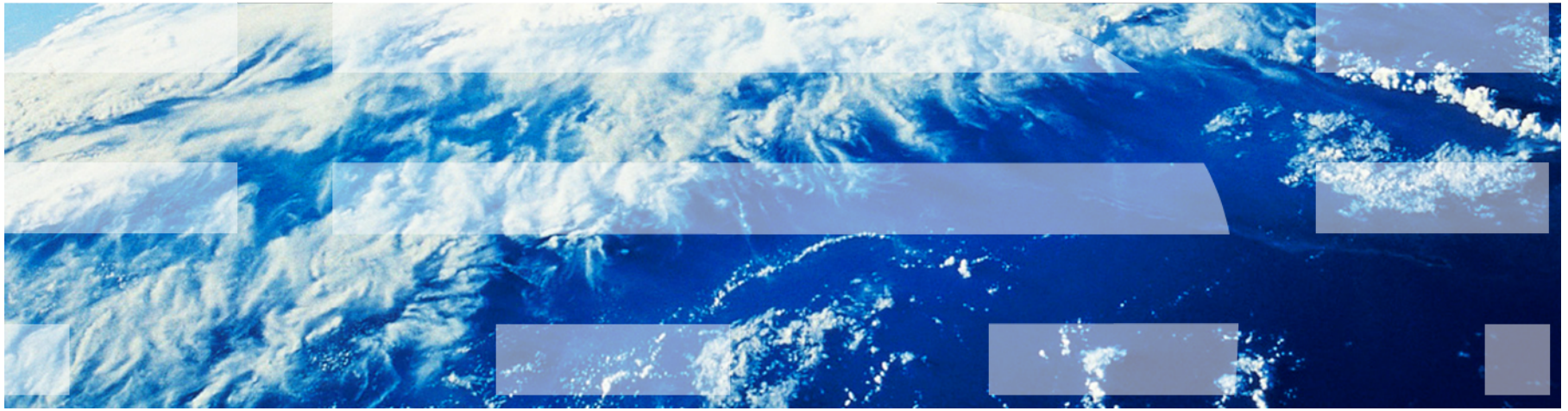
E6893 Big Data Analytics Lecture 7:

Streaming *Big Data Analytics*

Ching-Yung Lin, Ph.D.

Adjunct Professor, Dept. of Electrical Engineering and Computer Science



October 18th, 2019

Stream Analyses on Social Media Data

Social Media Stream Monitoring

Goal 1: Detect, classify, measure and track the
(a) formation, development, and spread of ideas & concepts (memes)
(b) purposeful or deceptive messaging and misinformation

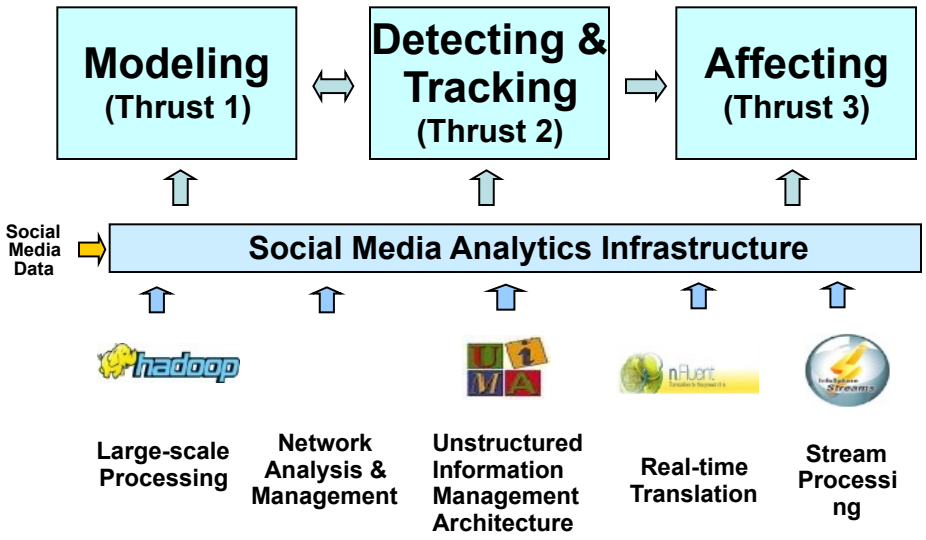
Goal 2: Recognize persuasion campaign structures and influence operations across social media sites and communities

Goal 3: identify participants and intent, and measure effects of persuasion campaigns

Goal 4: Counter messaging of detected adversary influence operations

53+ papers published, accepted, & submitted
12+ patents filed
ACM CIKM 2012 Best Paper Award
IEEE BigData 2013 Best Paper Award
PNAS Cover Article Jan 2013
Science (1)
Nature (2)

Approach: Modeling, Tracking and Affecting Information Dissemination in Context



Thrust 1. Modeling Information Dissemination in Context:
Models of Trust and Social Capital, Information Morphing, Persuasiveness and Competition of Memes, Dynamics of Social Influence

Thrust 2. Detecting and Tracking Information Dissemination in Context:
Detecting Malicious Info Propagation, Evolution History and Authenticity of Multimedia Memes,

Thrust 3. Affecting Information Dissemination in Context:
Automated Generation of Counter Messaging, Influencing the Outcome of Competing Memes and Counter Messaging

Social Media Solution

Live Monitoring

Monitoring real-time tweets on keyword:

Monitor live tweets »

Trend Monitoring

Analyzing trend of conversations based on hashtags

View trends »

Multimedia Monitoring

Recognizing visual content and analyzing visual sentiments

View multimedia »

Geo Monitoring

Monitoring the places that people are sending out tweets

View places »

Scope Identification

Define user-specified sets of keywords for monitoring and analytics

Define scopes »

Concept Analytics

Analyzing statistics of groups based on time, topics, etc

Concept searches »

Link Exploration

Visualizing relationships, discussion sequences and graphs

View relationships »

Impact Prediction

Analyzing conversations and predicting their impact to business

View impacts »

Story Detection

Detecting live developing stories on social media and their evolution

View stories »

Person Analytics

Analyzing a person's personality, trustworthiness, etc.

View person »

Target Discovery

Inspecting potential users for bot detection, marketing, or influencing

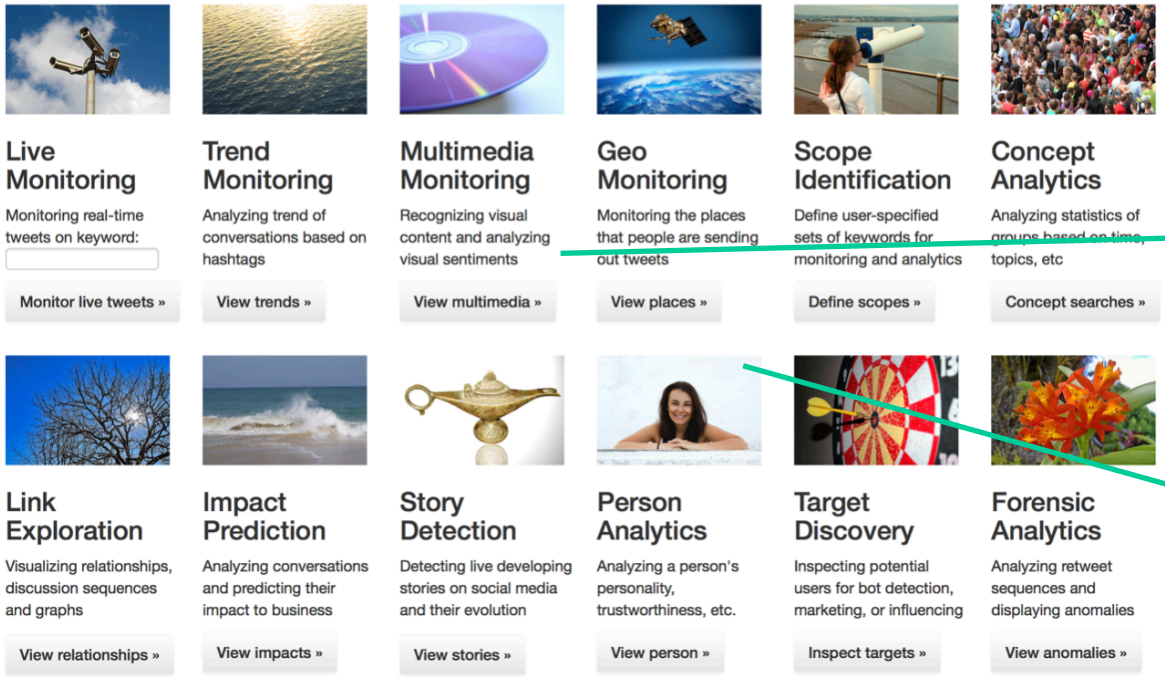
Inspect targets »

Forensic Analytics

Analyzing retweet sequences and displaying anomalies

View anomalies »

Social Media Stream Monitoring Applications



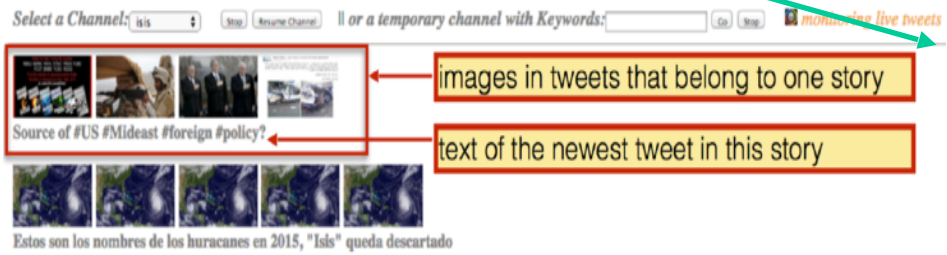
Sun Jan 25 02:00:04 +0000 2015 GMT

RT @IraqLiveUpdate: Pics - Large convoy of KH heading to undisclosed location (3 of 3) #Iraq http://t.co/i47ZjPkmsC

Retweet Count: *3*
Follower Count: *207*
Tweet Sender Location: *N/A*

Automatic Tagging: *deadly_attack, horizontal_text, excellent_book, annoying_reflection, busy_bridge*

Visual Sentiment Detection: *Negative*



Bank Use Case

- **Objective:** Detect unexpected social media movements that may impact a major bank's business

- **X-Bank:**
  - Major bank in Spain

- **Client needs:**
  - Monitor Catalan independence movement: independence may bring bankruptcy since X-Bank needs ECB support
  - Detect potential PR crisis by analyzing the formation and spreading of grassroots opinion on their employees and services

- **Challenges:**
  - Existing social media monitoring tools miss important tweets that don't contain specified keywords and are not from specified users
  - Existing tools lack of predictive capability of tweets' potential influence

An image tweet (without mentioning "*the bank name*") sparks a lot critiques of their unfair practice

City Government Use Case

- **Objective:** Detect unexpected social media discussions that may impact a government's campaign

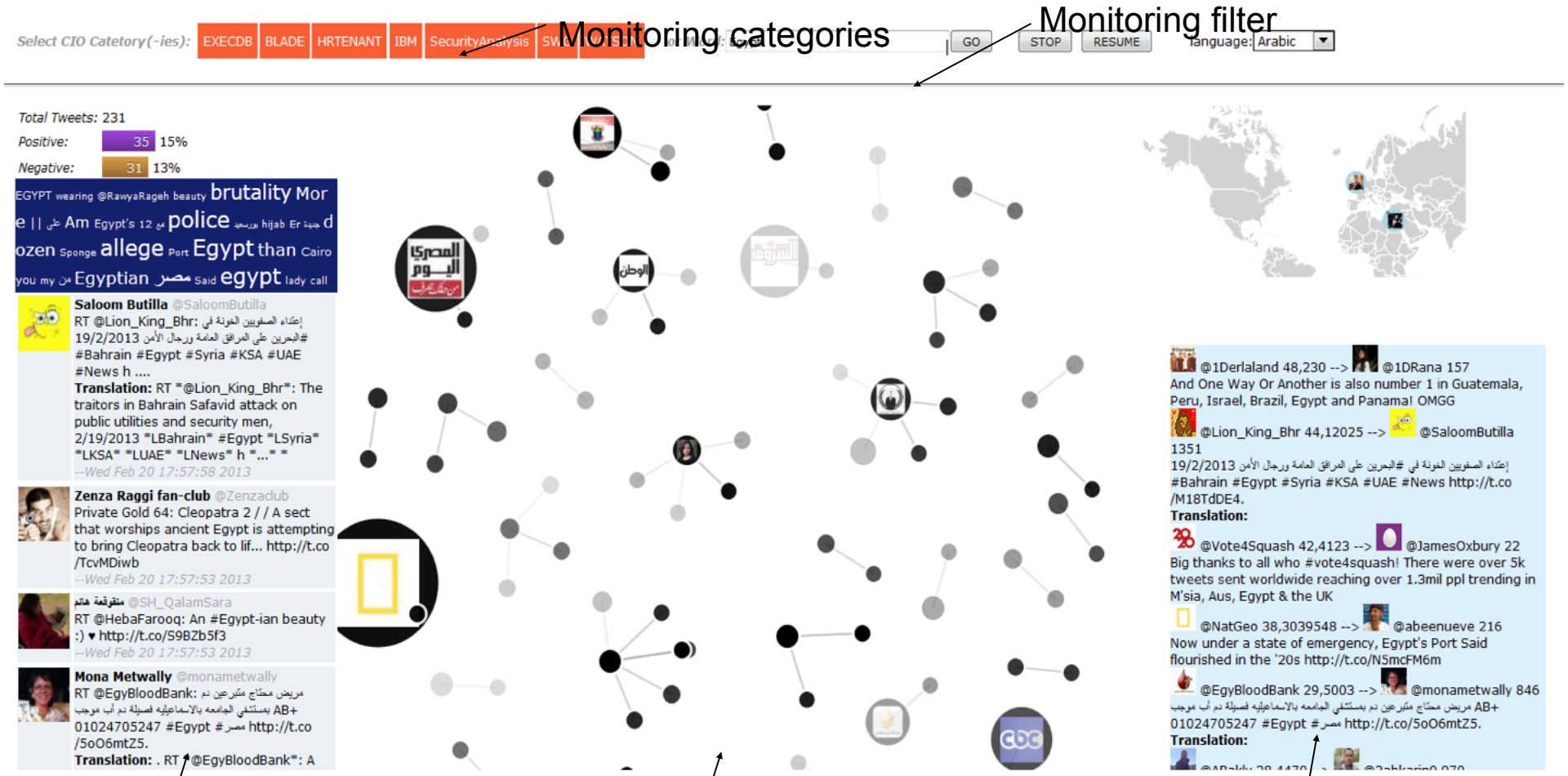
- **X-Government:**
  - A government in Asia

- **Client needs:**
  - Foster society diversity and fusion: Need to understand the citizens' responses to a government policy on social welfare on minority
  - Detect growing topic trends and multimedia discussions around the policy

- **Challenges:**
  - Detecting emerging trends as well as the powerful multimedia memes
  - Existing tools lack of predictive capability of tweets' potential influence

Live Monitoring

Live Monitoring



Live Tweets, Sentiment, Keywords

Growing Influential Retween Graphs

Real-Time Translation, Locations, Top Retweets


Trend Monitoring

Trend Monitoring

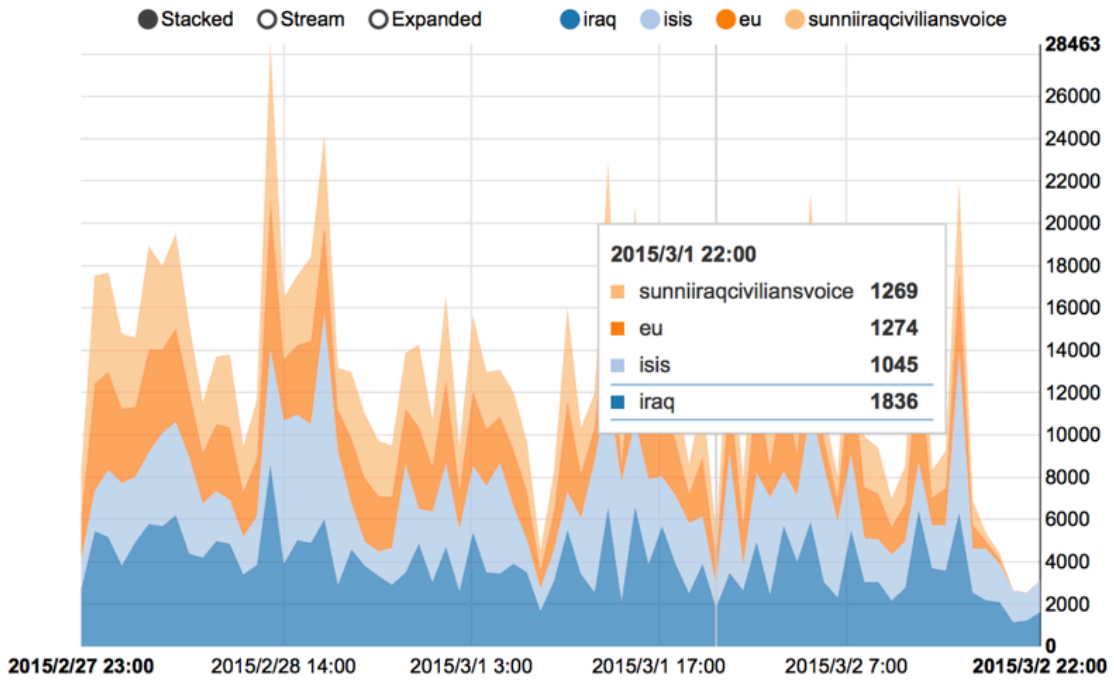
Real-time Trend Analytics

ISIS | Turkey | Top Hashtags 4 | Days 3 | Update

The chart shows top **4** hashtags for **'isis'** topic durning **February 27, 2015** to **March 2, 2015**

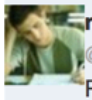
(**285667** tweets for hashtag **iraq** loaded. **236532** tweets for hashtag **isis** loaded. **199351** tweets for hashtag **eu** loaded. **198716** tweets for hashtag **sunniiraqciviliansvoice** loaded.)

Click on the histogram below to see raw tweets.



Showing the tweets for **iraq** on **2015/3/2 16:00**

giddey_up
@giddey_up
"@thetimes: #Iraq troops begin major offensive to drive #Isis out of Tikrit http://t.co/Xyjy7x8iFm http://t.co/4UJxfjSzuU" Gogetm!
-- 2015-03-02T22:44:46Z



rashidrashidi
@rashidrashidi5
RT @paige_antoine: 'Mullahs' regime Deputy President, Foreign Minister visit #Iraq to skirt sanctions and transfer huge http://t.co/nrwtQJ…
-- 2015-03-02T22:44:47Z

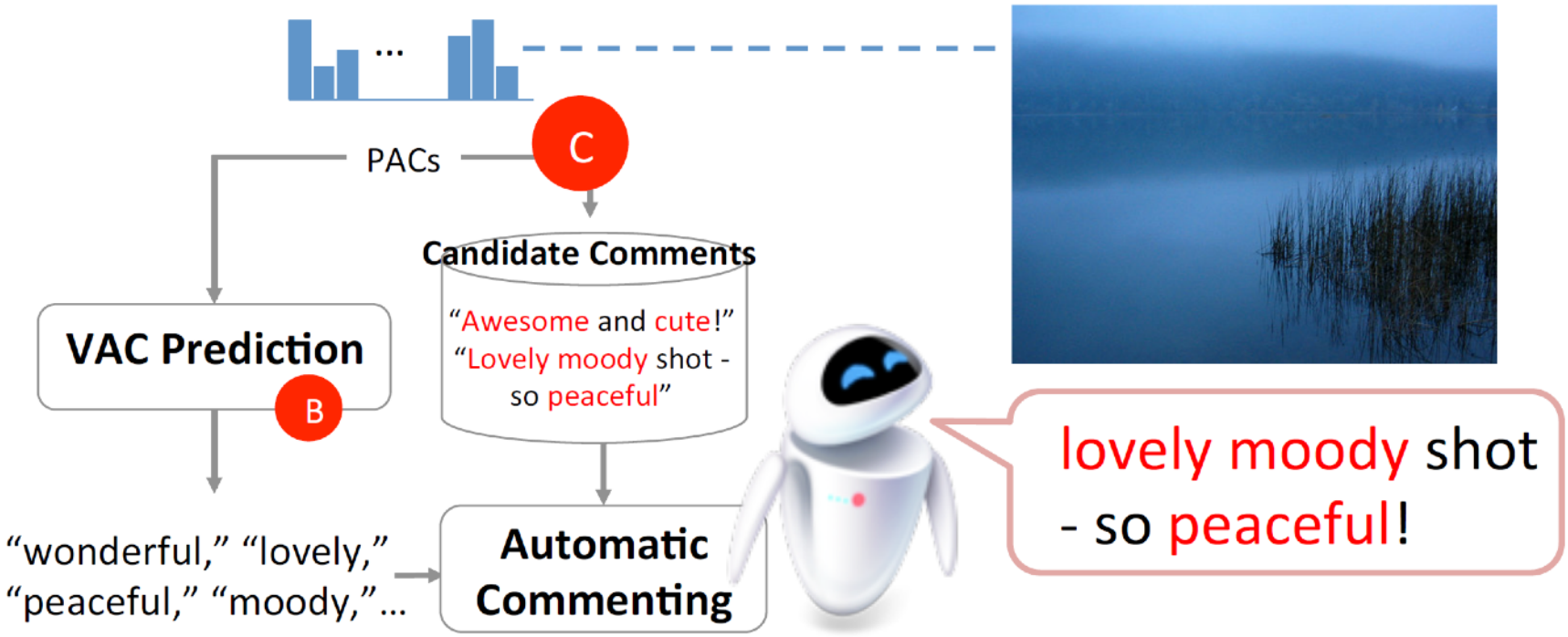
Befree
@BEFREEinFL
RT @AlanTonelson: BREAKING: #Obama interviewed by @Reuters on #Netanyahu, #Iraq, #nukes: http://t.co/OZ8Ne5RhGM
-- 2015-03-02T22:44:48Z

Chris Anoya
@ChrisAnoya
RT @Matt_VanDyke: Brave soldiers of the Nineveh Plain Protection Units (#NPU) training to defend Christians against #ISIS in #Iraq. http://…
-- 2015-03-02T22:45:29Z

HaSSIe-Proctor
@proctor_hassie

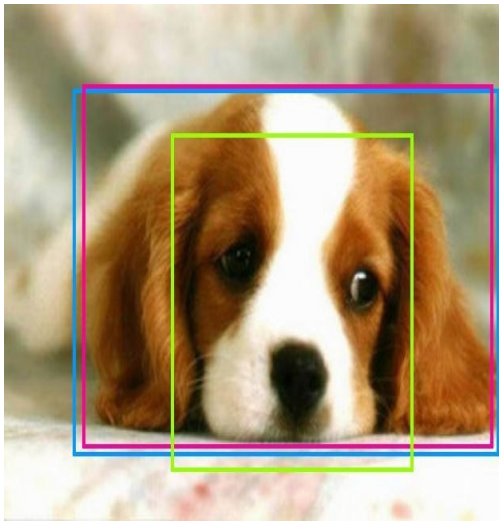
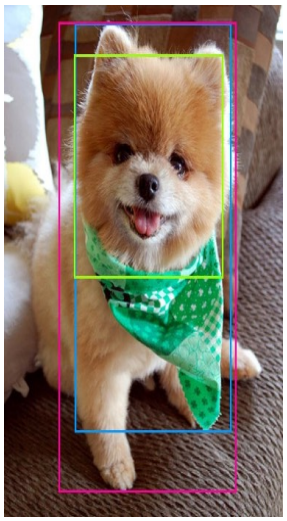

Multimedia Monitoring

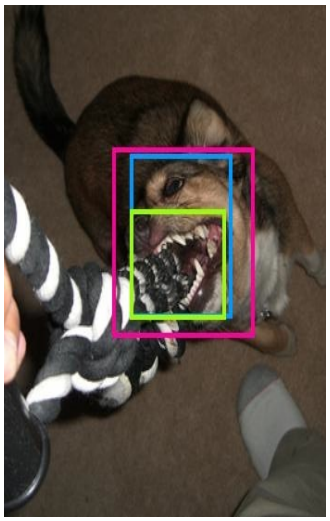
To be integrated soon — Automatic Affective Comment



What and Where of Image Sentiment Analysis

What's in the image that makes a dog cute vs. scary?

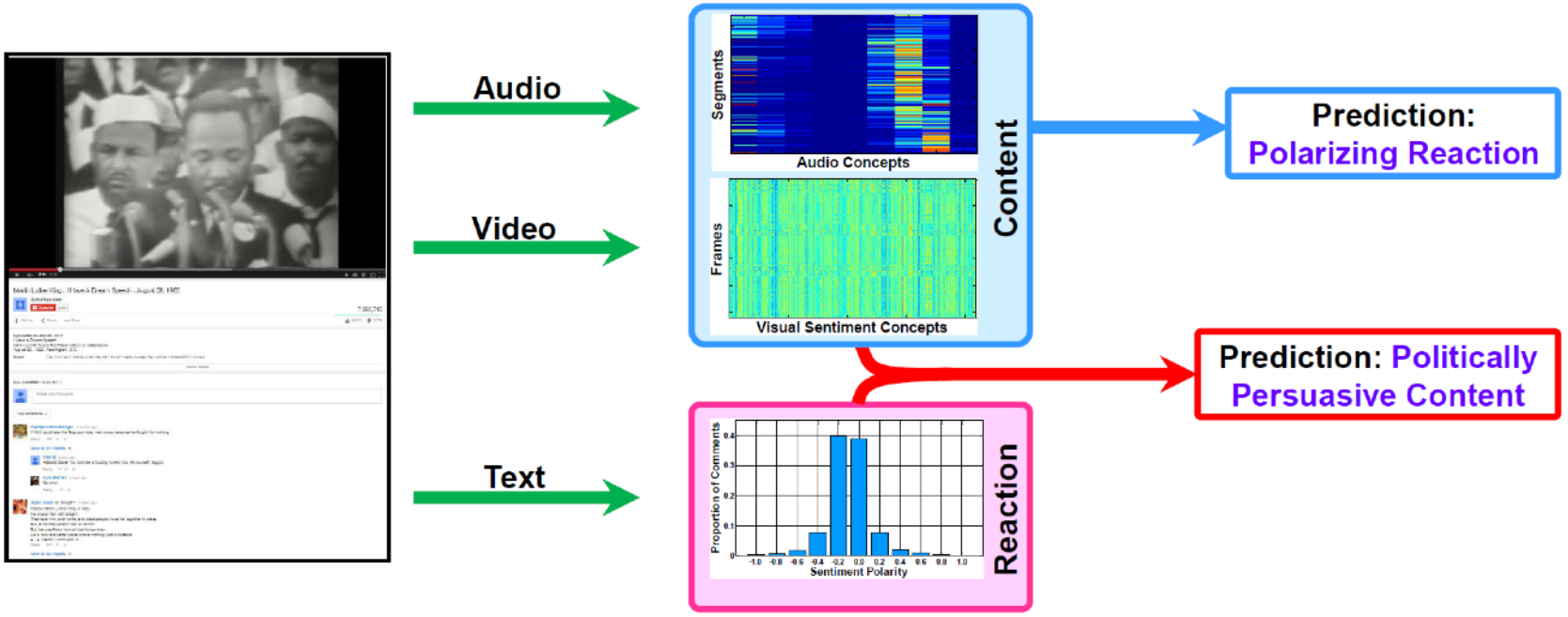





What makes a beautiful landscape beautiful?


Detecting Politically Persuasive Web Content

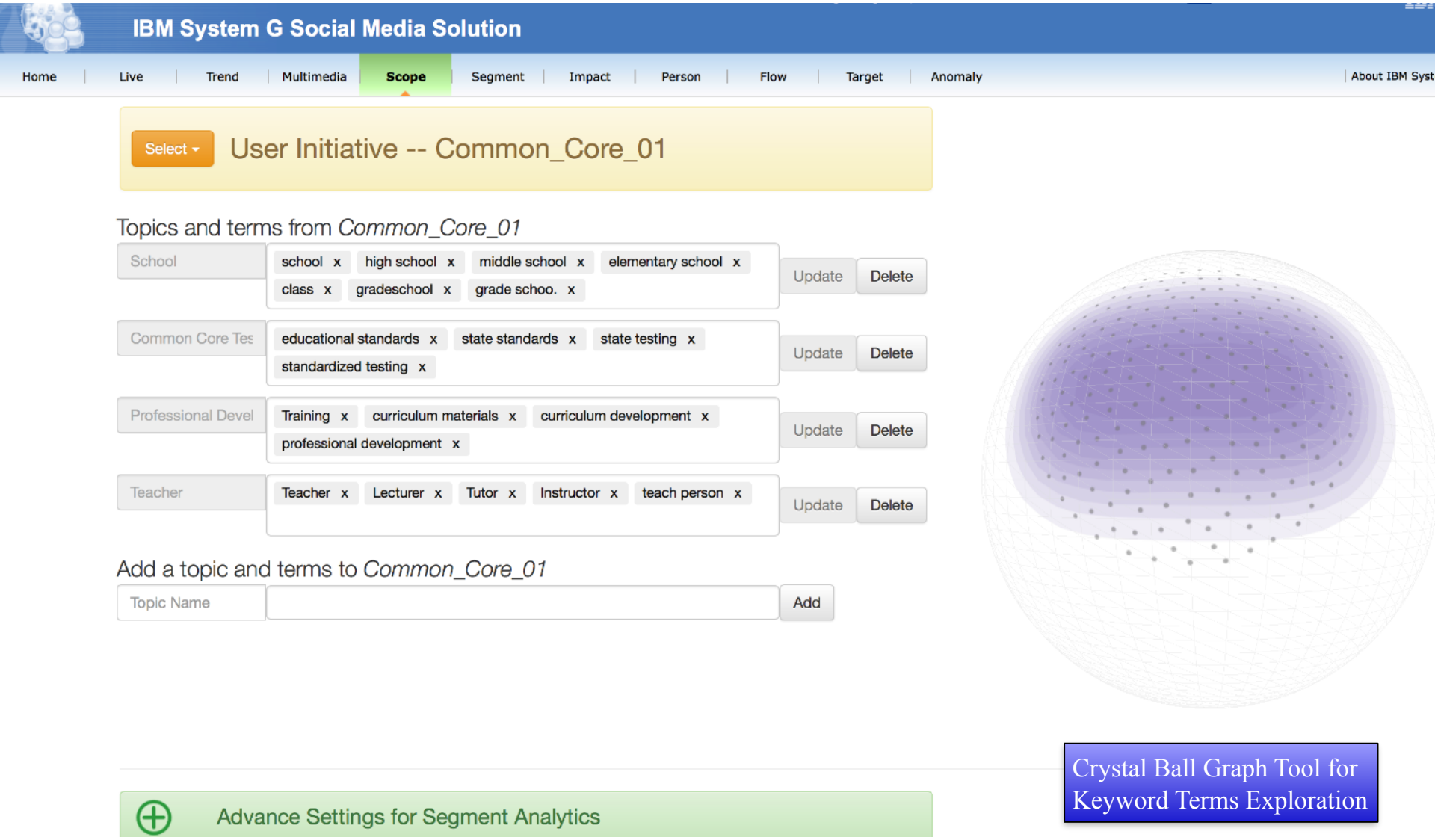
Summary



- **Extract Affective and Semantic Information**
  - Audio, Visual and Textual Modalities
- **Detection of Politically Persuasive Content**
  - Additionally predict viewer response in advance


Scope Identification

Scope Discovery



Crystal Ball Graph Tool for Keyword Terms Exploration


Segment Analytics

Select ▾ User Initiative -- ISIS_07

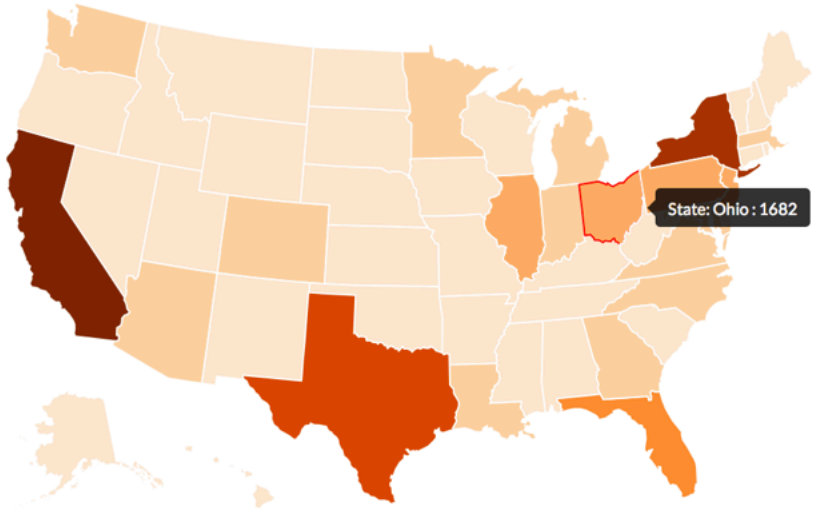
Map Hot Topics Topic Anomalies Segments

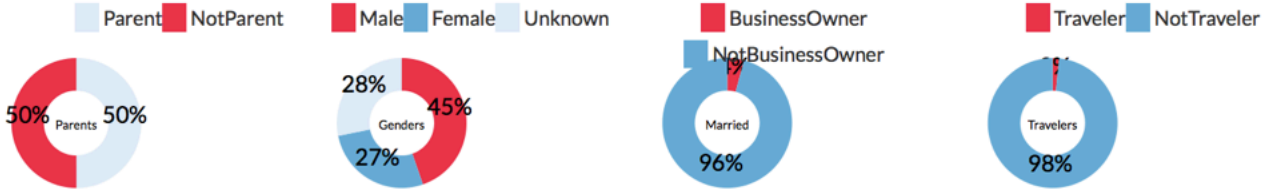
Initiative: ISIS_07

Foci

_ISIS_Name_Occurrences	21426
_ISIS_in_Iraq_or_Syria	9734
_ISIS_Marriage_Interest	108
_ISIS_Extreme_Military_Action	266
_ISIS_Military_Action	833
_Syrian_Opposition_Groups	15162
_ISIS_Social_Media_Recruiting	1727
_Islamic_Religion_Interest	16895
_Western_Political_Response_Interest	3078
_ISIS_Travel	8969

U.S. Only | Worldwide





Impact Prediction

Impact Prediction

Real-time hashtag monitoring

Predicting the business impact of tweet messages grouped by hashtags.

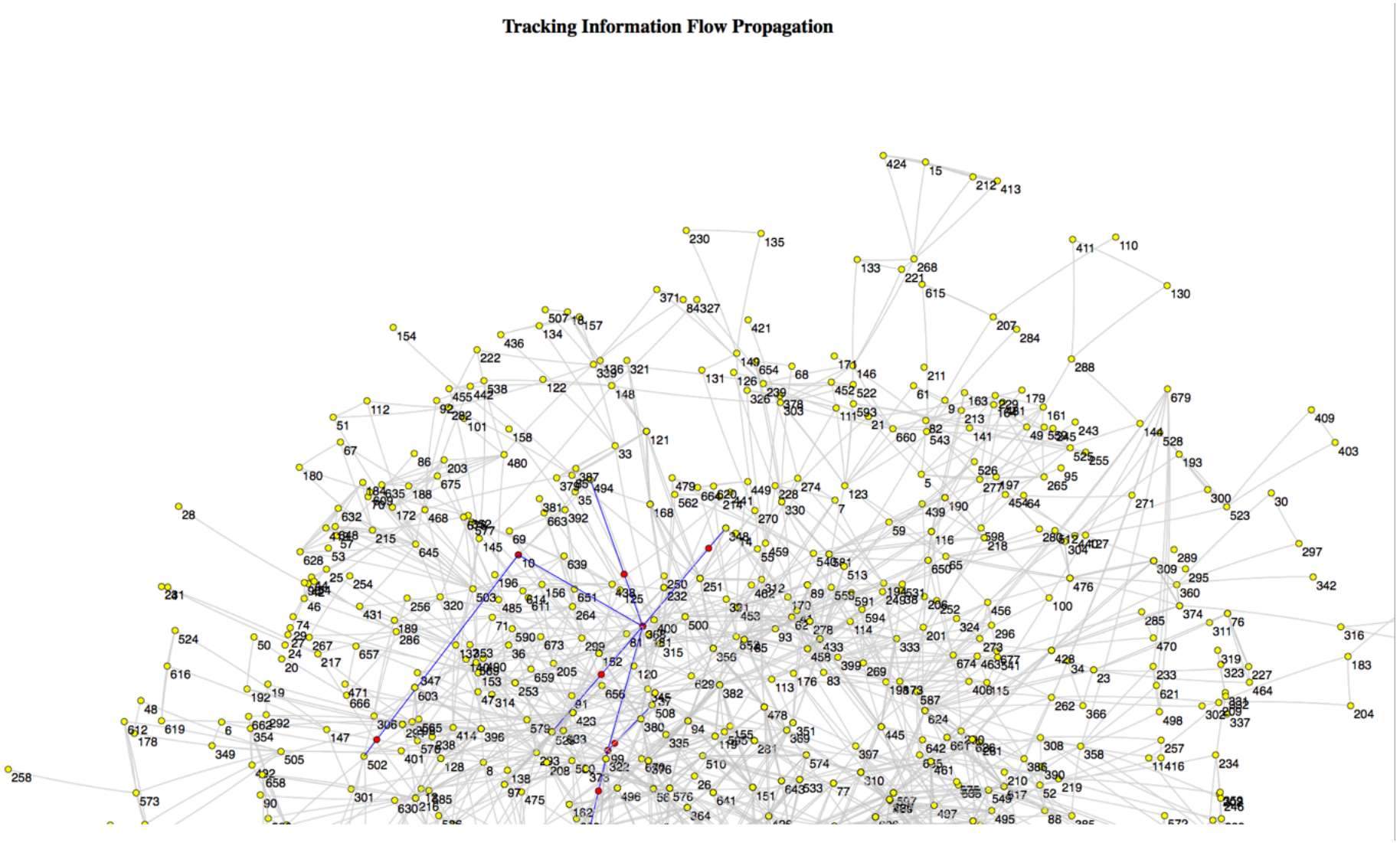
Please click on the "hashtags" to learn more about each conversations content.
Last updated at 2015-03-03 03:20:01 GMT

Sort By Date

#	Conversations	Impact	Impact Score	Prediction Detail	First Tweet Time	Last Tweet Time	Duration
1	russia, turkey	HIGH	52.93	URL	2015-02-18 17:36:59	2015-03-01 09:10:57	255 hours
2	isis, syria	HIGH	44.66	URL	2015-02-20 14:29:51	2015-03-01 08:58:20	210 hours
3	isis, syria, turkey	HIGH	42.1	URL	2015-02-20 07:12:33	2015-03-01 08:58:20	217 hours
4	syria, turkey	HIGH	40.77	URL	2015-02-27 14:31:05	2015-03-01 09:13:34	42 hours
5	erdogan	HIGH	39.92	URL	2015-02-27 23:35:06	2015-03-01 09:12:05	33 hours
6	syria, turkey, us	HIGH	38.28	URL	2015-02-19 02:45:53	2015-02-28 11:50:52	225 hours
7	erdogan, turkey	HIGH	36.89	URL	2015-02-24 02:16:10	2015-03-01 09:12:05	126 hours

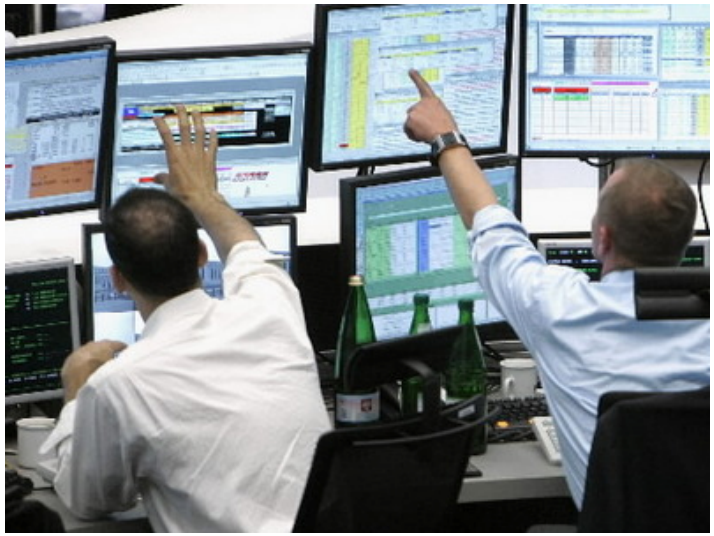

Flow Analytics

Flow Analytics



Collective intelligence and predicting market movement

Data:
- instant messages and trades by employees of a large hedge fund
- 24 traders, 95 analysts, 63 portfolio managers, 8646 outside contacts
- 47K trades
- 12 million IMs (2010–2011)

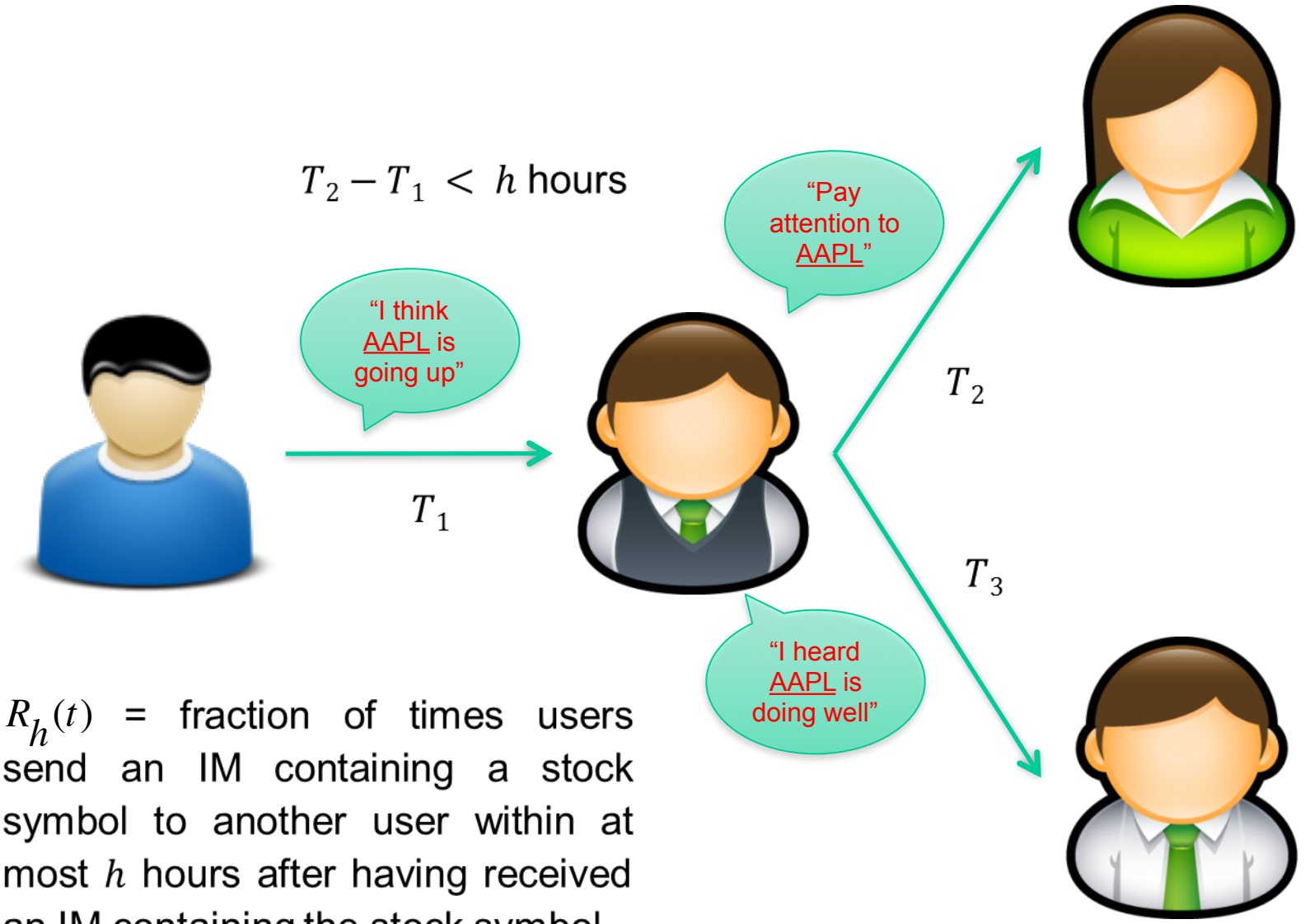


Findings: We identify two behavioral patterns that correlate with changes in the market:
- Reaction to IMs containing relevant information
- Attention to people who work with the same/different stocks

Using these two signals, we can make predictions of the market movement with better accuracy than the hedge fund did

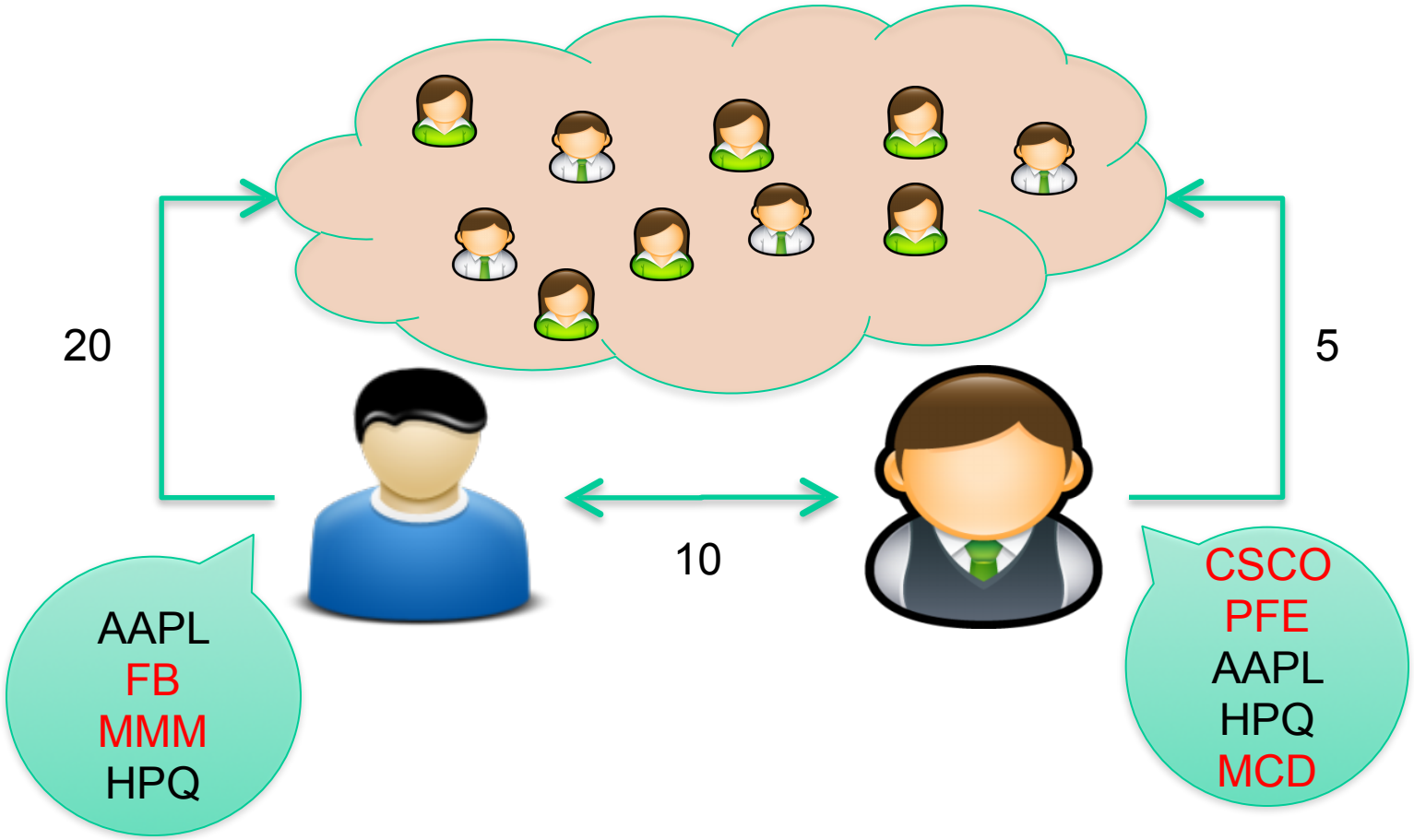
13:11:33, I was thinking all this AAPL anti-trust might be actionable
13:11:42', not great for AAPL
13:11:47, When GOOG had that big issue in Europe stock underperformed right?
13:11:52, true
13:14:01, Also not sure if you caught, but GSCO is going to allow employees to bring own phone device for corporate email
13:14:24, Maybe GSCO allowing that could be positive for AAPL, as security focused firm saying iPhone works
13:14:35, But bad for RIMM
13:14:42, Maybe all this is priced in
13:16:50, Did you see speculation that Bing is actually quietly going to be default search on iPhone 4?
13:17:18, heard a lot of talk of that
13:17:23, but didn't see that specifically like that
13:17:44, Okay let me figure out where I saw that and get back to you
13:17:45, One sec
13:20:08, http://thenextweb.com/apple/2010/06/07/wait-bing-is-default-search-on-iphone-4

Relay: How urgently do employees relay information?



$R_h(t)$ = fraction of times users send an IM containing a stock symbol to another user within at most h hours after having received an IM containing the stock symbol

Concentration: Do employees focus on others who discuss the same stock symbols?



$C_{(A,B)}(t)$ = fraction of messages between A and B and all messages of A and B

In example: 10/(20+10+5) = 0.29

$C_k(t)$ = average $C_{(A,B)}(t)$ among all pairs (A, B) that share k stock symbols

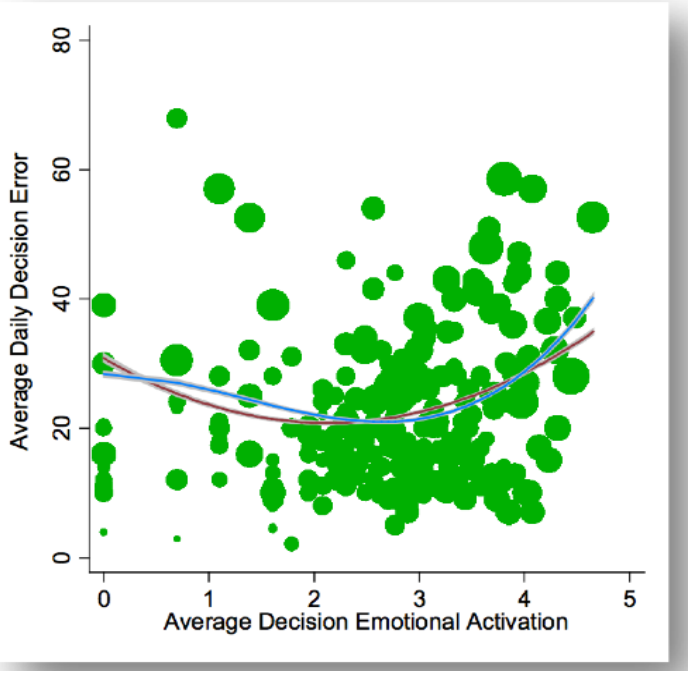
Emotional Activation and Information Processing (Northwestern University)

- Motivation

  Emotional states can effect how information is processed. Good information can be undermined or strengthened by emotional states.

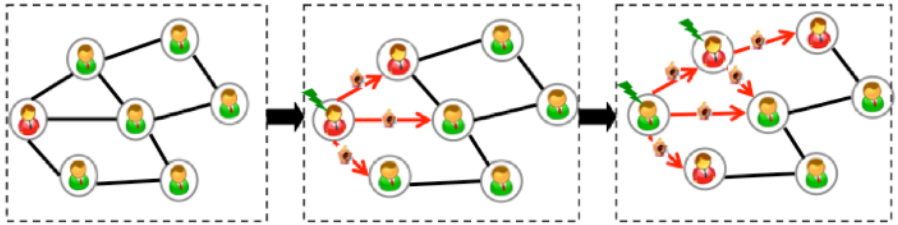
- Approach
  - Measure emotional activation in tweets using the ANEW dictionary
  - Control for the number of words in the text

- Preliminary experiments
  - Traders at a hedge fund are more likely to make decision errors when they are very emotionally activated or very emotionally deactivated.
  - Users who retweeted the 20 detected most anomalous sequences tend to post tweets with higher level of emotion than a baseline of 20 million tweets from June 2009.



Finding optimal edges to slow down propagation

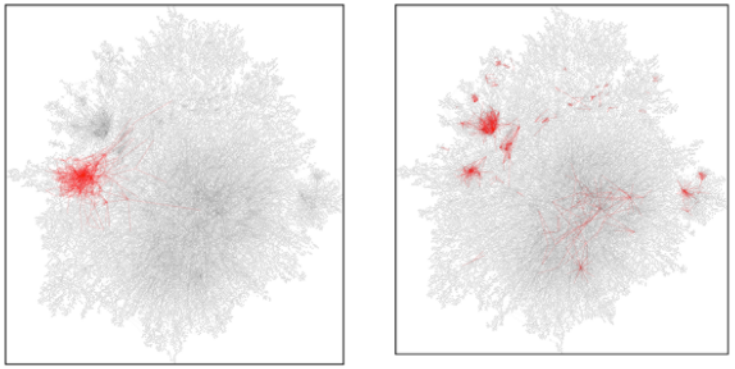
- **Task:** Minimize propagation by deleting edges of a graph



- **Problem:** Given a graph *G and* a budget *k*, find the *k* edges to delete in order to get the largest drop in the leading eigenvalue λ_1 of *G*

Yahoo IM!

NetMelt *MET*



- Red edges are selected for deletion
- *NetMelt* [Tong *et al.*, CIKM 2012] tracks one eigenvalue, λ_1
- Our *MET* tracks on average 5.17 eigenvalues

Person Analytics

Measuring Human Essential Traits in Social Media

- **Personality:** Mapping personal/ organizational social media postings to scores of BIG 5 Personality (*Openness, Conscientiousness, Extraversion, Agreeableness, and Neurocism*)

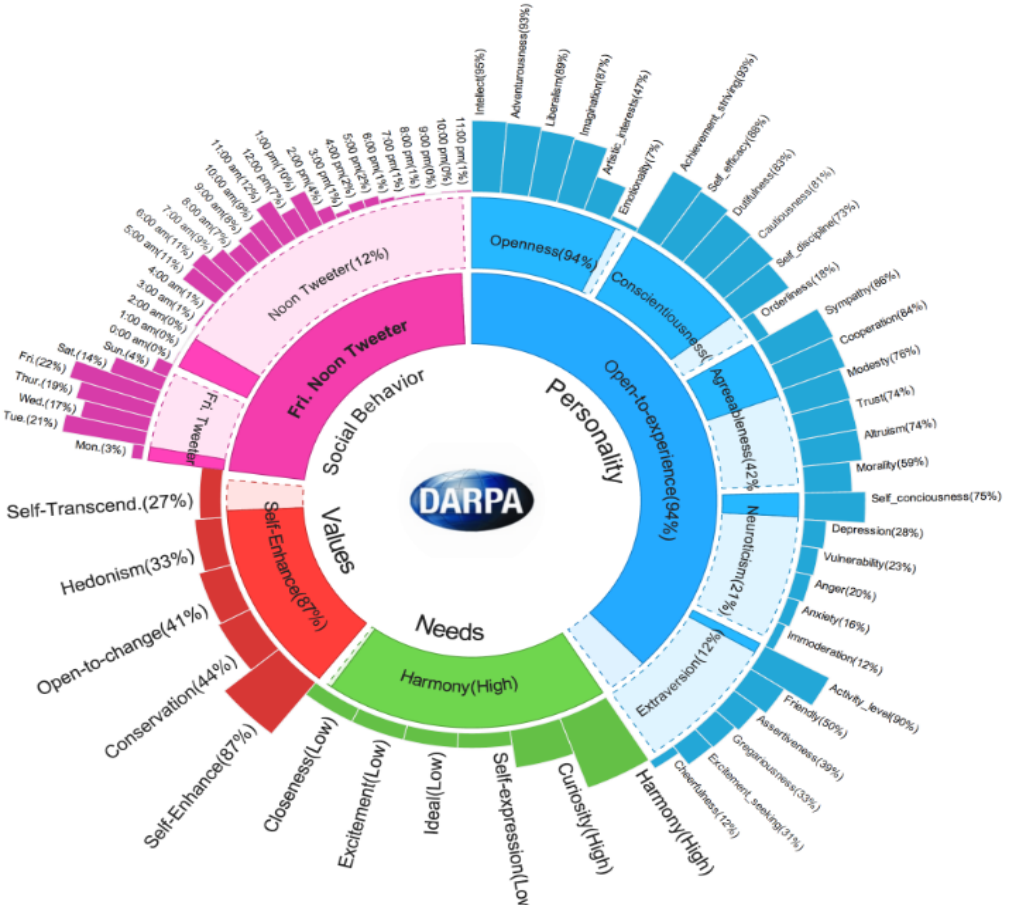
- **Needs:** Mapping personal/organizational social media postings to scores of *Harmony Curiousity, Self-expression, Ideal, Excitement, and Closeness.*

- **Values:** Mapping personal/organizational social media postings to scores of *Self-Enhance. Conservation, Open-to-Change, Hedonism, and Self-Transcend.*

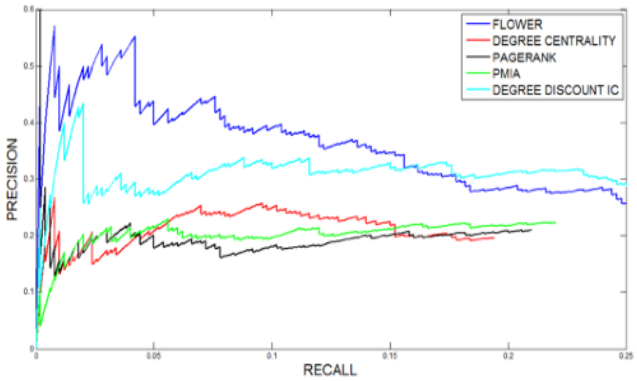
- **Trustingness and Trustworthness:** Deriving from *interaction and propagation* history between the user and his followers and the people he follows.

- **Influence:** Total *attention* received by user as leader across all discovered flows.

==> Preliminary studies showed propagation behavior is related to these social cognitive traits

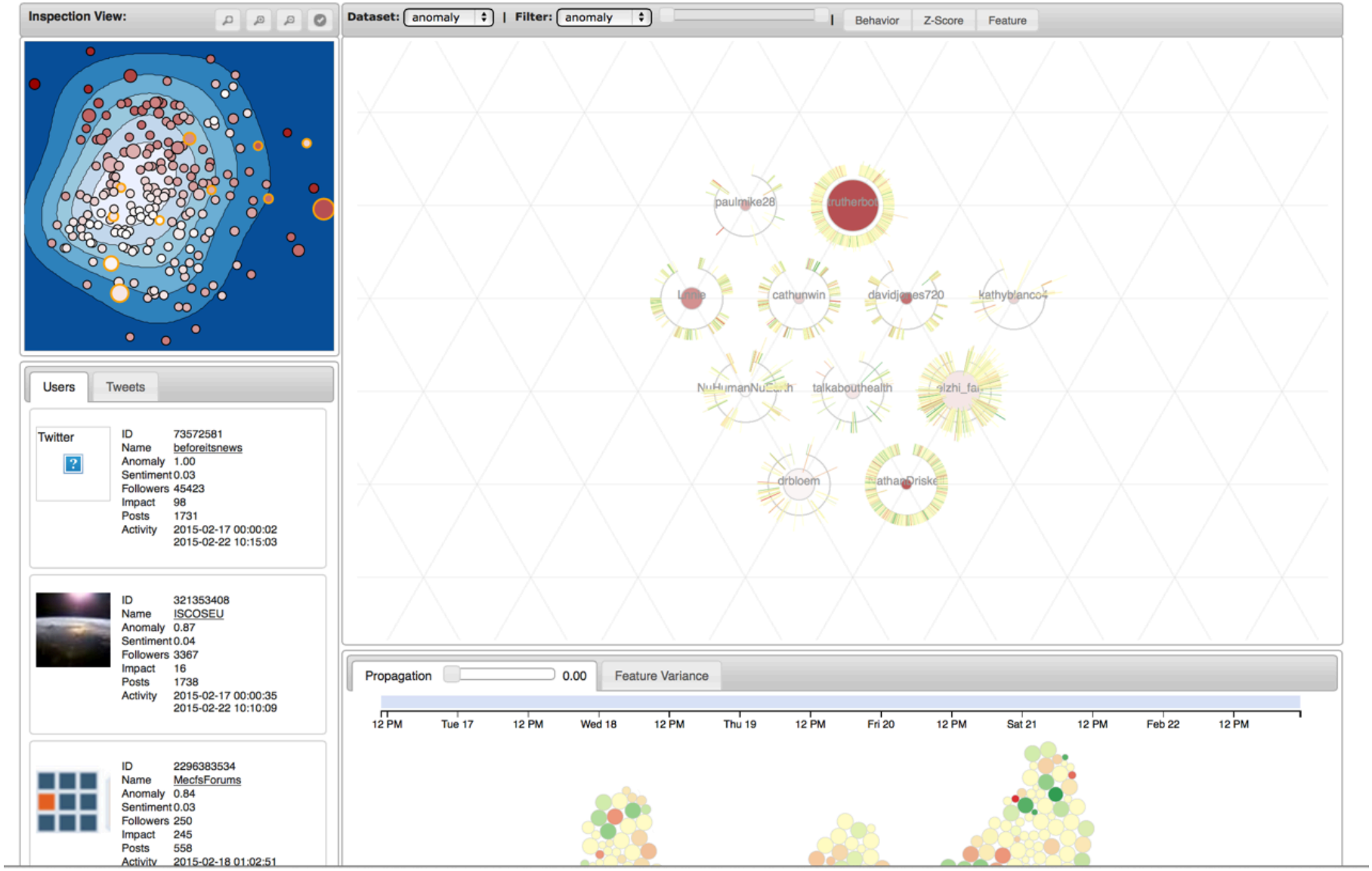


Precision-Recall performance of predicting info propagation by different features
(Our proposed influence index: FLOWER)




Target Discovery

Target Discovery



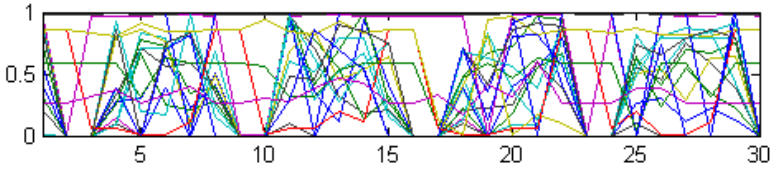

Anomaly Detection

Detecting Anomalous Information Spreading

- Motivation
  - People's dynamic reactions to information (e.g., retweeting) give clues to information credibility and quality
    - For example, trustworthy people may take time to verify uncertain information from strangers before spreading it
- Approach
  - Use one-class conditional random field to model people's behavior in information spreading sequences and detect anomalous sequences
    - Features: content features such as emotion, network features such as tie strengths and clustering coefficients
- Preliminary experiments
  - Detect anomalies in retweeting sequences during Hurricane Sandy
    - Including hijacker, fake pictures spreaders

Detected as top 1 anomaly in Sandy Tweets

One-class CRF to detect temporal anomalie



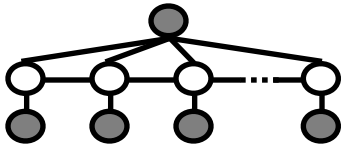
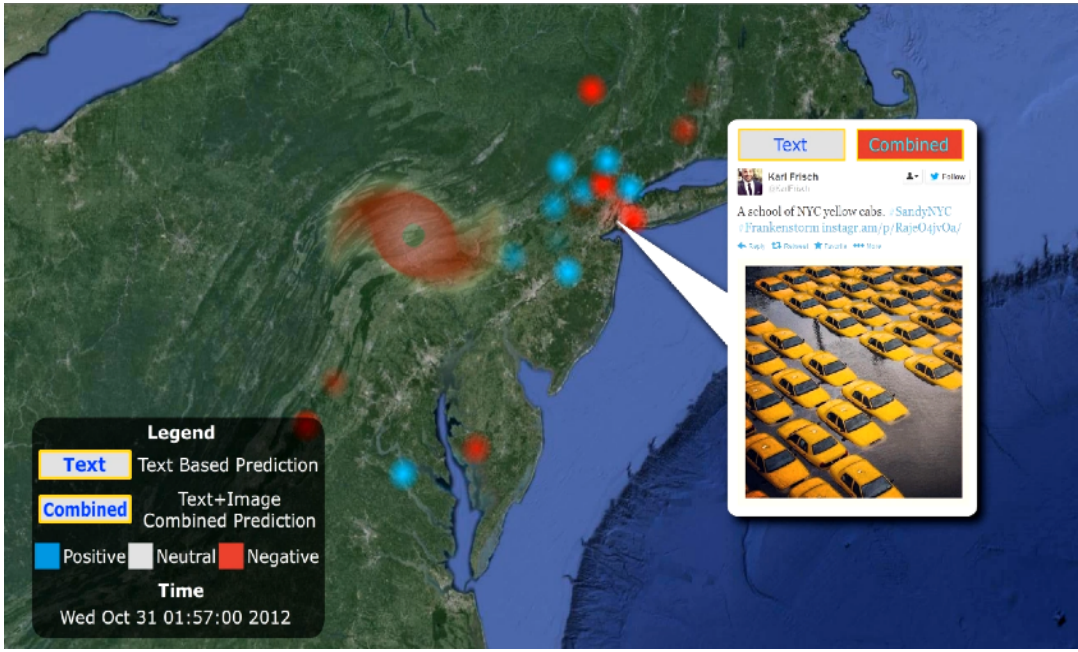
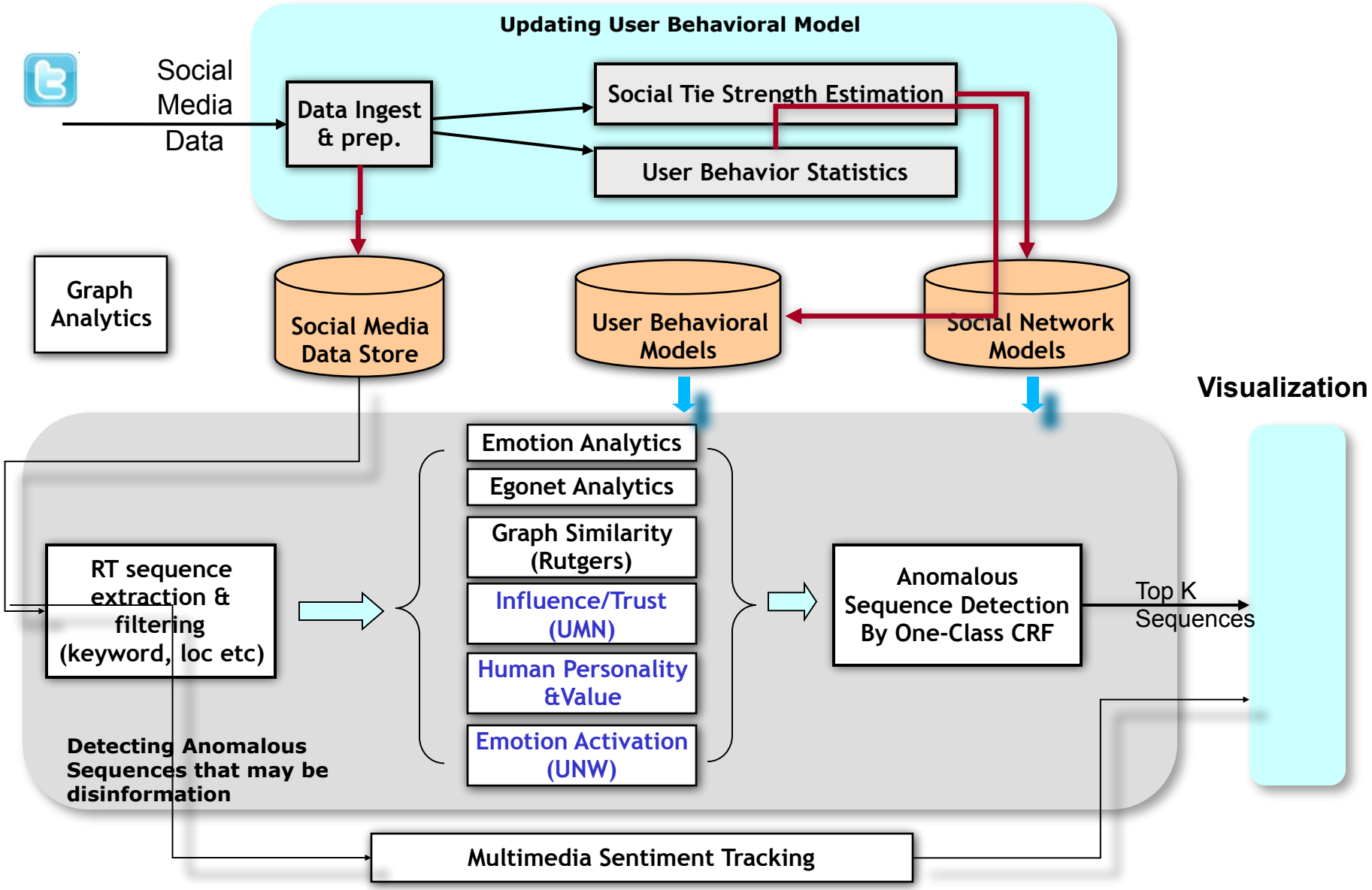


Photo Tweet Sentiment Tracking during Hurricane Sandy (Columbia University)

- **Goal:** Detect sentiment during Hurricane Sandy.
- **Data collection:**
  - Date: Oct 25 – Nov 02
  - Hashtags (based on popularity): #prayforusa, #frankenstorm, #nyc,#hurricane,#sandy,#hurricanesandy, #staysafe, #redcross,#myheartgoesouttoyou,…
  - 2000 Photo Tweets collected
- **Ground Truth Labeling:**
  - 1340 unanimously agreed labels from 2 individuals
- **Training Classifier:**
  - Text (SentiStrength)
  - Visual(SentiBank, Logistic Regr.)
  - Training/Testing ratio: 4:1
  - 5-fold cross-validation
  - Accuracy(Text-Visual Combined): 72%

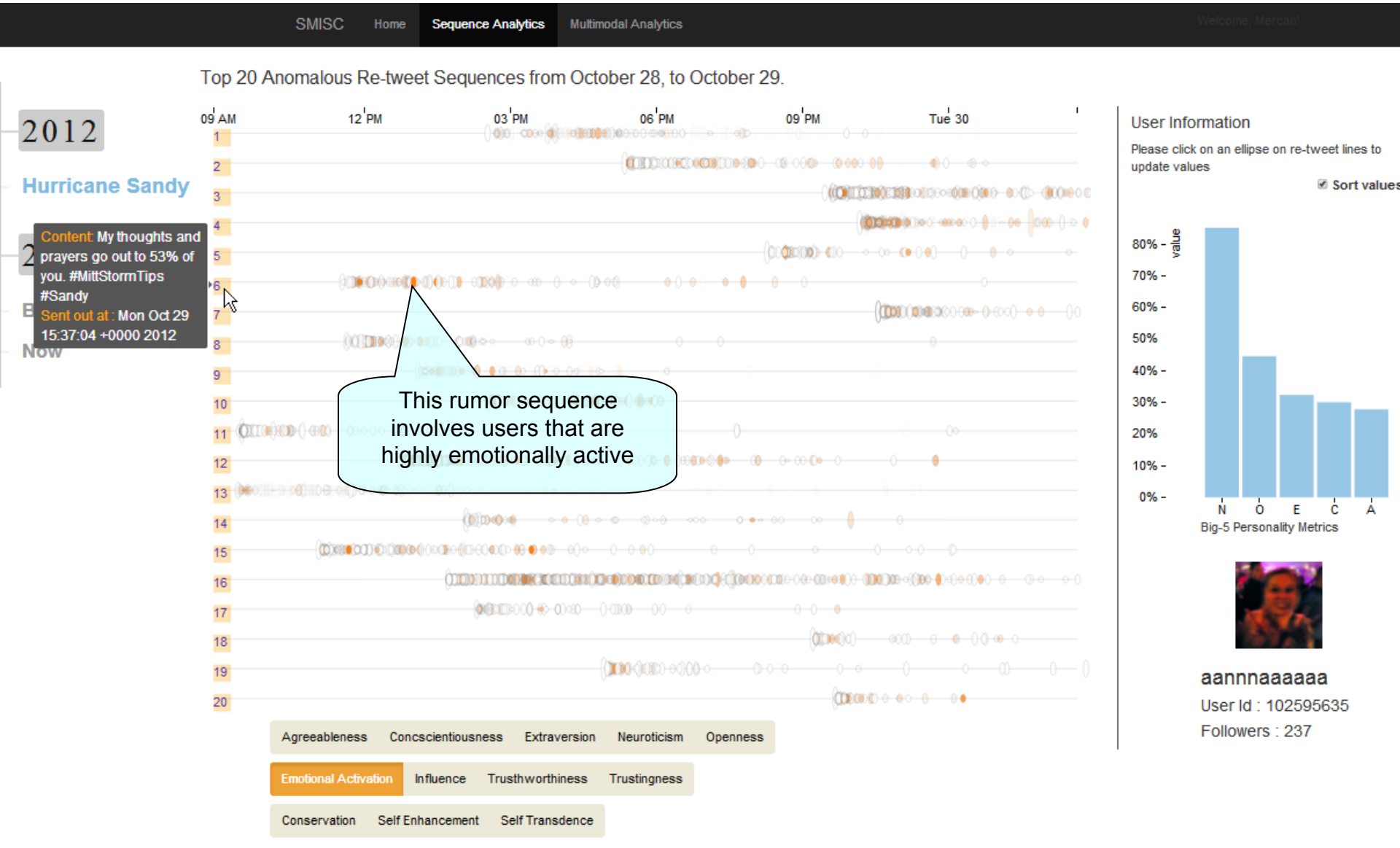


Forensic System



*Note: Components in blue are work in progress and applied to the top-K sequences

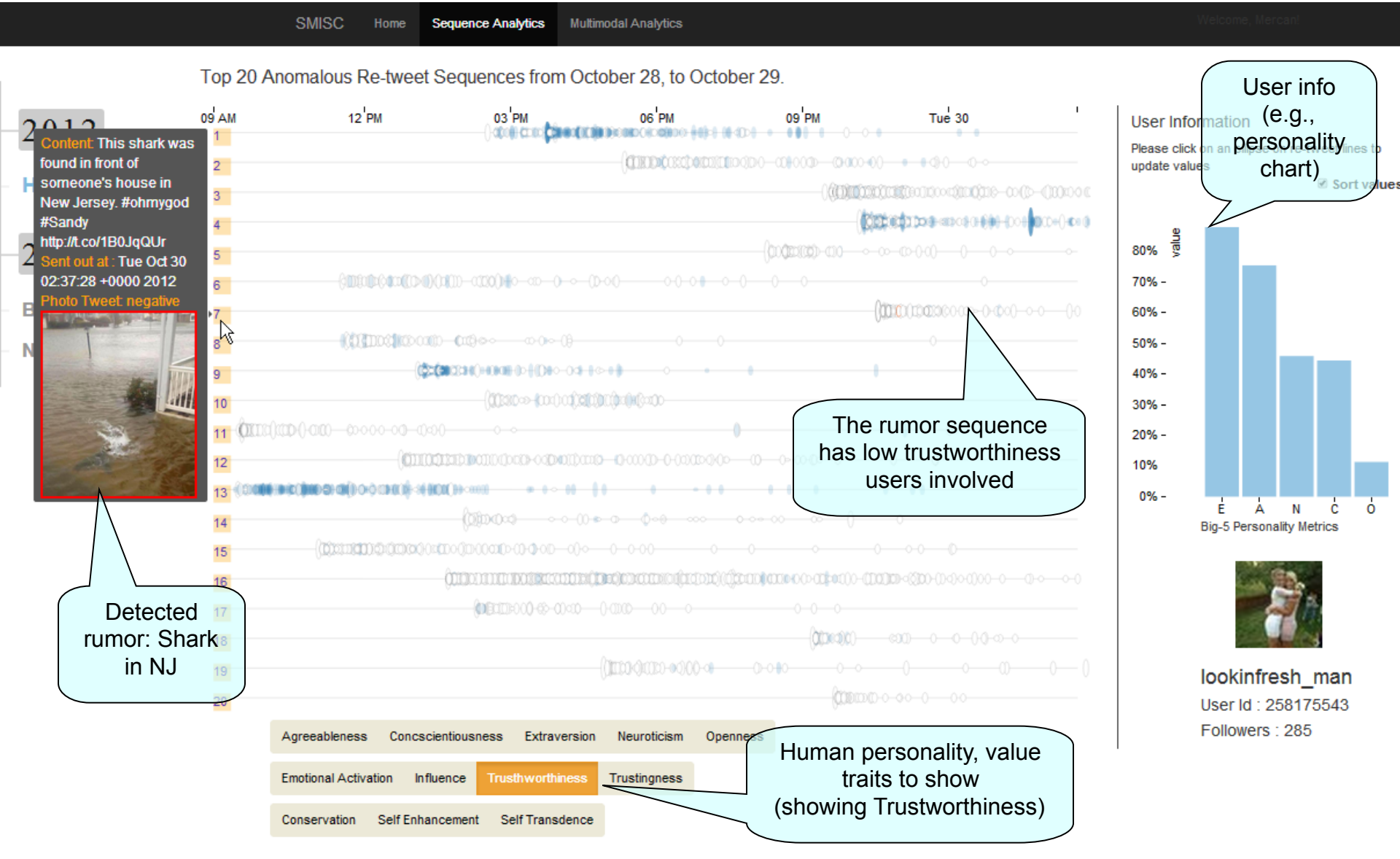
Users can interactively analyze the personality traits, emotion activation, value and other traits of the people involved



Finding Anomalies

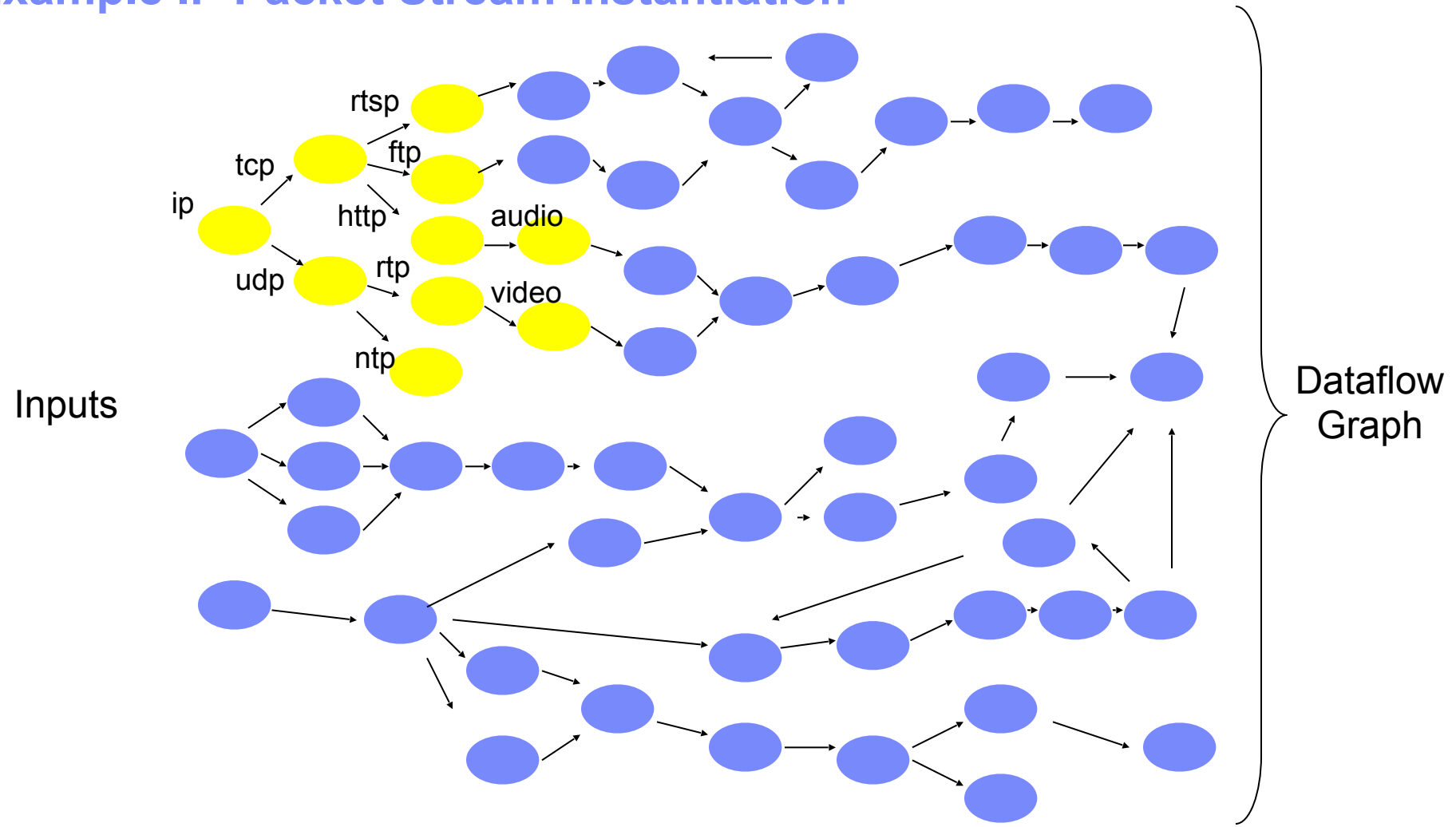
Columbia University

Detect disinformation spreading (retweet) during Hurricane Sandy by people's dynamic behavior in response to the information



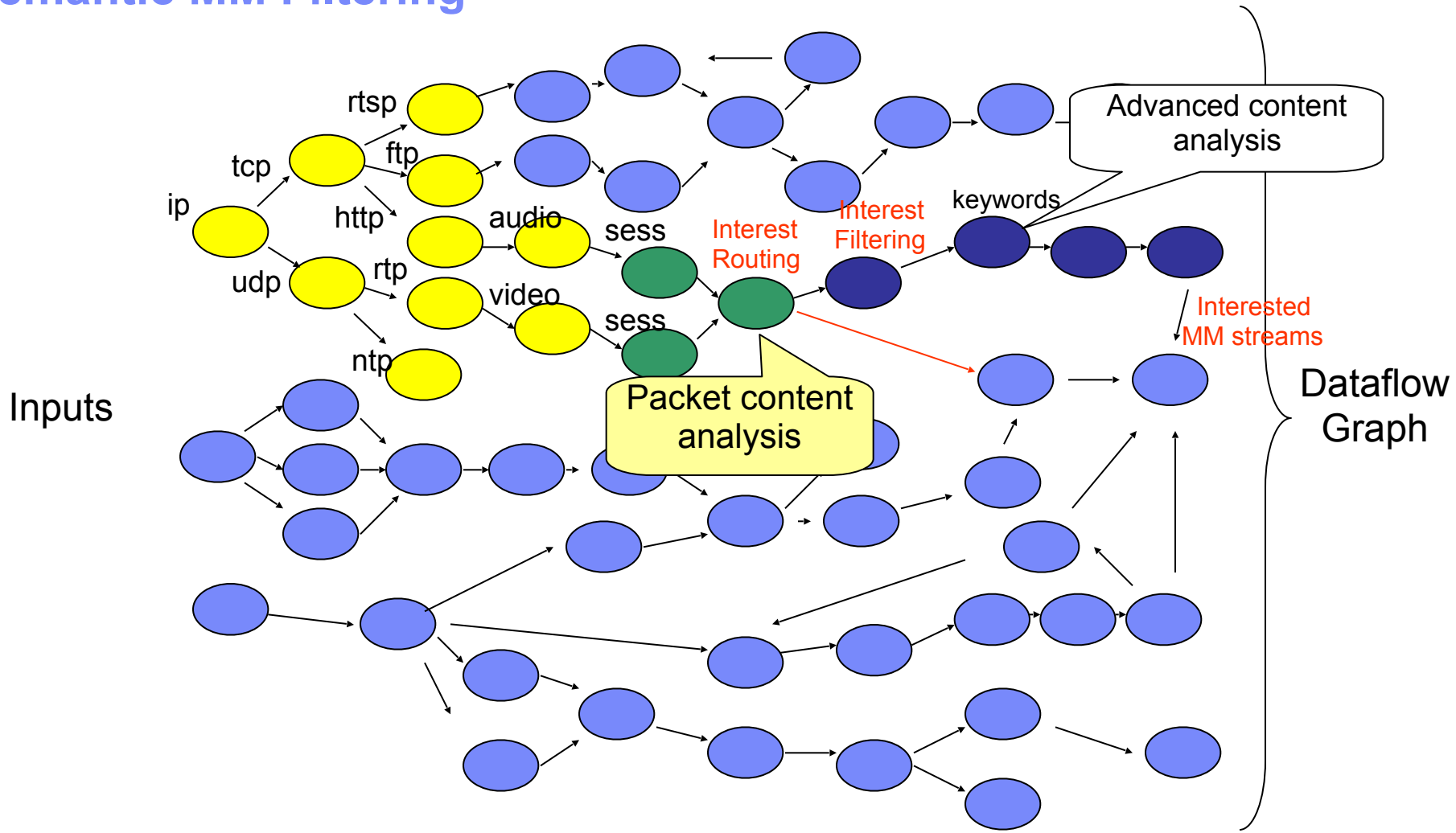
Stream Analyses Technical Challenges

Example IP Packet Stream Instantiation



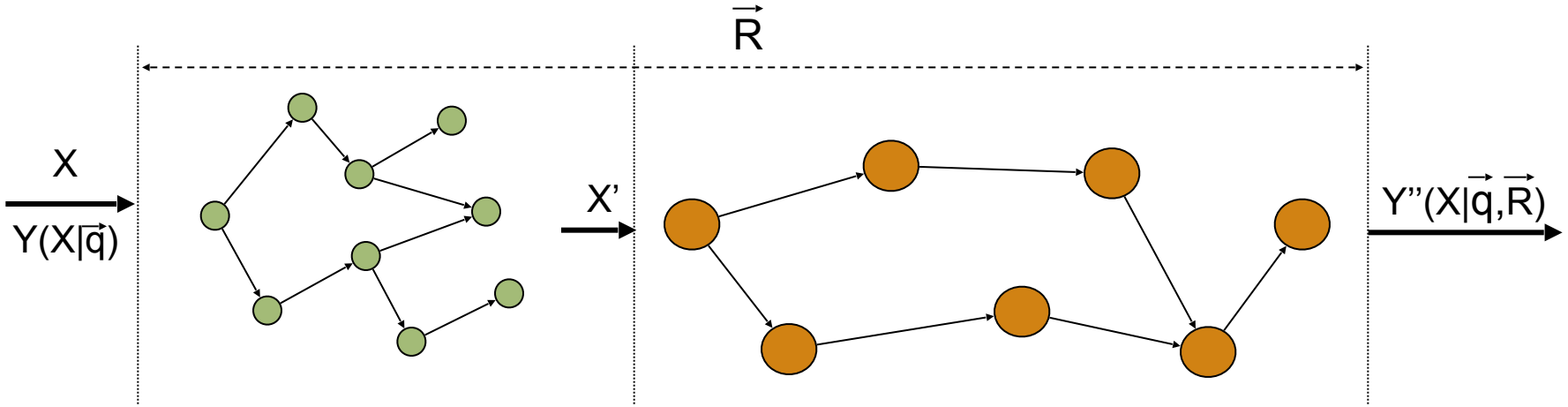
By IBM Dense Information Gliding Team

Semantic MM Filtering



per PE rates	200-500MB/s	~100MB/s	10 MB/s

Resource-Accuracy Trade-Offs



Configurable Parameters of Processing Elements to maximize relevant information:

$Y''(X \mid q, R) > Y'(X \mid q, R)$,

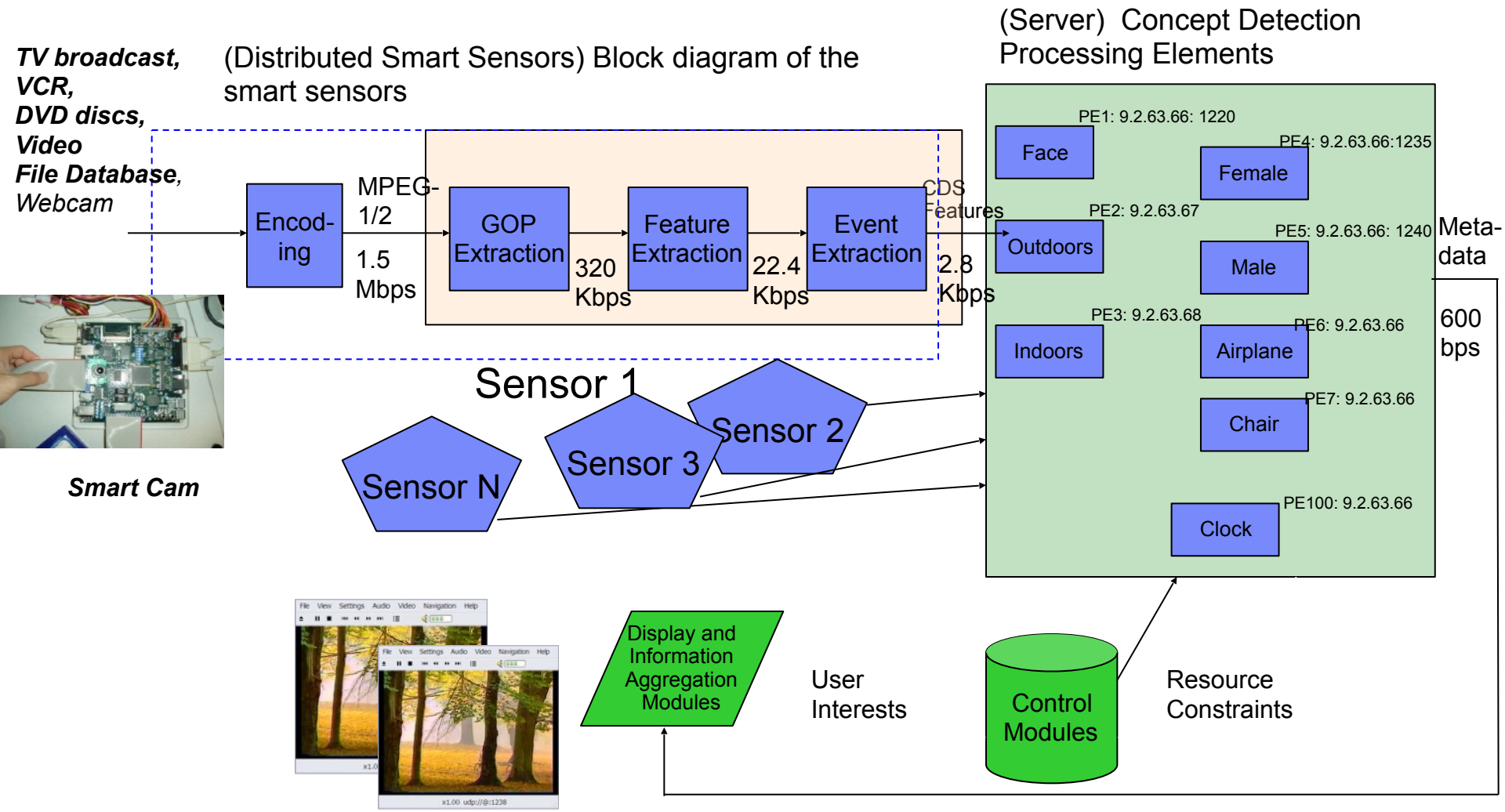
with resource constraint.

Required resource-efficient algorithms for:

Classification, routing and filtering of signal-oriented data: (audio, video and, possibly, sensor data)

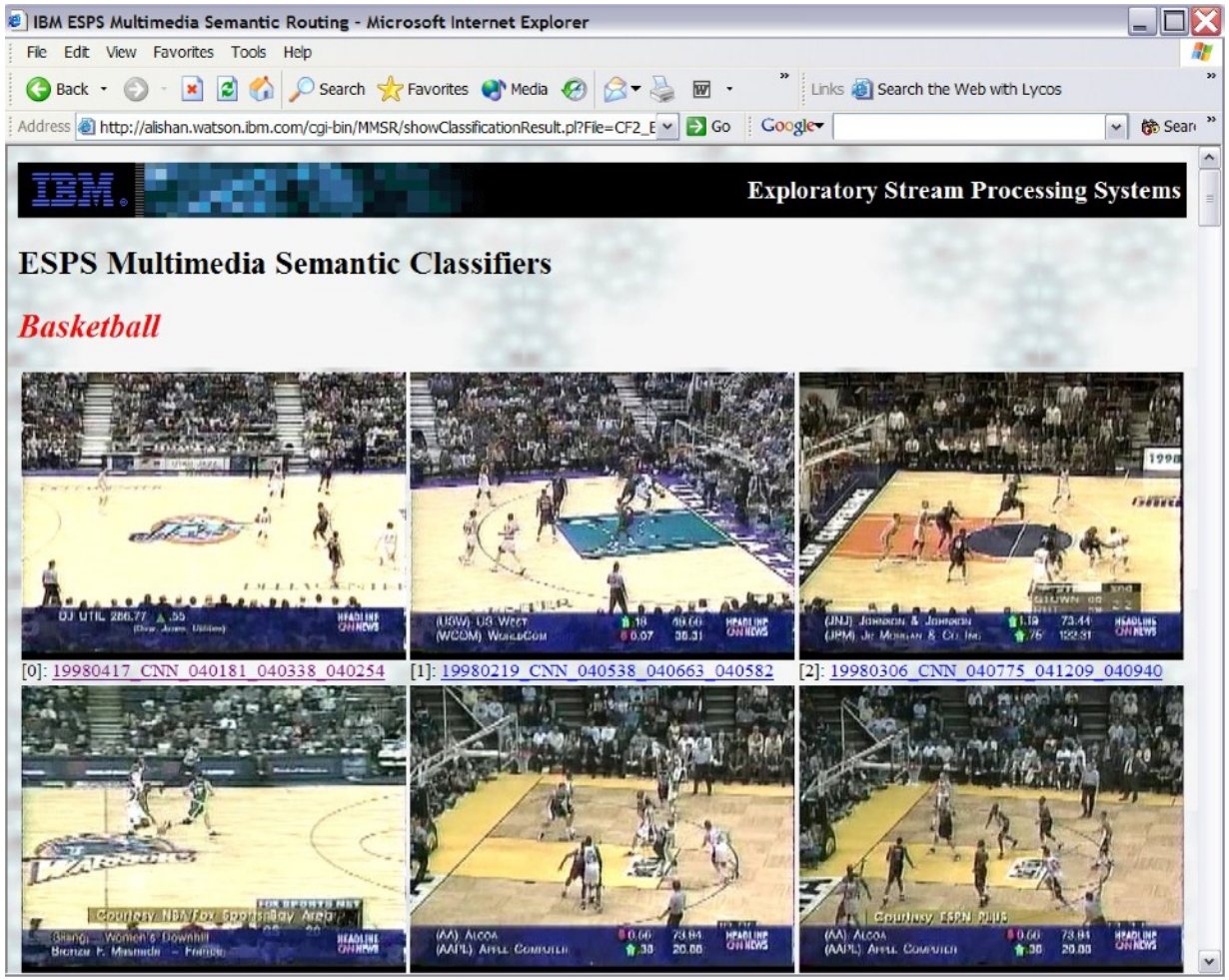
- **Input data X – Queries q – Resource R**
  - $Y(X \mid q)$: Relevant information
  - $Y'(X \mid q, R) \grave{} Y(X \mid q)$: Achievable subset given R

Example: Distributed Video Signal Understanding (Lin et al.)



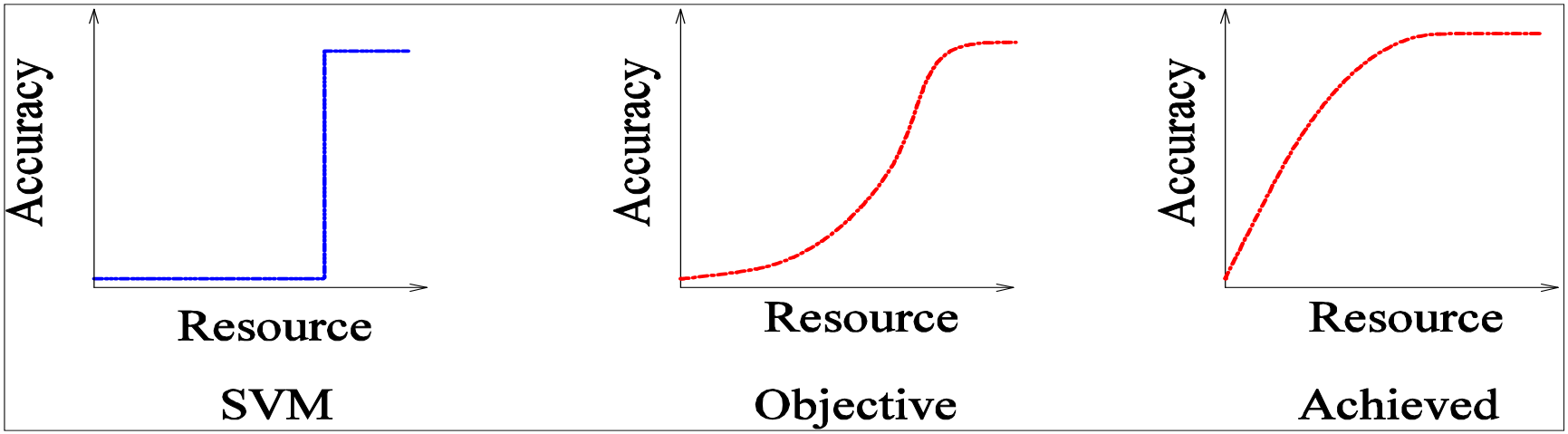
Semantic Concept Filters

E.g.:



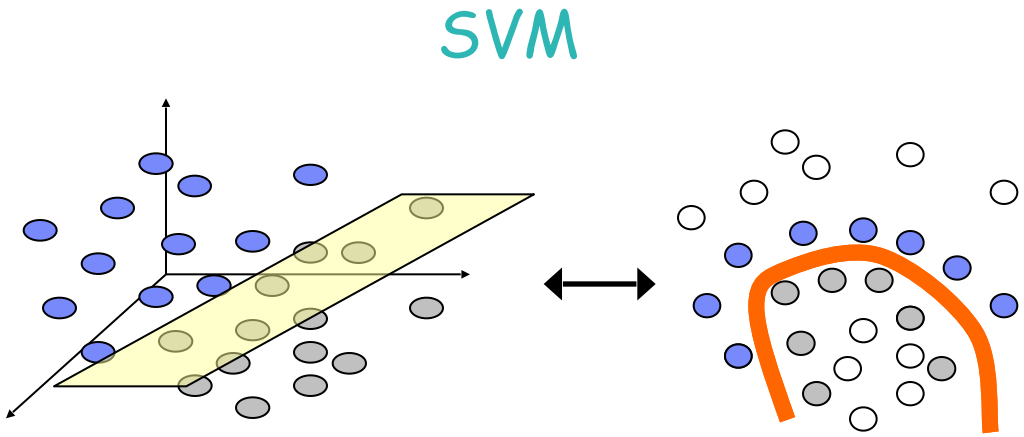
Complexity Reduction Introduction

- Objective: Real-time classification of instances using Support Vector Machines (SVMs)
- Computationally efficient and reasonably accurate solutions
- Techniques capable of adjusting tradeoff between accuracy and speed based on available computational resources



SVM formulation

- **Given :**
  - Training instances $\{\mathbf{x}_i\}$ with labels y_i
- **Objective :**
  - Find maximum margin hyperplane separating positive and negative training instances



Decision

- **Score of unseen instance** $\mathrm{u}_j : \mathrm{w} \cdot \phi(u_j)$
- **In terms of Lagrangian multipliers**

$$\sum_i \alpha_i y_i k(x_i, u_j)$$

- **Computational Cost : O($\mathrm{n}_{sv} d$)**
  - n_{sv}: Number of support vectors
  - d : Dimensionality of each data instance

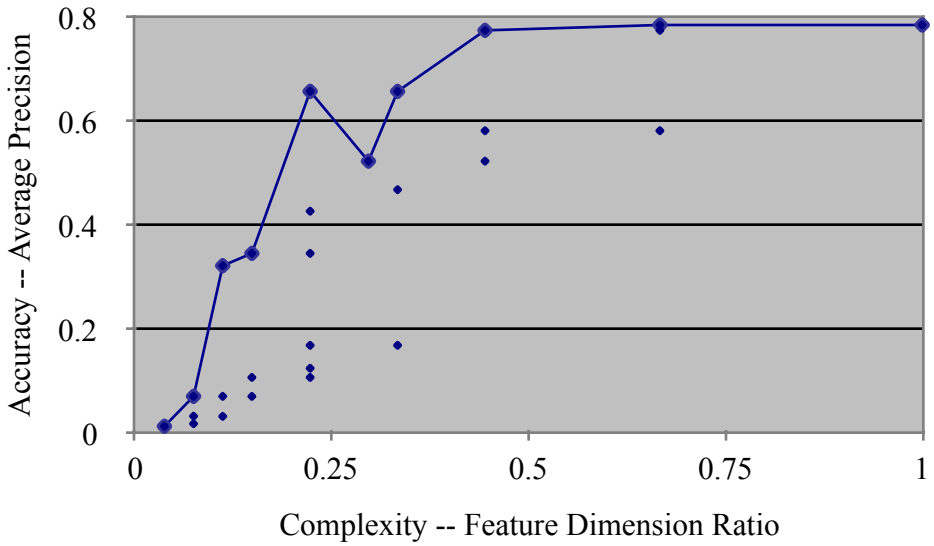
Problems

- **Number of support vectors grows quasi-linearly with size of training set [Tipping 2000]**
- **Inner product with each support vector of dimensionality d expensive**
  - Example TREC2003
    - Human : 19745 support vectors
    - Face : 18090
- **High data rates(10Gbits/sec) means large number of abandoned data**

Example

- Processing Power 1 Ghz
- 10000 support vectors
- 1000 / 2 features per instance
- Order of at least 10^7 operations required per stream per sec
- Translates to less than 100 instances evaluated per sec with only one classifier

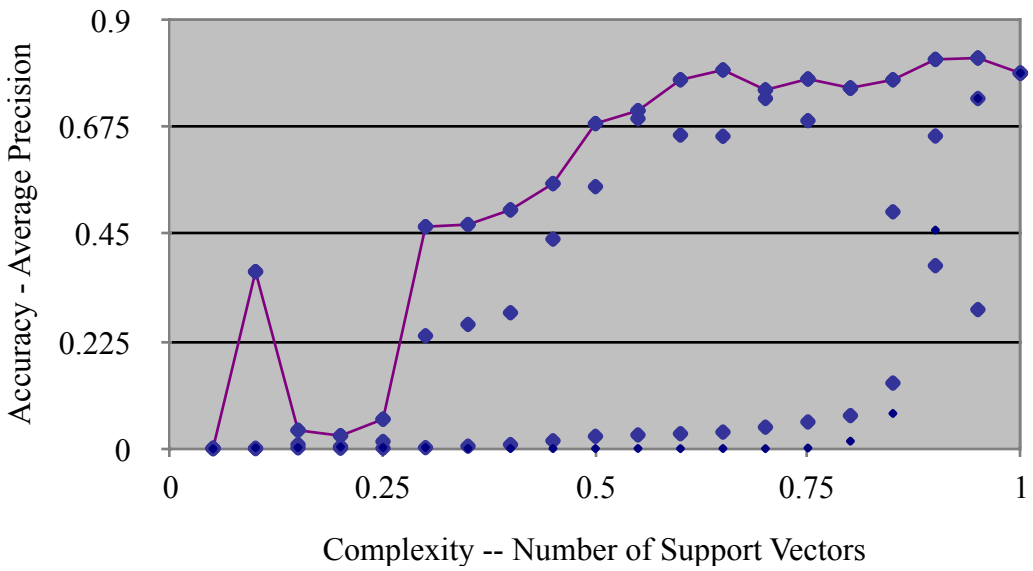
Naïve Approach I – Feature Dimension Reduction



- Experimental Results for Weather_News Detector
- Model Selection based on the Model Validation Set
- E.g., for Feature Dimension Ratio 0.22, (the best selection of features are: 3 slices, 1 color, 2 texture selections), the accuracy is decreased by 17%.

Slice	Color	Texture	Feature Dimension Ratio	AP
3	3	3	1	0.7861
3	3	2	0.666666667	0.7861
3	2	3	0.666666667	0.7757
2	3	3	0.666666667	0.5822
3	2	2	0.444444444	0.7757
2	3	2	0.444444444	0.5822
2	2	3	0.444444444	0.5235
3	3	1	0.333333333	0.4685
3	1	3	0.333333333	0.6581
1	3	3	0.333333333	0.1684
2	2	2	0.296296296	0.5235
3	2	1	0.222222222	0.427
3	1	2	0.222222222	0.6581
2	3	1	0.222222222	0.1241
2	1	3	0.222222222	0.3457
1	3	2	0.222222222	0.1684
1	2	3	0.222222222	0.1065
2	2	1	0.148148148	0.0699
2	1	2	0.148148148	0.3457
1	2	2	0.148148148	0.1065
3	1	1	0.111111111	0.3219
1	3	1	0.111111111	0.0314
1	1	3	0.111111111	0.07
2	1	1	0.074074074	0.0318

Naïve Approach II – Reduction on the Number of Support



- Proposed Novel Reduction Methods:
  - **Ranked Weighting**
  - **P/N Cost Reduction**
  - **Random Selection**
  - **Support Vector Clustering and Centralization**
- Experimental Results on Weather_News Detectors show that complexity can be at 50% for the cost of 14% decrease on accuracy

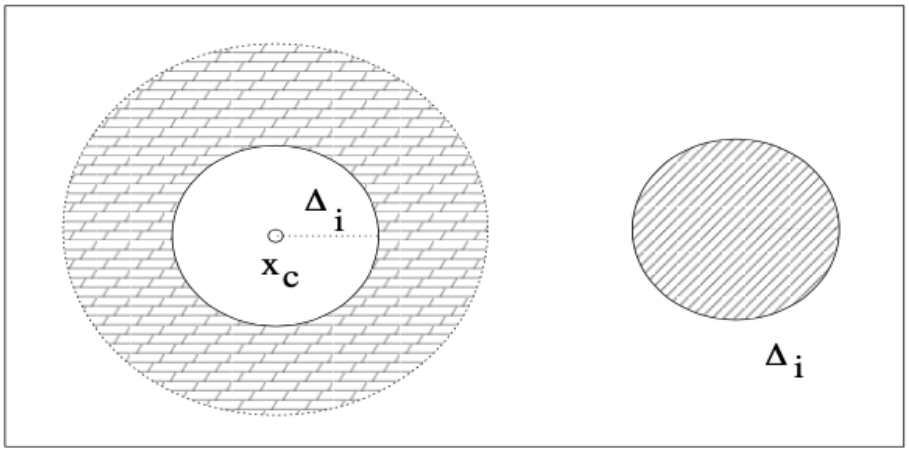
Weighted Clustering Approach

- **Basic steps**
  - Cluster support vectors
  - Use cluster center as representative for all support vectors in cluster
  - Determine scalar weight associated with each cluster center
  - Use only cluster centers to score new instances

Cluster center weight (contd.)

- Choose γ_i minimizing square of difference in scores over all $\pm_i$ and d
- Sub-cases :

$$d \geq \Delta_i \qquad\qquad d < \Delta_i$$



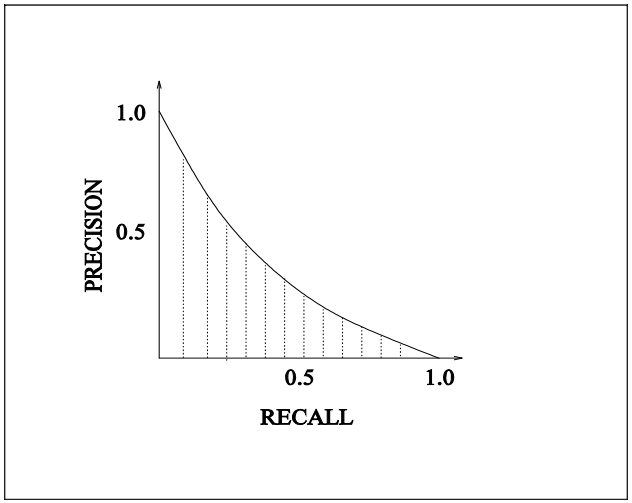
Using the weights

- **For every support vector in cluster**
  - Distance Δ_i known
  - Two weights computed
- **Cumulative effect of all support vectors in clusters additive**
  - Δ_i because of various support vectors added up at center to simulate effect of all support vectors
- Δ_i **sorted, weight arrays rearranged**

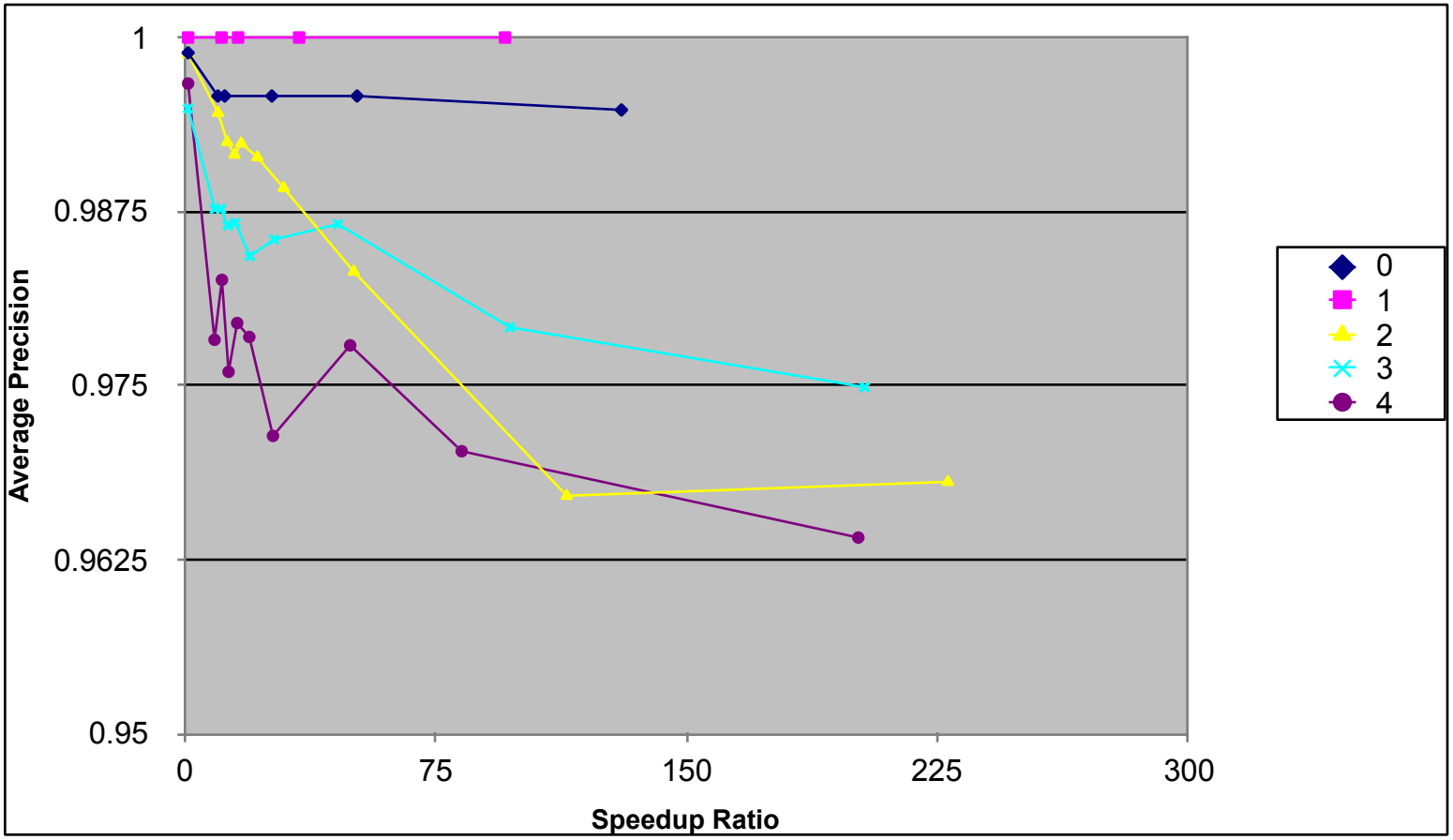
Experiments

- Datasets
  - TREC video datasets (2003 and 2005)
    - 576 features per instance
    - > 20000 test instances overall
  - MNist handwritten digit dataset (RBF kernel)
    - 576 features
    - 60000 training instances, 10000 test instances

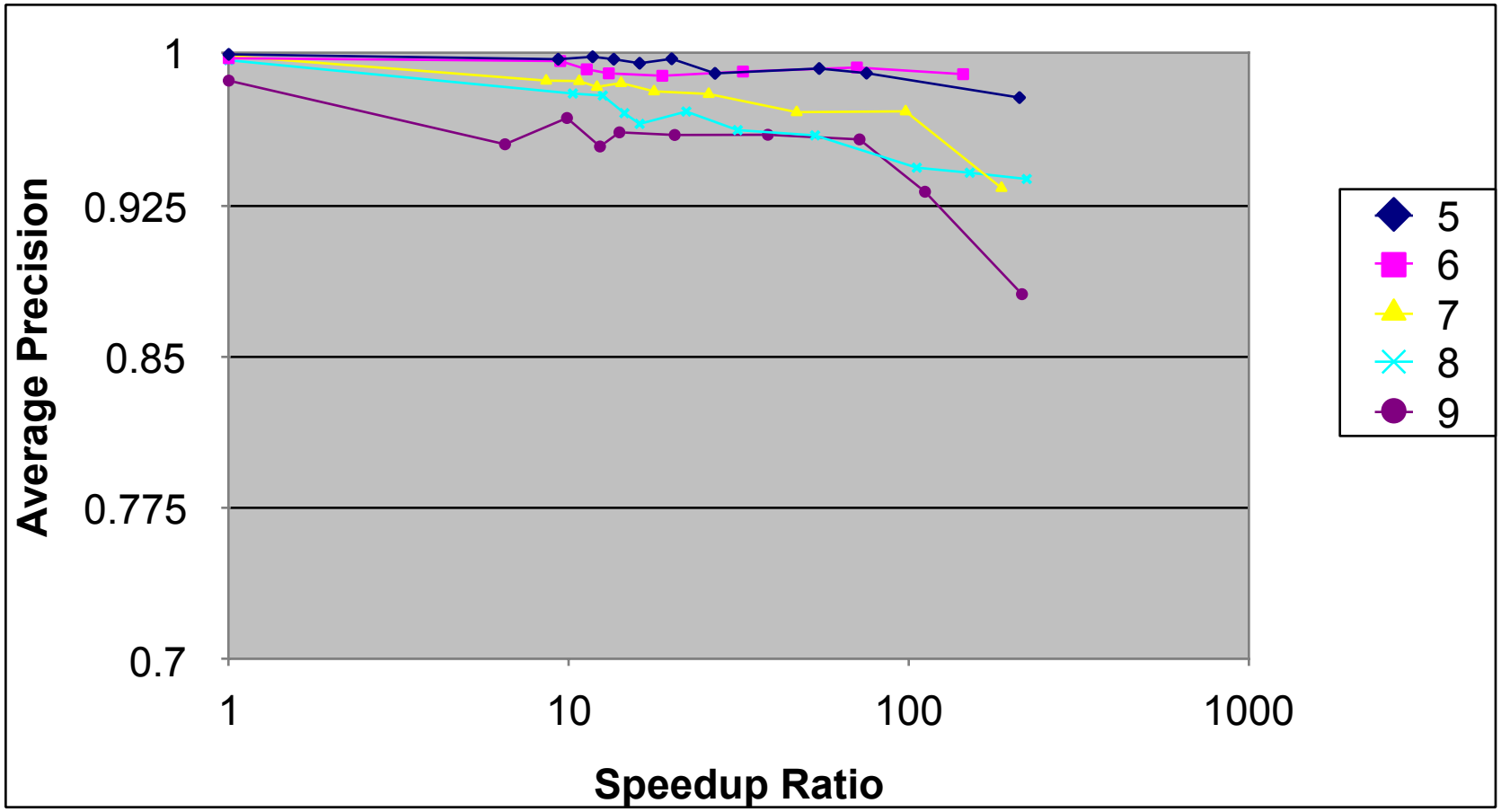
- Performance metrics
  - Speedup achieved over evaluation with all support vectors
  - Average precision achieved



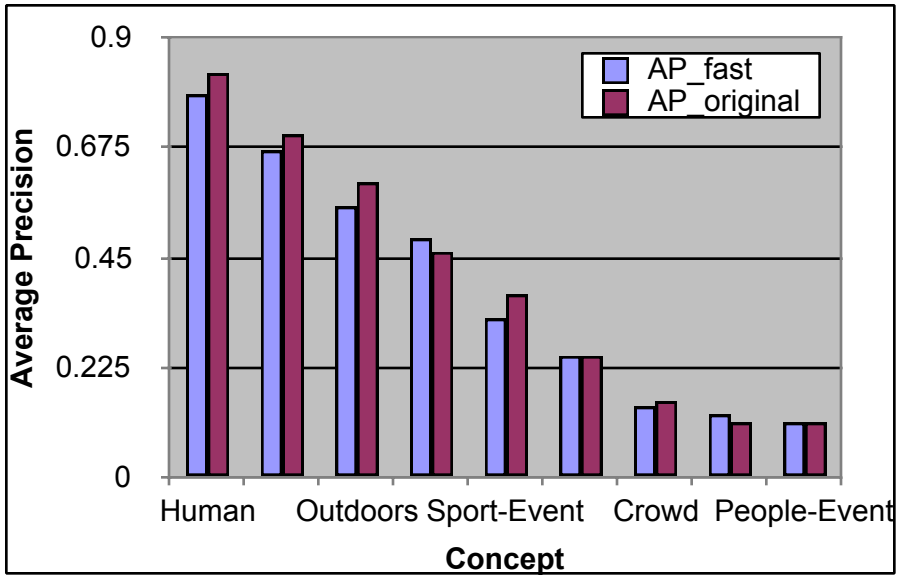
Results (Mnist 0-4)

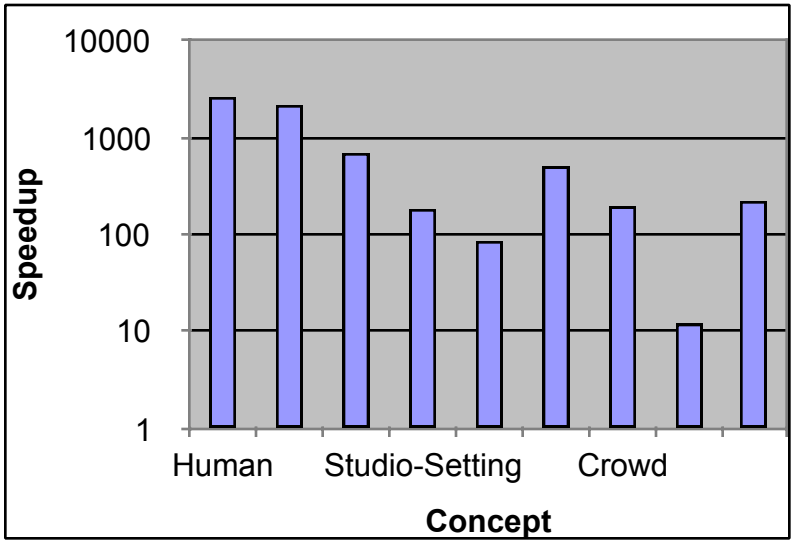


Results (Mnist 5-9)



Results (TREC 2003)





Summary of Complexity Reduction

- Techniques presented demonstrate reasonable performance in terms of both speedup and average precision over multiple concepts in datasets
- Speedups
  - MNist : All concepts at least 50 times faster with AP within 0.04 of original
  - TREC 2003: Eight out of nine concepts speedup greater than 80 times with AP within 0.05 of original
  - TREC 2005: APs in some cases along with speedup respectable
- APs of most concepts close to original APs

Acceleration of Neural Network for Streams

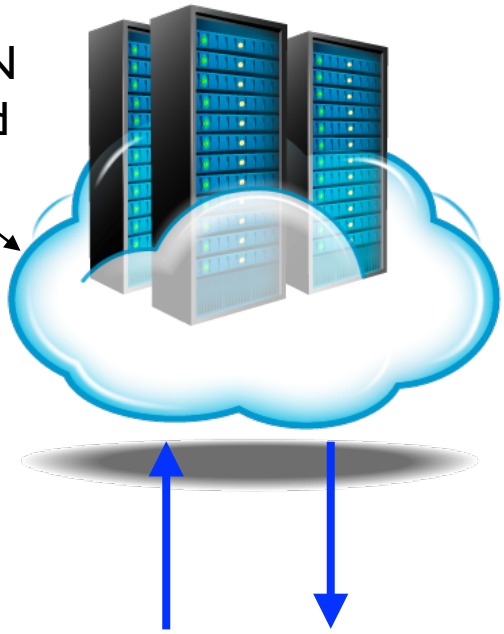
Summary of Acceleration of Neural Network on Mobile Devices

- Porting Deep Convolution Neural Network on iOS Device with near real-time computation
  - Reduce algorithmic complexity with ignorable performance degradation.
  - Utilize computational hardware to achieve better performance.

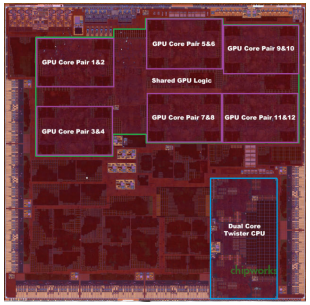
	iPhone 7	iPhone 7+	iPhone 6s	iPad Pro 12.9"
Alex Net	70 ms	70 ms	130 ms	69 ms
p-Alex Net	N/A	35 ms	N/A	28 ms
GoogLeNet	130 ms	128 ms	195 ms	110 ms
p-GoogLeNet	N/A	80 ms	N/A	70 ms
VGG16 Net	880 ms	883 ms	1450 ms	725 ms

Methods for Running CNNs on Mobile Devices

Sending CNN jobs to cloud



Acceleration CNN on Local Device



Apple A9X SoC, 12-core GPU

- How to trade off between algorithmic complexity and performance?
- How to utilize hardware effectively

Challenges for Running CNN on Mobile Devices

	Storage	Memory	Speed
	Model Size	Weights	Mult.s
AlexNet	243MB	61M	725M
VGG-S	393MB	103M	2640M
VGG-16	552MB	138M	15484M
GoogLeNet	51MB	6.9M	1566M

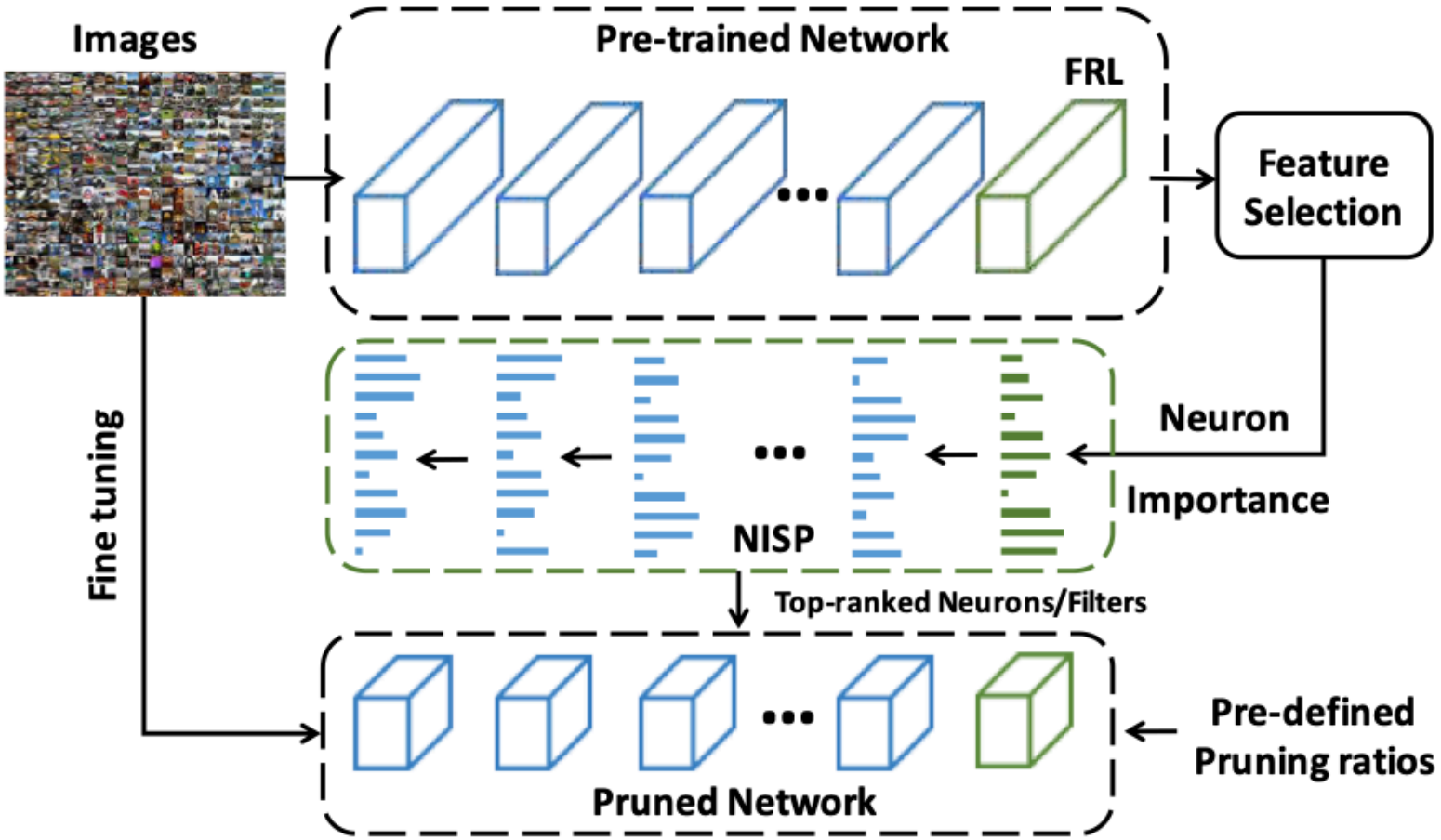
Statistics of some popular CNNS

Reference:
Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications

Computational Resource on iPhone and iPad

	iPhone 6S (Plus)	iPad Air 2	iPad Pro (12.9/9.7)	iPhone 7 (Plus)
SoC	A9	A8X	A9X	A10 Fusion
CPU	2x Twister @ 1.85 GHz	3x Typhone @ 1.5 GHz	2x Twister @ 2.26 GHz	4-core
GPU	PVR GT7600 (6 cluster)	PVR GXA6850 (8 cluster)	PVR 12 Cluster Series 7	6 cluster GPU?
RAM (shared memory)	2GB LDDR4	2GB LDDR3	4GB LDDR4	3GB on Plus?
Memory bus width	64-bit	128-bit	128-bit	?
Max # of threads per group	512	512	512	?

Neuron Importance Score Propagation (NISP, Yu et al 2018)



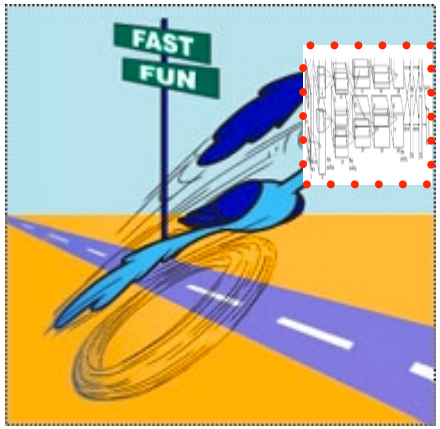
Methods for Running CNNs on Mobile Devices

Sending CNN jobs to cloud

Compression (pruning) of CNN

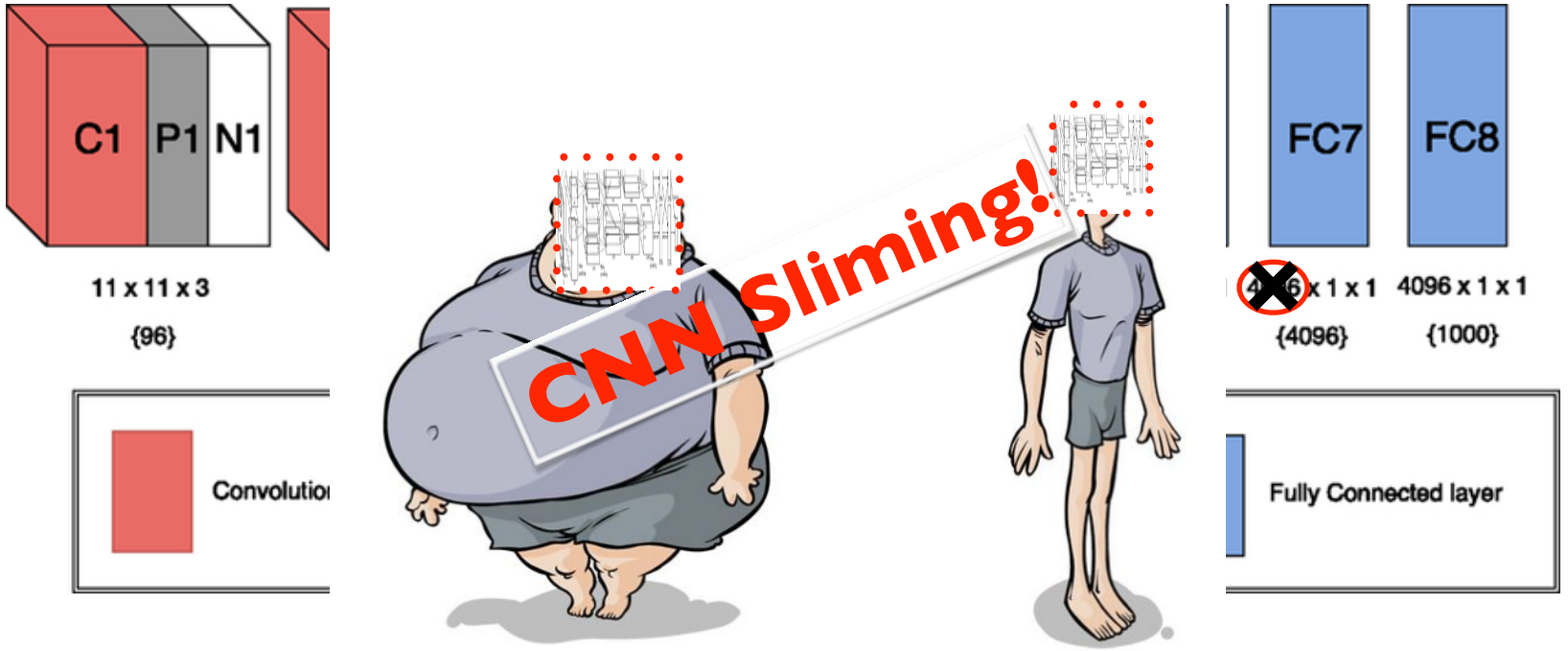
Speeding up CNN

+



Thinking Differently

- All existing methods can be viewed as approximations of an overly-redundant CNN, but do we really need such a CNN as the starting

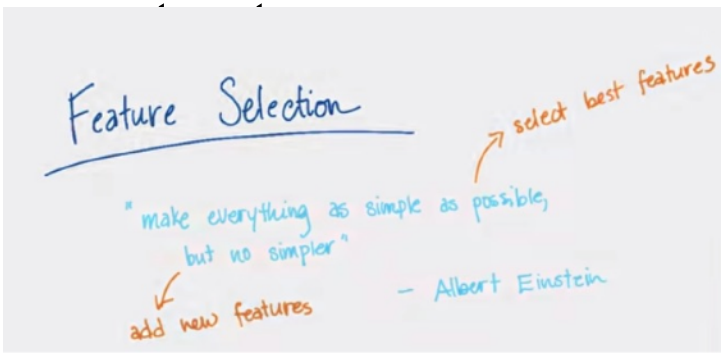


Slim CNN

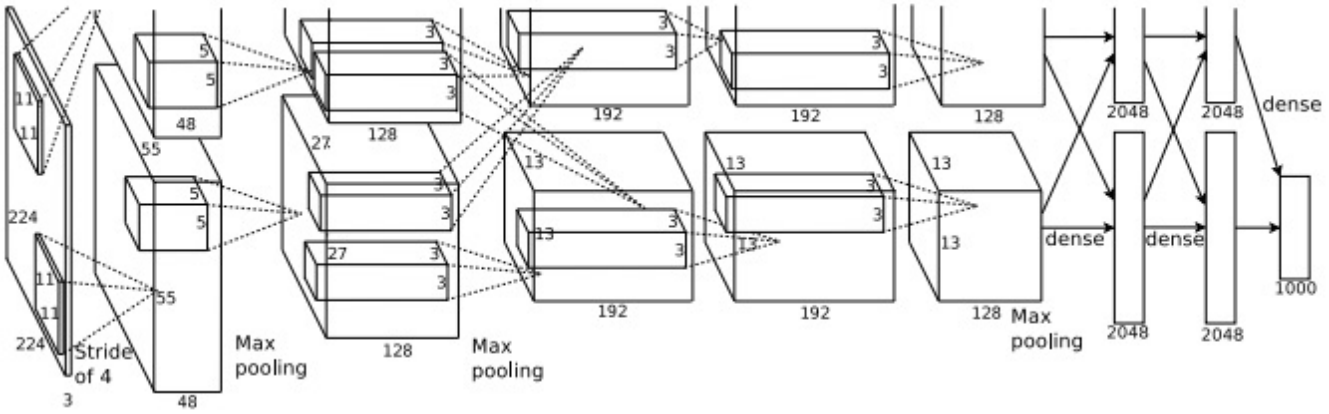
- Slim CNN leads to:
  - less storage space
  - less memory usage
  - less computation
  - less power consumption

Feature Selection on CNN

- CNNs can be viewed as a set of "overly-redundant" feature

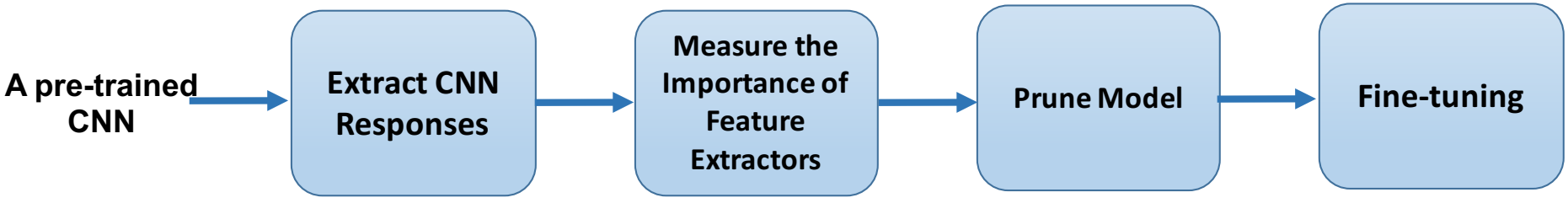


features



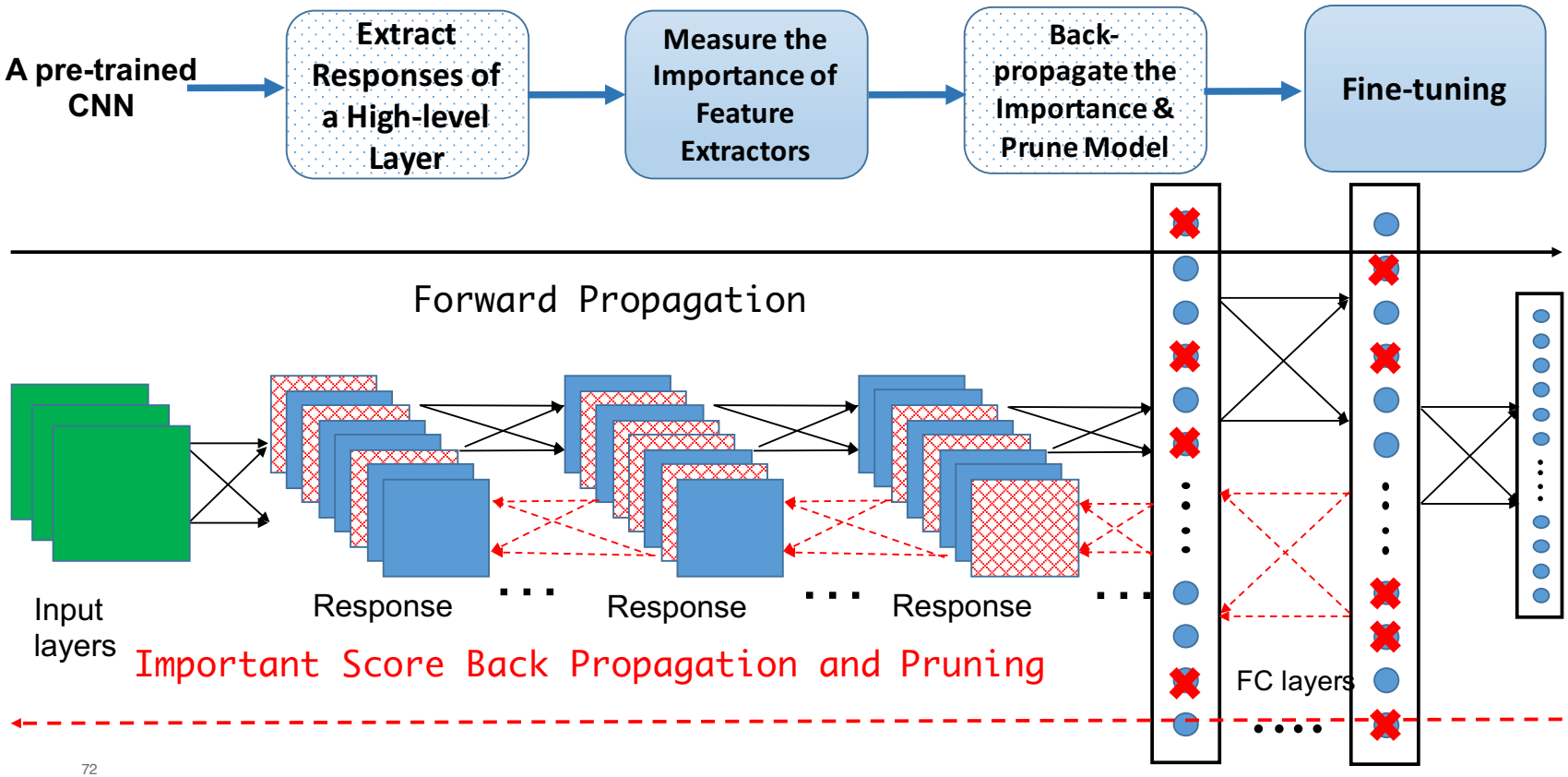
A method for Pruning Redundant Neurons and Kernels of

Apply thermal



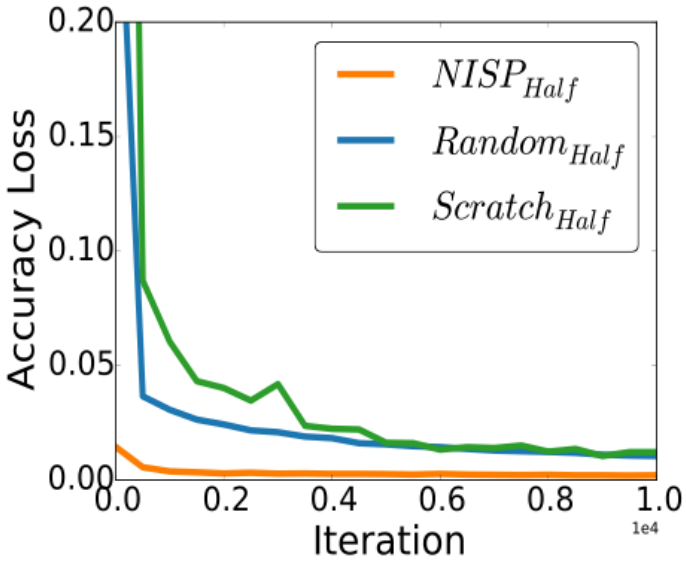
A method for Pruning Redundant Neurons and Kernels of Deep Convolutional Neural Networks (NISP)

- Intractable ⟶ tractable
- Inconsistent ⟶ consistent



Fine-tuning the Pruned Model

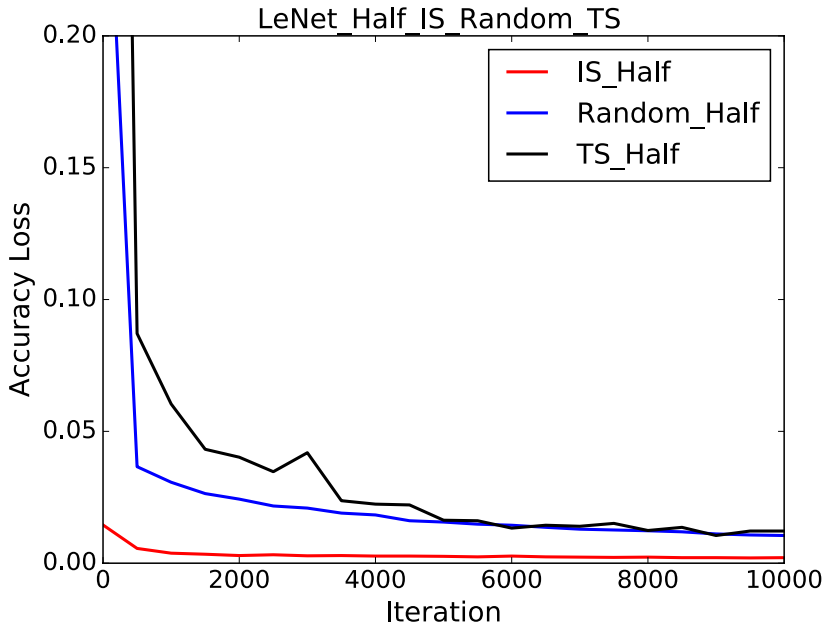
- Our method outperforms the baselines in three aspects
  - Very small accuracy loss at the beginning ==> retains the most important neurons
  - Converges much faster than baselines
  - For LeNet on MIST, our method only decreases 0.02% top-1 accuracy with a running ratio of 50% as compared to the pre-pruned network.

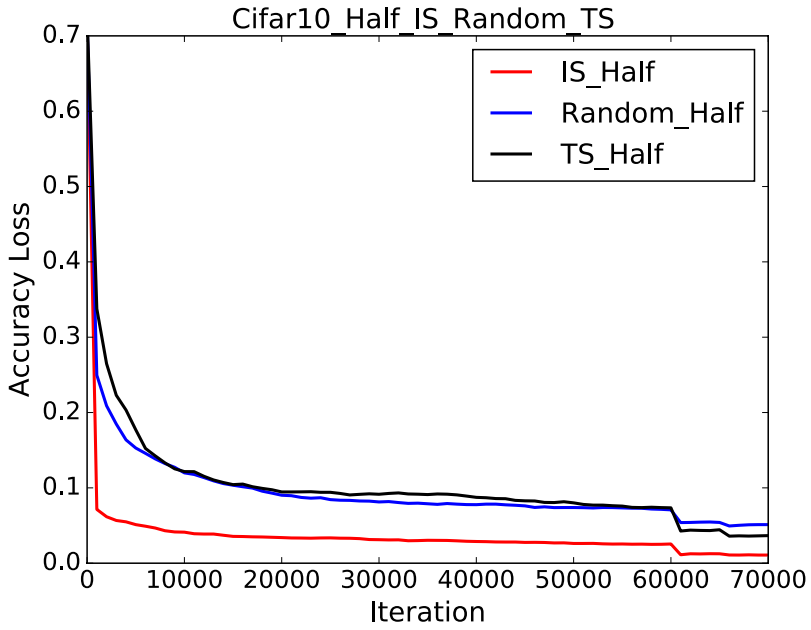


(a) MNIST

Fine-tuning the Pruned Model

- The pruned model consists of important feature extractors, but will suffer loss of accuracy due to loss of redundant features
  - Good starting point on the learning curve due to feature selection
  - Fine-tuning the pruned model with a lower learning rate to recover the performance



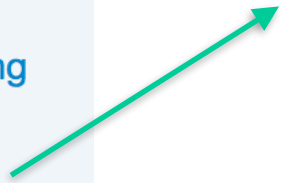


Spark ML Classification and Regression

Columbia University

MLlib: Main Guide

- Basic statistics
- Pipelines
- Extracting, transforming and selecting features
- Classification and Regression
- Clustering
- Collaborative filtering
- Frequent Pattern Mining
- Model selection and tuning
- Advanced topics

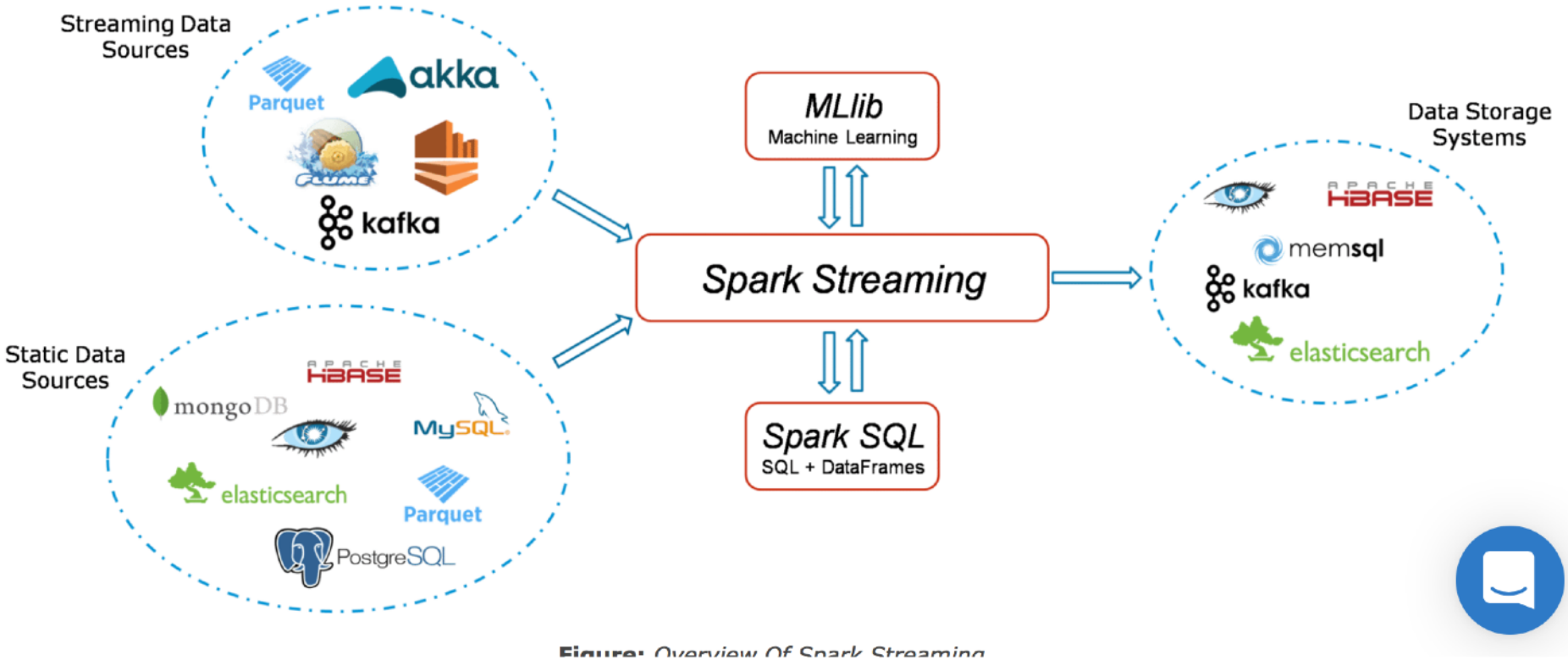


- Classification
  - Logistic regression
    - Binomial logistic regression
    - Multinomial logistic regression
  - Decision tree classifier
  - Random forest classifier
  - Gradient-boosted tree classifier
  - Multilayer perceptron classifier
  - Linear Support Vector Machine
  - One-vs-Rest classifier (a.k.a. One-vs-All)
  - Naive Bayes
- Regression
  - Linear regression
  - Generalized linear regression
    - Available families
  - Decision tree regression
  - Random forest regression
  - Gradient-boosted tree regression
  - Survival regression
  - Isotonic regression

Spark Streaming

Spark Streaming

Spark Streaming



https://www.edureka.co/blog/spark-streaming/

Spark Streaming

- Basic Concepts
  - Linking
  - Initializing StreamingContext
  - Discretized Streams (DStreams)
  - Input DStreams and Receivers
  - Transformations on DStreams
  - Output Operations on DStreams
  - DataFrame and SQL Operations
  - MLlib Operations
  - Caching / Persistence
  - Checkpointing
  - Accumulators, Broadcast Variables, and Checkpoints
  - Deploying Applications
  - Monitoring Applications
- Performance Tuning
  - Reducing the Batch Processing Times
  - Setting the Right Batch Interval
  - Memory Tuning
- Fault-tolerance Semantics

Spark Streaming Example

COLUMBIA UNIVERSITY

First, we import StreamingContext, which is the main entry point for all streaming functionality. We create a local StreamingContext with two execution threads, and batch interval of 1 second.

```
from pyspark import SparkContext
from pyspark.streaming import StreamingContext

# Create a local StreamingContext with two working thread and batch interval of 1 second
sc = SparkContext("local[2]", "NetworkWordCount")
ssc = StreamingContext(sc, 1)
```

Using this context, we can create a DStream that represents streaming data from a TCP source, specified as hostname (e.g. `localhost`) and port (e.g. `9999`).

```
# Create a DStream that will connect to hostname:port, like localhost:9999
lines = ssc.socketTextStream("localhost", 9999)
```

This `lines` DStream represents the stream of data that will be received from the data server. Each record in this DStream is a line of text. Next, we want to split the lines by space into words.

```
# Split each line into words
words = lines.flatMap(lambda line: line.split(" "))
```

```
# Count each word in each batch
pairs = words.map(lambda word: (word, 1))
wordCounts = pairs.reduceByKey(lambda x, y: x + y)

# Print the first ten elements of each RDD generated in this DStream to the console
wordCounts.pprint()
```

Spark Streaming Example

```
$ ./bin/spark-submit examples/src/main/python/streaming/network_wordcount.py localhost 9999
```

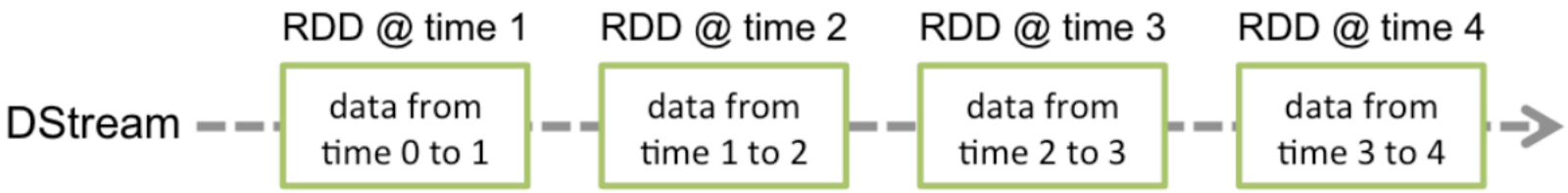
Then, any lines typed in the terminal running the netcat server will be counted and printed on screen every second. It will look something like the following.

Scala Java **Python**

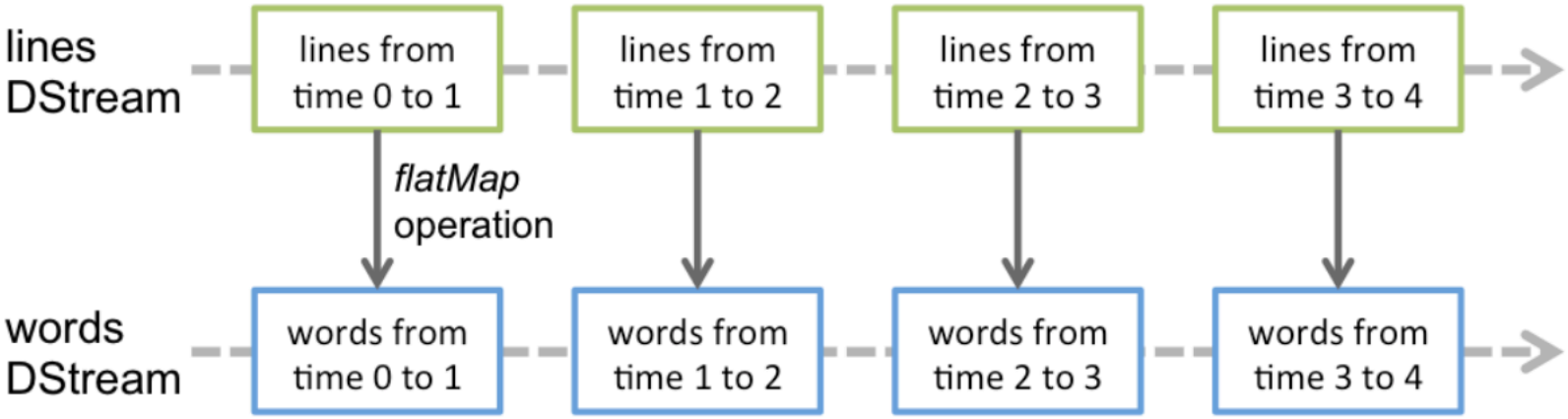
```
# TERMINAL 1:
# Running Net
cat

$ nc -lk 9999

hello world



...
```

```
# TERMINAL 2: RUNNING network_wordcount.py

$ ./bin/spark-submit examples/src/main/python/streaming/network_wordcount.py local
host 9999
...
-------------------------------------------
Time: 2014-10-14 15:25:21
-------------------------------------------
(hello,1)
(world,1)
...
```

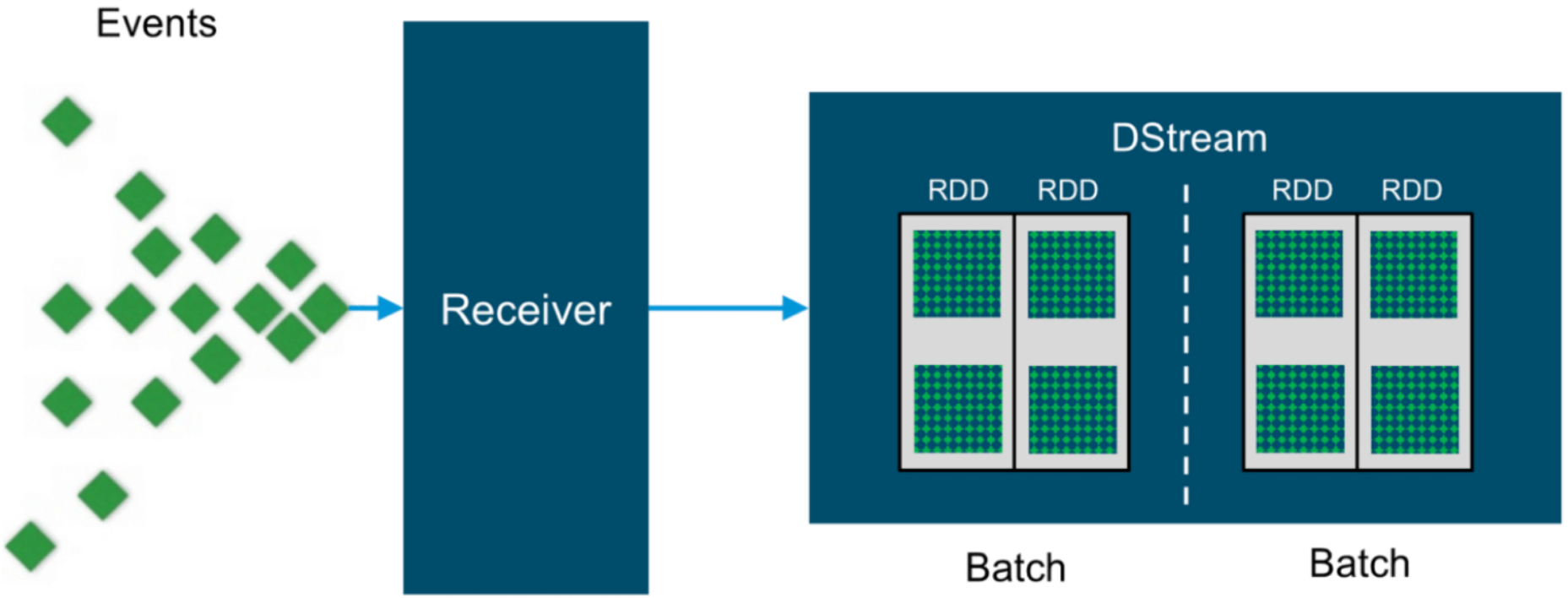
Discretized Streams



Discretized Streams

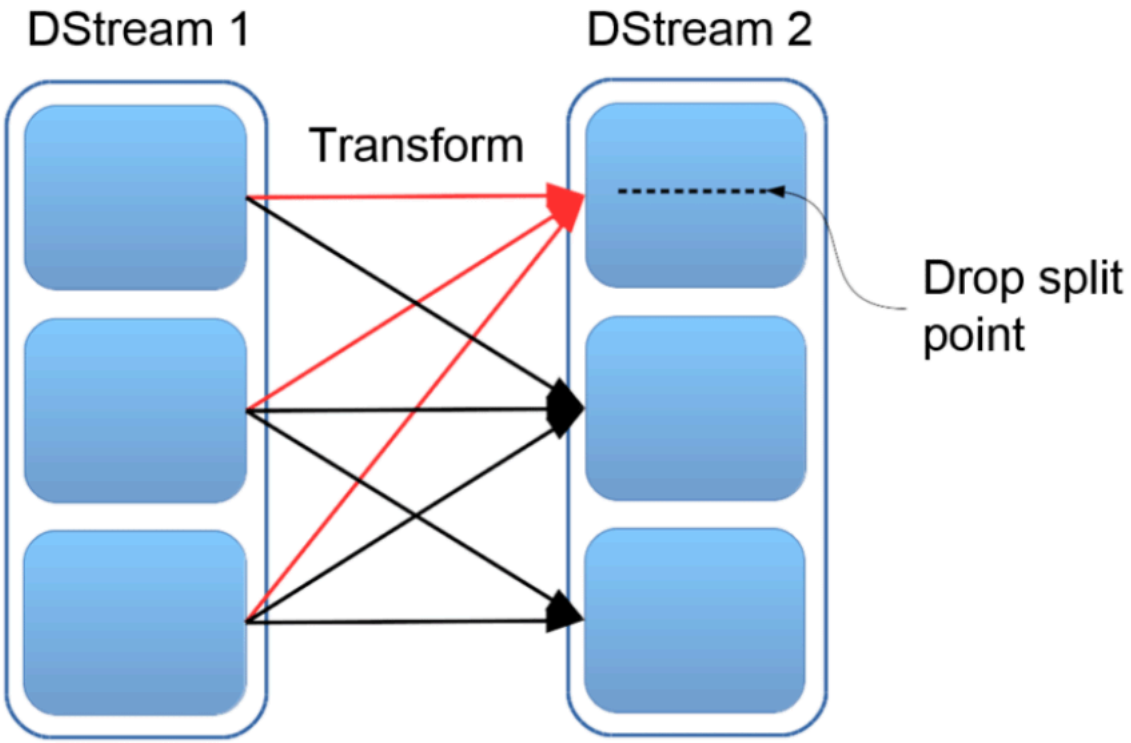


Discretized Streams



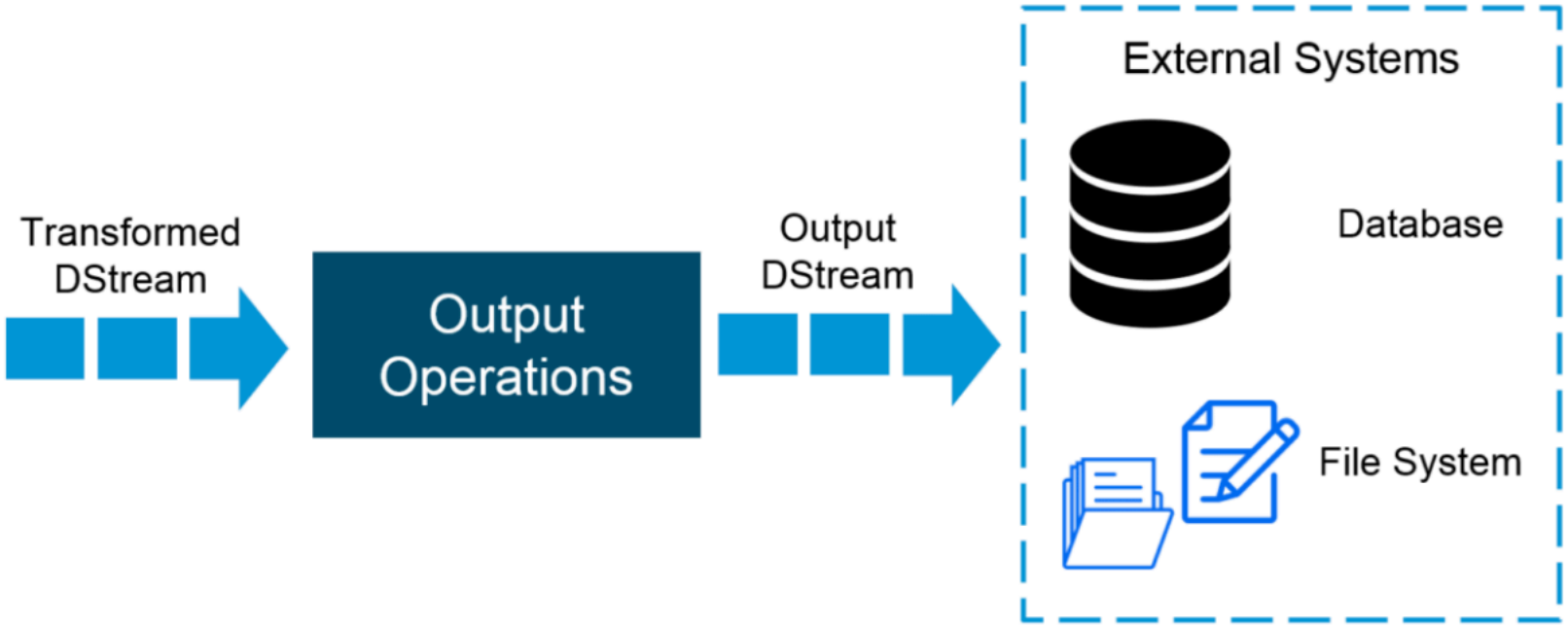
https://www.edureka.co/blog/spark-streaming/

DStream Transforms



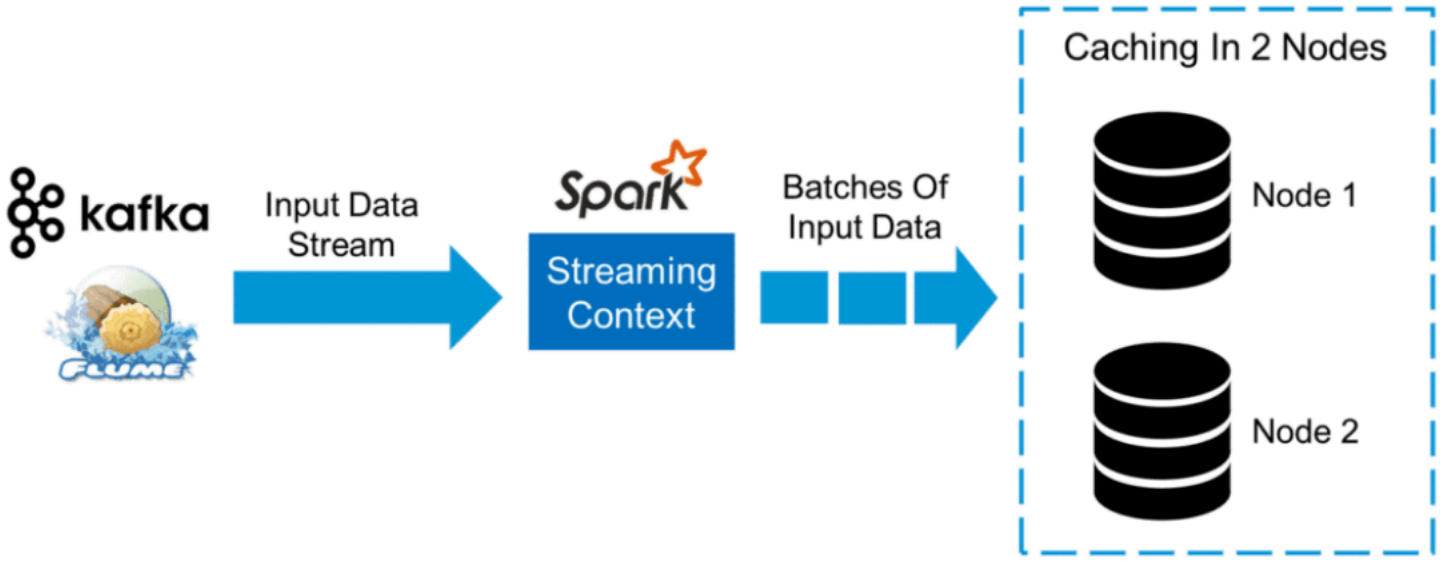
https://www.edureka.co/blog/spark-streaming/

Output DStreams



https://www.edureka.co/blog/spark-streaming/

DStreams Caching



https://www.edureka.co/blog/spark-streaming/

DStreams Example — Twitter Sentiment Analysis

```
//Import the necessary packages into the Spark Program
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.spark.SparkContext._
...
import java.io.File

object twitterSentiment {

def main(args: Array[String]) {
if (args.length < 4) {
System.err.println("Usage: TwitterPopularTags <consumer key> <consumer secret> " + "<access token> <access token secret>
System.exit(1)
}

StreamingExamples.setStreamingLogLevels()
//Passing our Twitter keys and tokens as arguments for authorization
val Array(consumerKey, consumerSecret, accessToken, accessTokenSecret) = args.take(4)
val filters = args.takeRight(args.length - 4)

// Set the system properties so that Twitter4j library used by twitter stream
// Use them to generate OAuth credentials
System.setProperty("twitter4j.oauth.consumerKey", consumerKey)
...
System.setProperty("twitter4j.oauth.accessTokenSecret", accessTokenSecret)

val sparkConf = new SparkConf().setAppName("twitterSentiment").setMaster("local[2]")
val ssc = new Streaming Context
val stream = TwitterUtils.createStream(ssc, None, filters)
```

https://www.edureka.co/blog/spark-streaming/

DStreams Example — Twitter Sentiment Analysis

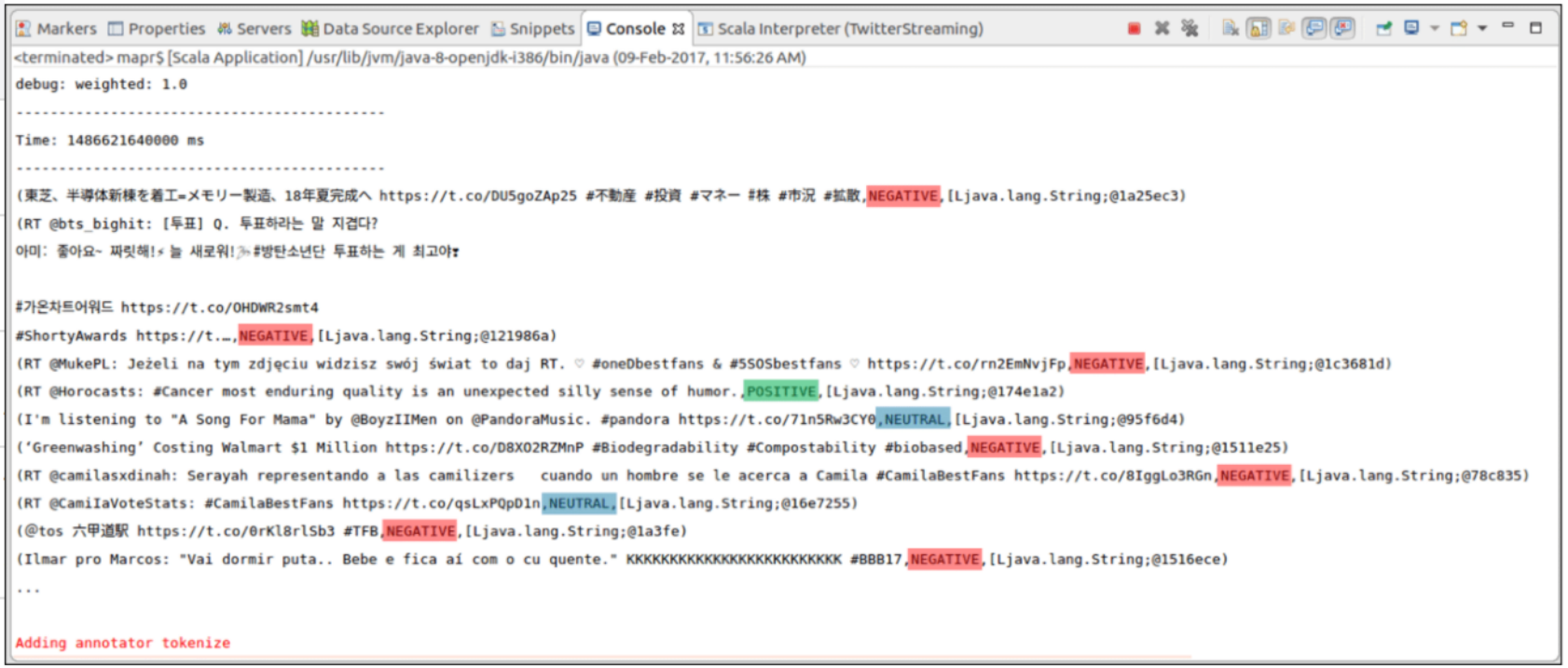
```
//Input DStream transformation using flatMap
val tags = stream.flatMap { status => Get Text From The Hashtags }

//RDD transformation using sortBy and then map function
tags.countByValue()
.foreachRDD { rdd =>
val now = Get current time of each Tweet
rdd
.sortBy(_._2)
.map(x => (x, now))
//Saving our output at ~/twitter/ directory
.saveAsTextFile(s"~/twitter/$now")
}

//DStream transformation using filter and map functions
val tweets = stream.filter {t =>
val tags = t. Split On Spaces .filter(_.startsWith("#")). Convert To Lower Case
tags.exists { x => true }
}

val data = tweets.map { status =>
val sentiment = SentimentAnalysisUtils.detectSentiment(status.getText)
val tagss = status.getHashtagEntities.map(_.getText.toLowerCase)
(status.getText, sentiment.toString, tagss.toString())
}

data.print()
//Saving our output at ~/ with filenames starting like twitters
data.saveAsTextFiles("~/twitters","20000")

ssc.start()
ssc.awaitTermination()
  }
}
```

https://www.edureka.co/blog/spark-streaming/

DStreams Example — Twitter Sentiment Analysis

Results:

The following are the results that are displayed in the Eclipse IDE while running the Twitter Sentiment Streaming program.



All the tweets are categorized into Positive, Neutral and Negative according to the sentiment of the contents of the tweets

https://www.edureka.co/blog/spark-streaming/

Homework #3 (Due 11/01/2019, 5pm)

Columbia University

See HW3 and Tutorial Slides

Questions?