DESIGN CONSIDERATIONS OF CONTEXT AWARE SYSTEMS
Overview

- Preliminary stuff
- Data Finds Data
- FeedMe
- Something completely different
Who is this guy?
Opening questions…
My Context

- BS in mathematics and computer science from Vanderbilt University
- Currently work for the US Government (Computer Systems Researcher)
- MS in computer science from UMD
- Currently a PhD candidate under Prof. Agrawala
My Interests

- “Big Data” Systems
  - Databases (MaybeSQL)
- “Cloud” scripting languages – Pig, Sawzall, Hive, etc.
- Stream Processing
- Machine Learning
Question #1

What is context?
Question #2

What is relevance?
Question #3

What makes a system context aware?
Data Finds Data

- Written by Jeff Jonas and Lisa Sokol.
- Chapter 7 of “Beautiful Data” from O’Reilly Media
- Concepts:
  - Just-in-Time Discovery
  - An argument against Federation (really, a case for consolidation – **What about cooperative silos?**).
  - “Directories” – we’ll call these (very) active indexes (really a trigger on an index update).
  - The Eight Building Blocks of “data finds data” systems.
The Eight Building Blocks
The Eight Building Blocks
The Eight Building Blocks

Observations

+ Features
+ Classification
The Eight Building Blocks

- Observations
- Features
- Classification
The Eight Building Blocks

Observations

+Features
+Classification

Discover

Repository
The Eight Building Blocks

Observations

+Features
+Classification

Discover

Repository

Make Assertions
The Eight Building Blocks

- Discover
- +Features
- +Classification
- Repository
- Make Assertions
- Invalidate Assertions
- Observations
The Eight Building Blocks

- Observations
- Discover
- Features + Classification
- Repository
- Make Assertions
- Invalidate Assertions
- Accumulate + Persist
The Eight Building Blocks

Observations

Repository

+Features
+Classification

Discover

Invalidate Assertions

Make Assertions

Determine Relevance

Accumulate + Persist
The Eight Building Blocks

- Observations
- Repository
- Features + Classification
- Make Assertions
- Invalidate Assertions
- Determine Relevance
- Notify/Act
- Discover
- Accumulate + Persist
Let’s play “what if?”

- Data model? RDF seems flexible enough – an attributed multi-graph (that link nodes to edges?).
- We can then infer new “facts” from the graph.
- Triples need to fade (link a relevance or “correctness” to an edge).
- On “active indexes” – data will find data via equality of subject, predicate, or object fields. Same mechanism for inference could be a hook point for “acting”.
- No general system like this exists today… hint ;-).
My Theory on Context Aware Systems

Three pillars...

Context

Relevance

Change

Context-Aware

James, those don’t look like pillars at all… (I know)
Context

Two interesting definitions:

- **Dey and Abowd**: Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application.

- **Merriam-Webster**: (1) the parts of a discourse that surround a word or passage and can throw light on its meaning.

Simply put, context is how a quantum of information is related to both static and dynamic knowledge.
Relevance

- With context provides scope on the information graph to filter out unrelated information.

- Relevance changes with context. Imagine a case where a driver is trying to find a restaurant using an in-dash information agent. If the car detects a possible collision this information trumps the current task.
Context is always changing. Change can best be detected and acted upon as information becomes known. Because of this context aware systems should be event driven (versus query driven).

Typical contexts include space and time. Another case that has already been mentioned is purpose.
We’ll get back to this. First, let’s consider our next paper...
FeedMe: A Collaborative Alert Filtering System

- Written by Sen and the “et al” army from IBM (Jonas and Sokol are also IBMers…)

- Concepts:
  - Interruptions are **expensive**, however an ML classifier can be trained against historic ratings to cut down on unwanted notifications.
Three forms of Naïve Bayes:

- Simple – All users mixed, only consider 15 features.
- Personalized – Train NB against individual user preference.
- Collaborative* – Use the global measurements to manage issues with sparse data.

*(Collaborative filtering was all the rage in 2006, the Netflix prize wasn’t cracked until 2009. I suppose at some level 20 people is collaborating...)*
HOWTO: Machine Learning

Step 1: Feature Engineering!
Step 2: Throw every ML algorithm at the wall.
Step 3: Repeat until you win or give up…

- Feature types:
  - Intrinsic (author, subject, source, means)
  - Contextual (time since last alert, # of alerts in last minute)
  - Environmental (user’s current open application – “Angry Birds is open, interrupt”)

Results

- Simple (naïve?) Naïve-Bayes – 64% accuracy
- Personalized (sophisticated?) Naïve-Bayes – 71.8%
- Collaborative (uber?) Naïve-Bayes – 73.4%

Conclusions:

- 73% isn’t good enough for production (missing 1 in 4 important updates).
- CNB didn’t provide much value in this case, but they point out their data set is too small.
- I’m curious if better features would help as well…
Other thoughts...

- Perhaps HCI issues – Google Priority Inbox
- This is an active mechanism, what about passive stuff? Again, Priority Inbox, but what about branching into other aspects of information consumption – say browsing habits based off of small stimuli (reddit, slashdot, Google news).
- What about group dynamics? That is, can this be used in a more structured environment or for incidence response? Murgency?
Aspects of FeedMe architecture?

- The eight building blocks?
  - Events
  - Feature extraction/classification
  - Discovery
  - Assertion
  - Invalidation
  - Memory/Accumulation
  - Relevance
  - Notification

- The three pillars?
  - Context
  - Relevance
  - Change
So what?
A Serious Problem…

- The rate of information production is accelerating.
- We work very hard to keep on top of things, but it’s a losing battle.
- Data can find data all it wants, but I want the relevant data to find me.
- A good computer scientist by their very nature is a certain kind of lazy — we do upfront work so the machine do the rest.
Can we infer context from tweets?

In October 2012, we collected 127 million tweets; of those only 2,238 tweets mentioned traffic with geo-location...
People ask me all the time how we delivered four surplus budgets. What new ideas did we bring? I always give a one-word answer: arithmetic.

Former President Bill Clinton
Speech to the 2012 Democratic National Convention
And Twitter says...
The 2012 Presidential Debates provided an excellent test case for our system. Could we track the real time response to the debates?
Nationmind Architecture

Sample Hose
Directed Filter

Twitter Stream
Client

NationMind
Storm
Topology

Redis
Concept
Lookup +
Tracking

Web app + REST
Server
Naïve Sentiment

- Full sentiment analysis tracks a 5-tuple:
  - \(<who, \ when, \ target, \ aspect, \ sentiment>\)
- Full sentiment is actually very hard in the best of cases (say product reviews).
- Instead of full sentiment, we’re going to take the naïve approach and count the number of positive and negative words associated with a concept.
- We’ll make up for it in the aggregate…
Spitkovsky and Chang of Google and Stanford released a data set that maps anchor text to Wikipedia articles from:

- English Wikipedia Titles
- Anchor texts from English inter-Wikipedia links
- Anchor texts into Wikipedia from non-Wikipedia web pages
Who/What/Where is Romney?

Baseball_glove

... mitt romney ...

Romney,_West_Virginia

13%

35%

53%

22%

98%

Mitt_Romney
## Concepts During the Debate

<table>
<thead>
<tr>
<th>Concept</th>
<th>Concept</th>
<th>Concept</th>
<th>Concept</th>
<th>Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abortion</td>
<td>Diplomacy</td>
<td>Johad</td>
<td>Osama_bin_Laden</td>
<td>Roe_v._Wade</td>
</tr>
<tr>
<td>Afghanistan</td>
<td>Drone</td>
<td>Labor_unions...</td>
<td>Outsourcing</td>
<td>Russia</td>
</tr>
<tr>
<td>Al-Qaeda</td>
<td>Economy</td>
<td>Libertarianism</td>
<td>Pakistan</td>
<td>Ryan</td>
</tr>
<tr>
<td>Arab_Spring</td>
<td>Foreign_policy</td>
<td>Libya</td>
<td>Palestine</td>
<td>Sanctions</td>
</tr>
<tr>
<td>Automotive_industry</td>
<td>Government</td>
<td>Marriage</td>
<td>Patient_Protection_and...</td>
<td>Social_Security</td>
</tr>
<tr>
<td>Bailout</td>
<td>Health Care</td>
<td>Medicare</td>
<td>Planned_Parenthood</td>
<td>Syria</td>
</tr>
<tr>
<td>Barack Obama</td>
<td>Homosexuality</td>
<td>Middle_class</td>
<td>Pre-Emptive_Strike</td>
<td>Tarp</td>
</tr>
<tr>
<td>Biden</td>
<td>Immigration</td>
<td>Middle_East</td>
<td>President</td>
<td>Tax</td>
</tr>
<tr>
<td>China</td>
<td>India</td>
<td>Military</td>
<td>Pro-Choice</td>
<td>Tax_break</td>
</tr>
<tr>
<td>Clinton</td>
<td>Insurance</td>
<td>Mitt_Romney</td>
<td>Pro-Life</td>
<td>Terror</td>
</tr>
<tr>
<td>Colonialism</td>
<td>Iran</td>
<td>Muammar_al-Gaddafi</td>
<td>Public_Broadcasting_Serv...</td>
<td>Trade</td>
</tr>
<tr>
<td>Debate</td>
<td>Iraq</td>
<td>Nation-building</td>
<td>Rape</td>
<td>Tunisia</td>
</tr>
<tr>
<td>Deficit</td>
<td>Israel</td>
<td>NATO</td>
<td>Regime</td>
<td>Unemployment</td>
</tr>
<tr>
<td>Democrat</td>
<td>Japan</td>
<td>Oil</td>
<td>Republican</td>
<td>(UN) + (Women’s Health)</td>
</tr>
</tbody>
</table>
Debate #2

We used a stream graph to show a basic comparative view of the topic hits. For this debate we had a couple of hundred visitors, there were numerous issues caused by poor REST server design. We introduced many improvements to address these issues in later runs. We also improved the stream graph by removing the high volume concepts for the candidates and added a scale.
NationMind: Twitter Analysis

Concept Frequency: Barack Obama

Sentiment

Concept Frequency: Mitt Romney
NationMind: Twitter Analysis

Concept Frequency: Mitt_Romney

Tracking many different concepts at once can be tricky for many reasons. The following StreamGraph provides a means of doing so. The visualization itself is a work in progress, but it should provide a minimally functional...
However…

Image Copyright 2012, Aislin, used with permission.
Where do we stand?

- **Context:**
  - The Wikipedia data provides ability to link tweets with the concepts. However, we’re not tracking concept co-occurrence.

- **Relevance:**
  - At this point it’s explicitly keyed in by the user for the anticipated use/topics.

- **Change:**
  - Can track volume and see when an event is occurring, but have no notion of drill down.
System work

- Currently working on a mechanism for bridging deep historical data with online streaming data.
- Not aiming at as sophisticated a data model as discussed here, but it should provide a stepping stone to allow us to consider more sophisticated things.
- If you’re still awake and we have time, we can discuss.
Flee or Discuss

Thank you for your time!