STiki: An Anti-Vandalism Tool for Wikipedia using Spatio-Temporal Analysis of Revision Metadata

A.G. West, S. Kannan, and I. Lee

Wikimania `10 – July 10, 2010
STiki = Huggle, but:

**CENTRALIZED**: STiki is always scoring edits, in bot-like fashion.

**QUEUING**: STiki uses 15+ ML-features to set presentation order (not a static rule set)

- **CROWD-SOURCED**: No competition over edits. Greater efficiency
Vandalism detection methodology [6]

Wikipedia revision metadata (not the article or diff text) can be used to detect vandalism

ML over simple features and aggregate reputation values for articles, editors, spatial groups thereof

The STiki software tool

Straightforward application of above technique

Demonstration of the tool and functionality

Alternative uses for the open-source code
Wikipedia provides metadata via dumps/API:

<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>Examine only articles in namespace zero (NS0)</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 comment</td>
<td>Text field where editor can summarize changes</td>
</tr>
</tbody>
</table>
ROLLBACK is used to label edits as vandalism:

Only true-rollback, no software-based ones

Edit summaries used to locate (Native, Huggle, Twinkle, ClueBot)

Bad ones = \{OFF. EDITS\}, others = \{UNLABELED\}

Why rollback?

Automated (v. manual)

High-confidence

Per case (vs. definition)

Why do edits need labels?:

(1) To test features, and train ML

(2) Building block of reputation building
Simple Features

• **Temporal props**: A function of when events occur
• **Spatial props**: Appropriate wherever a size, distance, or membership function can be defined

SIMPLE FEATURES

* Discussion abbreviated to concentrate on aggregate ones
Edit Time, Day-of-Week

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
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

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

<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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TS Editor Registration</th>
<th>OE</th>
<th>UnLbl</th>
</tr>
</thead>
<tbody>
<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>
## Revision comment length

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

- Privileged editors (and bots)
  - Huge contributors, but rarely vandalize

<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>
Aggregate Features

AGGREGATE FEATURES
CORE IDEA: No entity specific data? Examine spatially-adjacent entities (homophily).

Grouping functions (spatial) define memberships

Observations of misbehavior form feedback - and observations are decayed (temporal)

PreSTA [5]: Model for ST-rep:

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

Timestamps (TS) of vandalism incidents by group members
Intuitively some topics are controversial and likely targets for vandalism (or temporally so).

85% of OEs have non-zero rep (just 45% of random)
Category Reputation

- Category = spatial group over articles
- Wiki provides cats. /memberships – use only topical.
- 97% of OEs have non-zero reputation (85% in article case).

<table>
<thead>
<tr>
<th>CATEGORY (with 100+ members)</th>
<th>PGs</th>
<th>OEs/PG</th>
</tr>
</thead>
<tbody>
<tr>
<td>World Music Award Winners</td>
<td>125</td>
<td>162.27</td>
</tr>
<tr>
<td>Characters of Les Miserables</td>
<td>135</td>
<td>146.88</td>
</tr>
<tr>
<td>Former British Colonies</td>
<td>145</td>
<td>141.51</td>
</tr>
</tbody>
</table>

Example of Category Rep. Calculation

- Article: Abraha m Lincoln
- Category: President
  - Barack Obama
  - ... G.W. Bush
- Category: Lawyer
  - ...

Reputation:
- Presidents
- Lawyers
- MAXIMUM( ?)

Feat. Value
• **Problem**: Dedicated editors accumulate OEs, look as bad as attackers (normalize? No)

• Mediocre performance. Meaningful correlation with other features, however.

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

CDF of Country Reputation

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

<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>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>
Off-line Performance

- Similar performance to NLP-efforts [2]
- Use as an intelligent routing (IR) tool

Recall: % total OEs classified correctly
Precision: % of edits classified OE that are vandalism

![Graph showing precision and accuracy metrics.](image)
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
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 5000+ instances of vandalism

May be more appropriate in less patrolled installations

- Any of Wikipedia’s foreign language editions

Embedded vandalism: That escaping initial detection. Median age of STiki revert is 4.25 hours, 200× RBs.

- Further, average STiki revert had 210 views during active duration.
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

Offline inspection tool for corpus building

**Data re-use**

Incorporate vandalism score into more robust tools

Willing to provide data to other researchers
Crowd-sourcing

Shared queue: Pending changes trial

Abuse of “pass” by an edit hoarding user

Do ‘reviewers’ need to be reviewed?

- Where does it stop?
- Multi-layer verification checks to find anomalies
- Could reviewer reputations also be created?

Threshold for queue access?

- Registered? Auto-confirmed? Or more?

Cache-22: Use vs. perceived success

More users = more vandalism found. But deep in queue, vandalism unlikely = User abandonment.


