Detecting Wikipedia Vandalism via Spatio-Temporal Analysis of Revision Metadata

Andrew G. West

June 10, 2010

ONR-MURI Presentation
Where we left off....

FROM THE LAST MURI REVIEW
**Spatio-Temporal Reputation**

- **Single-entity reputation values** are the status quo
  - Issue: Sybil attacks (e.g., spam botnets)

- **Spatial reputation**:
  - No entity-specific data? Use broader groupings
  - Exploit homophily
  - Clarity in borderline classification cases
Hierarchical Groupings = TDG = QTM

- Spatial groupings for spam detection leverage the IP assignment hierarchy
  - Entities are IP addresses
  - \{AS, Subnet, IP\} groups used
- TDGs are hierarchies, thus spatio-(temporal) techniques may fulfill the reputation component of QTM/QuanTM
PreSTA for Spam Detection

PreSTA: Preventative Spatio-Temporal Aggregation

Incoming Emails → Cache DB → PreSTA Client

Bl Source DBs
Spamhaus

Subscription

Blacklist DB

PreSTA Server

Preventive Spatio-Temporal Aggregation

Spatial Analysis
Temporal Analysis
Reputation Engine

Classifier

Cache DB

Cache Hit

SMTP Server

Cache Miss

Decision
New Contributions...

APPLYING SPATIO-TEMPORAL PROPERTIES TO WIKIPEDIA
Vandalism


NLP effective for blatant instances. Subtle ones (e.g., insertion of "not", name replacement) – much harder to find.

Our method: Alternative means of detection, complementing NLP.

VANDALISM: Informally, an edit that is:

- Non-value adding
- Offensive
- Destructive in content removal
Big Idea

• Wikipedia revision metadata (not the article or diff text) can be used to detect instances of vandalism
  – As effective as language-processing [2] efforts
  – Machine-learning over spatio-temporal props:
    • Simple features: Straightforward metadata analysis
    • Aggregate features: Reputation values for single entities (editors, articles) and spatial groupings thereof (geographical location, topical categories)
Outline

• Labeling revisions (rollback)
• Simple features
  – Motivation: SNARE [1] spam-blocking
  – Edit time-of-day, day-of-week, comment length...
• Aggregate features
  – Article rep., editor rep., spatial reputations...
• Classifier performance
• STiki [4] (a real-time implementation)
Metadata

Wikipedia provides metadata via DB-dumps:

<table>
<thead>
<tr>
<th>#</th>
<th>METADATA ITEM</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>Timestamp of edit</td>
<td>In GMT locale</td>
</tr>
<tr>
<td>(2)</td>
<td>Article being edited</td>
<td>Able to deduce namespace from title</td>
</tr>
<tr>
<td>(3)</td>
<td>Editor making edit</td>
<td>May be user-name (if registered editor), or IP address* (if anonymous)</td>
</tr>
<tr>
<td>(4)</td>
<td>Revision <strong>comment</strong></td>
<td>Text field where editor can summarize changes</td>
</tr>
</tbody>
</table>
“Reversion” (i.e., undo)
• Any user can execute:
  • (1) Press button
  • (2) Enter edit summary
  • (3) Confirm reversion

“Rollback” (expedited revert)
• Privileged: ≈4,700 users
  • (1) Press button. Done.
  • Auto-summarization: “Reverted edits by x to last revision by y”

Test-set contains ≈50 million edits:
• (1) only NS0 edits (71% of all edits)
• (2) only edits within last year (2008/11+)
Rollback-based Labels

• Use rollback-based labeling:
  – (1) Find special comment format
  – (2) Verify permissions of editor
  – (3) Backtrack to find offending-edit (OE)
  – All edits not in set \{OE\} are \{Unlabeled\}

• Alternatives: Manual labeling, page-hashing

• Advantages of using rollback:
  – (1) Automated (just parsing)
  – (2) High-confidence (privileged users are trusted)
  – (3) Per-case (vandalism need not be defined)
SIMPLE FEATURES

* Discussion abbreviated to concentrate on aggregate ones
• **Temporal props**: A function of when events occur
• **Spatial props**: Appropriate wherever a size, distance, or membership function can be defined

Motivating work: SNARE [1]
• Spatio-temporal props. **effective in spam-mitigation**
  • Physical distance mail traveled, time-of-day, mail sent, message size (in bytes), AS-membership of sender... (13 in total)

• **Advantages of approach:**
  • NLP-filters **easy to evade**... More difficult for spatio-temporal props.
  • Computationally simpler than NLP
Use IP-geo-location data to determine origin time-zone, adjust UTC timestamp

Vandalism most prevalent during working hours/week: Kids are in school(?)

Fun fact: Vandalism almost twice as prevalent on a Tuesday versus a Sunday
Time-since (TS) ...

- High-edit pages most often vandalized
  - ≈2% of pages have 5+ OEs, yet these pages have 52% of all edits
  - Other work [3] has shown these are also articles most visited

<table>
<thead>
<tr>
<th>TS Article Edited</th>
<th>OE</th>
<th>UnLbl</th>
</tr>
</thead>
<tbody>
<tr>
<td>All edits (median, hrs.)</td>
<td>1.03</td>
<td>9.67</td>
</tr>
<tr>
<td>TS Editor Registration</td>
<td>OE</td>
<td>UnLbl</td>
</tr>
<tr>
<td>Regd., median (days)</td>
<td>0.07</td>
<td>765</td>
</tr>
<tr>
<td>Anon., median (days)</td>
<td>0.01</td>
<td>1.97</td>
</tr>
</tbody>
</table>

- Long-time participants vandalize very little
  - “Registration”: time-stamp of first edit made by user
  - Sybil-attack to abuse benefits?
Misc. Simple Features

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>OE</th>
<th>UnLbl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revision comment (average length in characters)</td>
<td>17.73</td>
<td>41.56</td>
</tr>
<tr>
<td>Anonymous editors (percentage)</td>
<td>85.38%</td>
<td>28.97%</td>
</tr>
<tr>
<td>Bot editors (percentage)</td>
<td>00.46%</td>
<td>09.15%</td>
</tr>
<tr>
<td>Privileged editors (percentage)</td>
<td>00.78%</td>
<td>23.92%</td>
</tr>
</tbody>
</table>

- Revision comment length
  - Vandals leave shorter comments
    (lazy-ness? or just minimizing bandwidth?)

- Privileged editors (and bots)
  - Huge contributors, but rarely vandalize
AGGREGATE FEATURES
CORE IDEA: No entity specific data? Examine spatially-adjacent entities (homophily)

PreSTA [5]: Model for ST-rep:

\[ \text{Rep}(\text{group}) = \sum \frac{\text{time}_\text{decay}(\text{TS}_{\text{vandalism}})}{\text{size}(\text{group})} \]

- Grouping functions (spatial) define memberships
- Observations of misbehavior form feedback — and observations are decayed (temporal)
Example Reputation

<table>
<thead>
<tr>
<th>Time</th>
<th>Behavior</th>
<th>Rep.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{S1}$</td>
<td>Calculate</td>
<td></td>
</tr>
<tr>
<td>$T_{S2}$</td>
<td>User Vandalizes</td>
<td>Calculate</td>
</tr>
<tr>
<td>$T_{S3}$</td>
<td>User Vandalizes</td>
<td>Calculate</td>
</tr>
<tr>
<td>$T_{S4}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T_{S5}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$T_{S6}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Rep. Calculation

No history?
Reputation = 0.0
Completely Innocent!
Example Reputation

Time | Behavior | Rep.
---|---|---
TS₁ |  | Calculate
TS₂ | User Vandalizes |  
User Vandalizes | Calculate
TS₃ |  | Calculate
TS₄ |  | Calculate
TS₅ |  | Calculate
TS₆ |  | Calculate

Rep. Calculation
Example Reputation

Rep. Calculation

One incident in history

Reputation:
\[
decay(TS_3 - TS_2) = 0.95
\]

\(decay()\) returns values on \([0,1]\)
Example Reputation

Time | Behavior | Rep.
--- | --- | ---
TS₁ | User Vandalizes | Calculate
TS₂ | User Vandalizes | Calculate
TS₃ | User Vandalizes | Calculate
TS₄ | User Vandalizes | Calculate
TS₅ | User Vandalizes | Calculate
TS₆

Rep. Calculation
Example Reputation

Two incidents in history

Reputation:

\[ \text{decay}(TS_6 - TS_2) + \text{decay}(TS_6 - TS_5) = 0.50 + 0.95 = 1.45 \]

Values are relative
Rollback as Feedback

Use rollbacks (OEs) as neg. feedbacks for entities

- Key notion: A bad edit is not part of reputation until \( (T_{\text{flag}} > T_{\text{vandalism}}) \). Thus, vandalism must be flagged quickly so reputations are not latent.
  - Fortunately, median time-to-rollback: \( \approx 80 \) seconds
Intuitively some topics are controversial and likely targets for vandalism (or temporally so).

- Trivial spatial grouping (size=1)
- 85% of OEs have non-zero rep (just 45% of random)
• Category = spatial group over articles
• Wiki provides cats. /memberships – use only topical ones
• $\text{size}() = \text{Number of category members}$
• Overlapping grouping
• 97% of OEs have non-zero reputation (85% in article case)
**Editor Reputaion**

- **Problem**: Dedicated editors accumulate OEs, look as bad as attackers (normalize? No)
- **Mediocre performance. Meaningful correlation with other features, however.**

```
Note: Both OE plots are remarkably similar and are collapsed for clarity

<table>
<thead>
<tr>
<th>Series</th>
<th>Graph Symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnLbl-Reg</td>
<td>■</td>
</tr>
<tr>
<td>UnLbl-Anon</td>
<td>▼</td>
</tr>
<tr>
<td>OE-Both</td>
<td>●</td>
</tr>
</tbody>
</table>
```

• Straightforward use of the `rep()` function, one-editor groups
Country Reputation

- Country = spatial grouping over editors
- Geo-location data maps IP → country
- Straightforward: IP resides in one country

<table>
<thead>
<tr>
<th>RANK</th>
<th>COUNTRY</th>
<th>%-OEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Italy</td>
<td>2.85%</td>
</tr>
<tr>
<td>2</td>
<td>France</td>
<td>3.46%</td>
</tr>
<tr>
<td>3</td>
<td>Germany</td>
<td>3.46%</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12</td>
<td>Canada</td>
<td>11.35%</td>
</tr>
<tr>
<td>13</td>
<td>United States</td>
<td>11.63%</td>
</tr>
<tr>
<td>14</td>
<td>Australia</td>
<td>12.08%</td>
</tr>
</tbody>
</table>

CDF of Country Reputation

OE-rate (normalized) for countries with 100k+ edits

good rep. ----> poor rep.

% edits, by series

6/10/2010
CLASSIFICATION & PERFORMANCE
ML Training

- Calc. features for all edits. Normalize onto [0,1]; polarity
- SVM: Support Vector Machine
- **ISSUE**: {Unlabeled} set is just that. Very low cost penalties so no over-compensation.
- Train over prior subset to classify now (100+ edits/sec).

### Review of features used (only IP-editors)

<table>
<thead>
<tr>
<th>#</th>
<th>FEATURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Edit time-of-day</td>
</tr>
<tr>
<td>2</td>
<td>Edit day-of-week</td>
</tr>
<tr>
<td>3</td>
<td>Time-since page edited</td>
</tr>
<tr>
<td>4</td>
<td>Time-since user reg.</td>
</tr>
<tr>
<td>5</td>
<td>Time-since last user OE</td>
</tr>
<tr>
<td>6</td>
<td>Rev. comment length</td>
</tr>
<tr>
<td>7</td>
<td>Article reputation</td>
</tr>
<tr>
<td>8</td>
<td>Category reputation</td>
</tr>
<tr>
<td>9</td>
<td>Editor reputation</td>
</tr>
<tr>
<td>10</td>
<td>Country reputation</td>
</tr>
</tbody>
</table>
• ISSUE: Edits classified as OE but in UnLbl may not be FPs:
  – Manual inspection
  – Raw vs. adjusted
  – Corpus produced*

• Similar performance to NLP-efforts [2]

• Use as an intelligent routing (IR) tool

• Shown steady-state

* http://www.cis.upenn.edu/~westand

Recall: % OEs classified as such

Precision: % of edits classified OE that are actually vandalism
Conclusions

• Showed spatio-temporal properties can locate Wikipedia-vandalism comparably to NLP
  – Complementary; still some advantages:
    • Content/language independent
    • Harder to evade (analysis needed)
    • Faster (100+ edits/sec vs. 5 edits/sec)

• Spatio-temporal reputation as a general-purpose technique for content-based access control?
  – This work shows it also works for Wikipedia
References


STiki [4]: A real-time, on-Wikipedia implementation of the technique
EDIT QUEUE: Connection between server and client side

- Populated: Priority insertion based on vandalism score
- Popped: GUI client shows likely vandalism first
- De-queued: Edit removed if another made to same page
Client Demonstration

STiki
Client Demo
• Competition inhibits maximal performance
  – Metric: Hit-rate (% of edits displayed that are vandalism)
  – Offline analysis shows it could be 50%+
  – Competing (often autonomous) tools make it ≈10%

• STiki successes and use-cases
  – Has reverted over 3500+ instances of vandalism
  – May be more appropriate in less patrolled installations
    • Any of Wikipedia’s foreign language editions
    • Corporate Wiki’s and other small installations
  – Embedded vandalism: That escaping initial detection. Median age of STiki revert is 4.25 hours, 200× conventional
Alternative Code Uses

• All code is available [4] and open source (Java)
• Backend (server-side) re-use
  – Large portion of MediaWiki API implemented (bots)
  – Trivial to add new features (including NLP ones)
• Frontend (client-side) re-use
  – Useful whenever edits require human inspection
• Data re-use
  – Corpus building; crowd-sourcing
  – Incorporate vandalism score into more robust tools
Future Direction: Wiki-Spam

• Many people “see” vandalism and do nothing:
  – Becomes “embedded” for days/weeks accumulating views
  – Traffic spikes: During American Idol finale, the “Crystal Bowersox” article was vandalized for just 28 seconds, but 12,000+ viewed the page during this duration.
  – Shows evade-ability, apathy, or both

• What if vandalism was spam?
  – If immature vandalism can get this many views, what about the less detectable and incentivized spam?
  – Could it be more profitable than email spam?
  – What evasion strategies would work best?