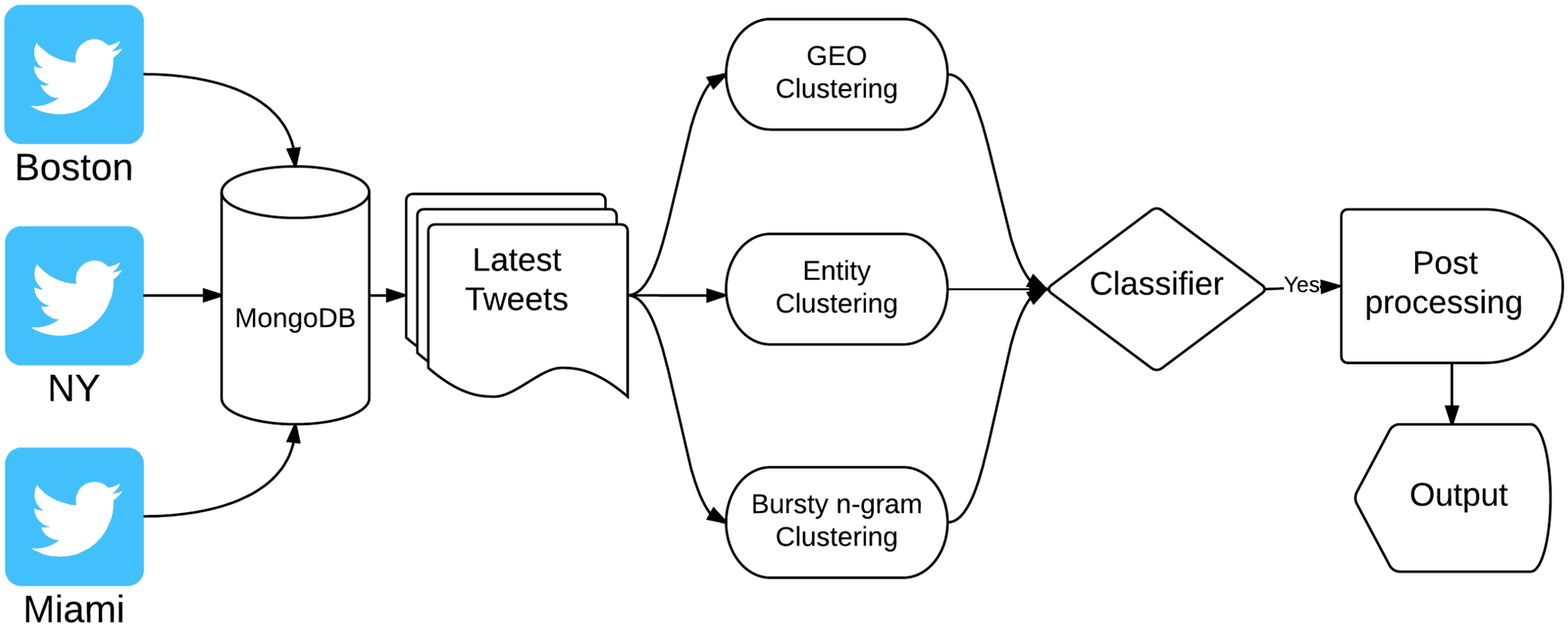
by Denis Antyukhov

Motivation

- Social networking platforms such as Twitter have emerged in recent years, creating a radically new mode of communication between people.
- Monitoring and analysing rich and continuous flow of user-generated content can yield unprecedentedly valuable information, which would not have been available from traditional media outlets.
- However, learning from microblog streams poses new challenges, due to the noisy nature of user-generated content

- **Goal: develop a system for detecting new and ongoing events from streams of Twitter messages**

System Architecture



Data

- **Sources**: Boston, New York, Miami, Chicago and Vancouver tweets

- **Mining**: self-developed tool built against Twitter public API

- **Storage**: MongoDB (supports temporal and spatial indexing)

- **Total**: 8 million tweets in json format, 30 GB size

Language processing

Tweets are tokenised using regular expressions, stop words are removed, hyperlinks, hashtags, user mentions, check-ins and emoji are extracted.

To facilitate vectorising and classification operations, two models were learned using our tweet corpus:

- **TF-IDF:** 75 000 - vocabulary weighting statistic
- **word2vec:** 512-dimensional word embeddings

Clustering: GEO

- 10% of tweets have precise geo position
- Very useful for event detection: people tweeting about a certain event are often close to each other
- **DBSCAN** algorithm is used to cluster data points in 2-dimensional space
- This model is really good at detecting **sports** and **music** events, and other major happenings

Clustering: entity

- **~50%** tweets contain **#hashtags** and @ **check-ins**, we refer to these objects as entities

- Entities are useful because people tend to use same hashtags when relating to a certain event, and check-ins correspond to a location in space. They are also prone to misspelling.

- We extract all entities found within a time window, vectorise, and apply hierarchical clustering based on cosine similarity

- This model efficiently detects popular events described by hashtags, or happening at a certain venue

Clustering: bursty n-gram

- This approach is based on detecting n-grams that demonstrate bursty behaviour (i.e. start appearing unusually often compared to historic data)

- We start by extracting 2,3-grams that are frequently appearing within a time window, vectorise documents that contain them and apply hierarchical clustering.

- We rank the clusters using formula (Aiello et al. 2013)

$$df - idf_t = \frac{df_i + 1}{\log\left(\frac{\sum_{j=i}^{t} df_{i-j}}{t} + 1\right) + 1}$$

Classification

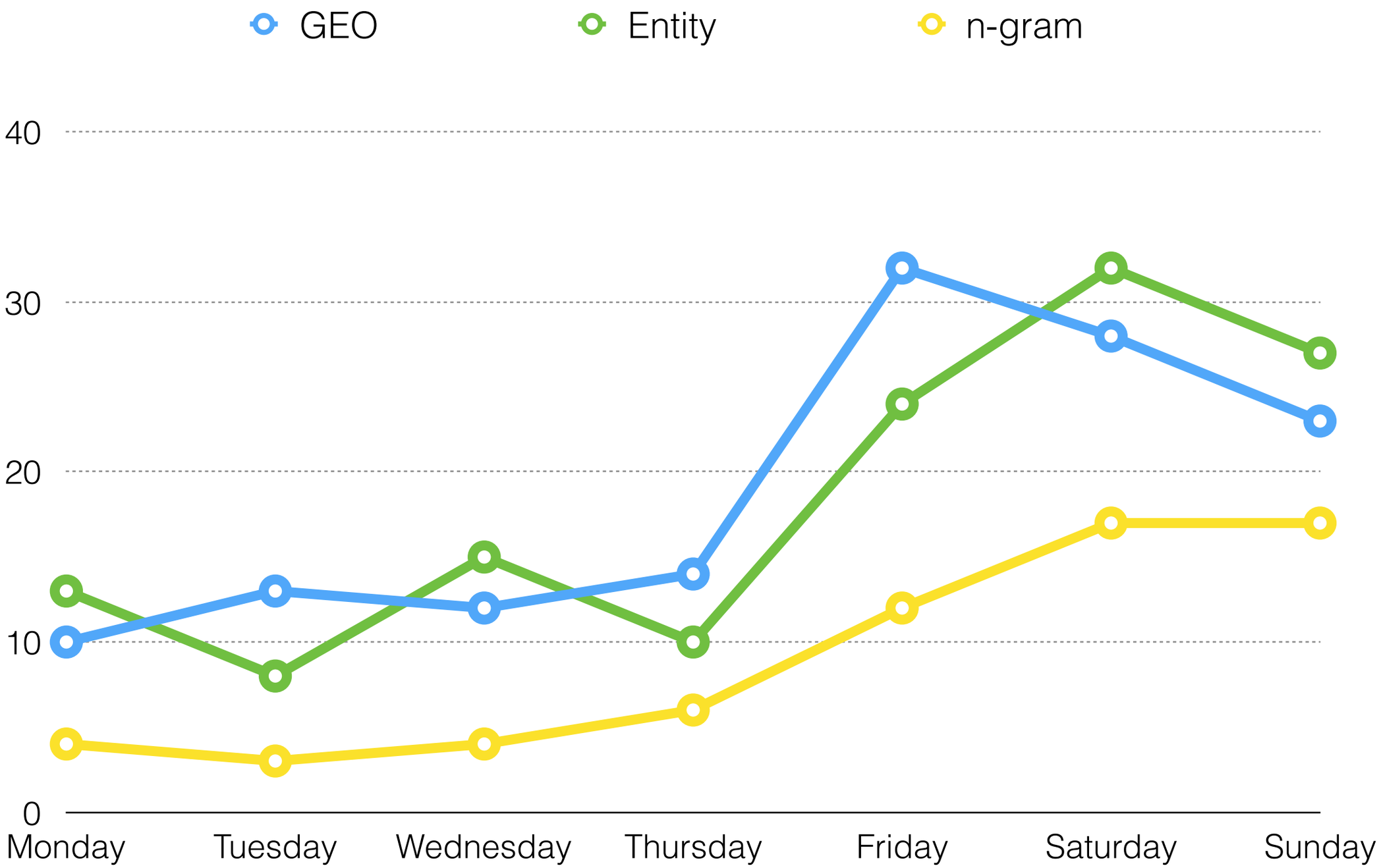
- Our models detect event-related clusters, as well as heterogeneous collections, rumours and spam.
  To classify clusters we extract 20 rich features:

- **Textual:** number of unique unigrams, proper nouns, hashtags, mean tf-idf/word2vec similarity …
  Social: number of friends, followers, retweets
  Meta: hyperlinks, instagram links, entities, etc

- We use a perceptron model to classify clusters based on this feature representation (2 hidden layers)

	precision	recall	f1 score
event	0.69	0.79	0.74
spam	0.73	0.72	0.72

Events of NY


GEO
Entity
n-gram
40
30
20
10
0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

Sample output

#SantaCon #fun #santaconnewport #santacon @ Newport, Rhode Island




#concert #The1975 #NYC #terminal5 @ TERMINAL 5




#jets #giants #NYGiants @ New York Giants vs. New York Jets