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

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

WikiSym `10 – July 7, 2010

Vandalism

Barack Hussein Obama II (/bəˈrɑːk huːˈseɪn oʊˈbɑːmə/; born August 4, 1961) is !!! THE WORSTEST PRESIDENT EVER. PLEASE RESIGN IMMEDIATELY!!! the 44th and current President of the United States. He is the first African American to hold the office. Obama previously served as the junior United States Senator from Illinois, from January 2005 until he resigned after his election to the presidency in November 2008.

Originally from Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he was the president of the *Harvard Law Review*. He was a community organizer in Chicago before earning his law degree. He worked as a civil rights attorney in Chicago and taught constitutional law at

Barack Obama

VANDALISM: Informally, an edit that is:

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

- Serious problem. One source [3] estimates hundreds of millions of `damaged page views'
- 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

Outline/Summary

- 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

Metadata

Wikipedia provides metadata via dumps/API:

#	METADATA ITEM	NOTES
(1)	**Timestamp** of edit	In GMT locale
(2)	**Article** being edited	Examine only articles in namespace zero (NS0)
(3)	**Editor** making edit	May be user-name (if registered editor), or IP address (if anonymous)
(4)	Revision **comment**	Text field where editor can summarize changes

Labeling Vandalism

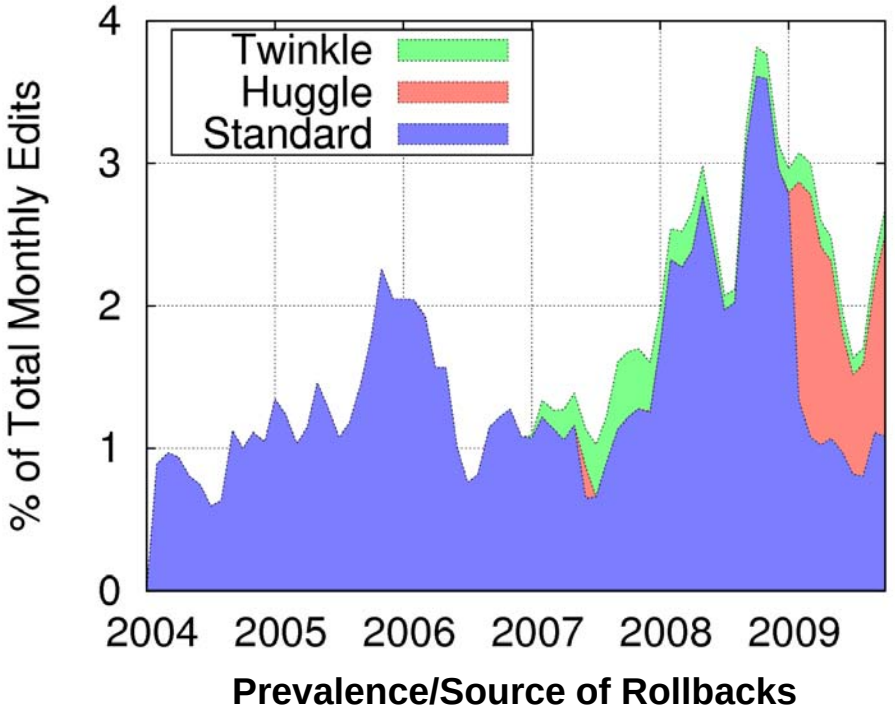
"Reversion" (*i.e.*, undo)

- 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*"

Why do edits need labels?:
- (1) To test features, and train ML
- (2) Building block of reputation building



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

Spatio-Temporal Basics

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

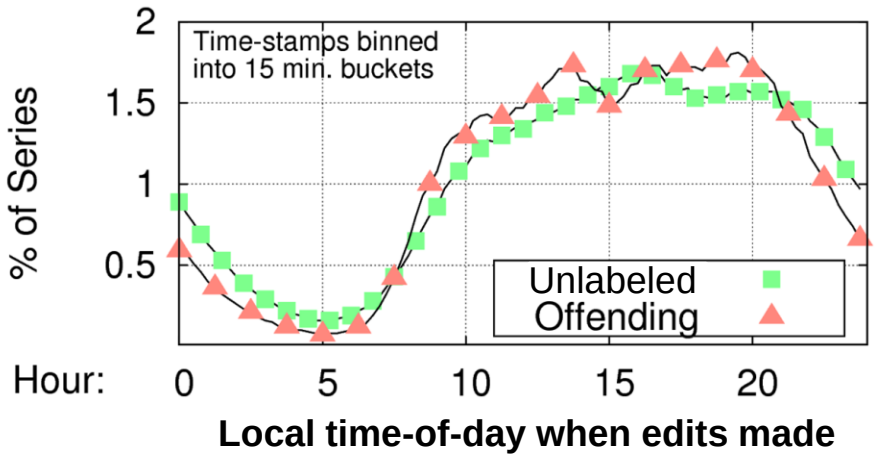
---

Motivating work: SNARE [1]

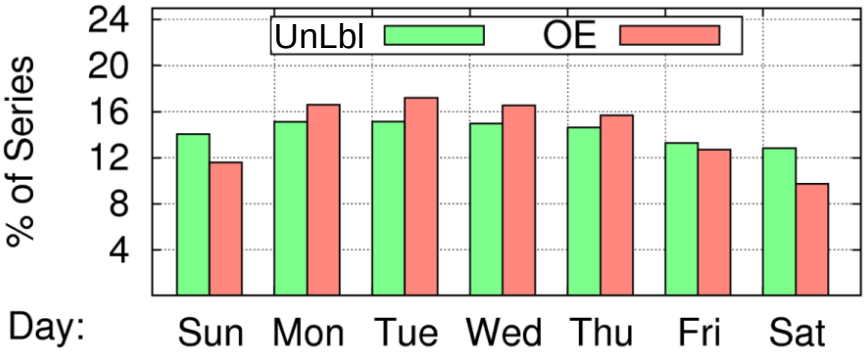
- Spatio-temporal props. effective in spam-mitigation
  - Physical distance mail traveled, time-of-day, mail sent, message size (in bytes), AS-membership of sender… (13 in total)
- Advantages of approach:
  - NLP-filters easy to evade… More difficult for spatio-temporal props.
  - Computationally simpler than NLP

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



Local time-of-day when edits made



Local day-of-week when edits made

Time-Since (TS)…

TS Article Edited	OE	UnLbl
All edits (median, hrs.)	1.03	9.67
TS Editor Registration	**OE**	**UnLbl**
Regd., median (days)	0.07	765
Anon., median (days)	0.01	1.97

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

- 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

Misc. Simple Features

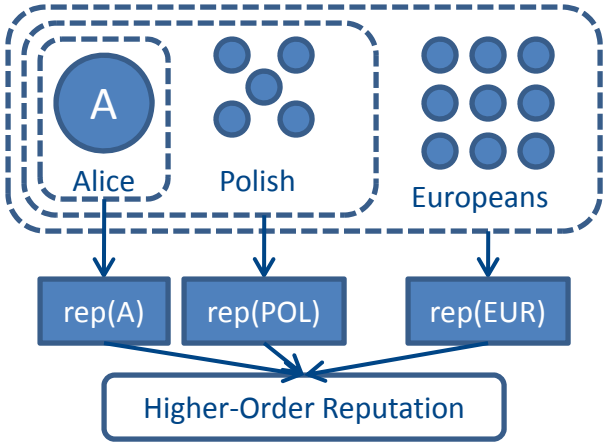
FEATURE	OE	UnLbl
Revision comment (average length in characters)	17.73	41.56
Anonymous editors (percentage)	85.38%	28.97%
Bot editors (percentage)	00.46%	09.15%
Privileged editors (percentage)	00.78%	23.92%

- Revision comment length
  - Vandals leave shorter comments
    (lazy-ness? or just minimizing bandwidth?)
- Privileged editors (and bots)
  - Huge contributors, but rarely vandalize

AGGREGATE FEATURES

PreSTA Algorithm

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



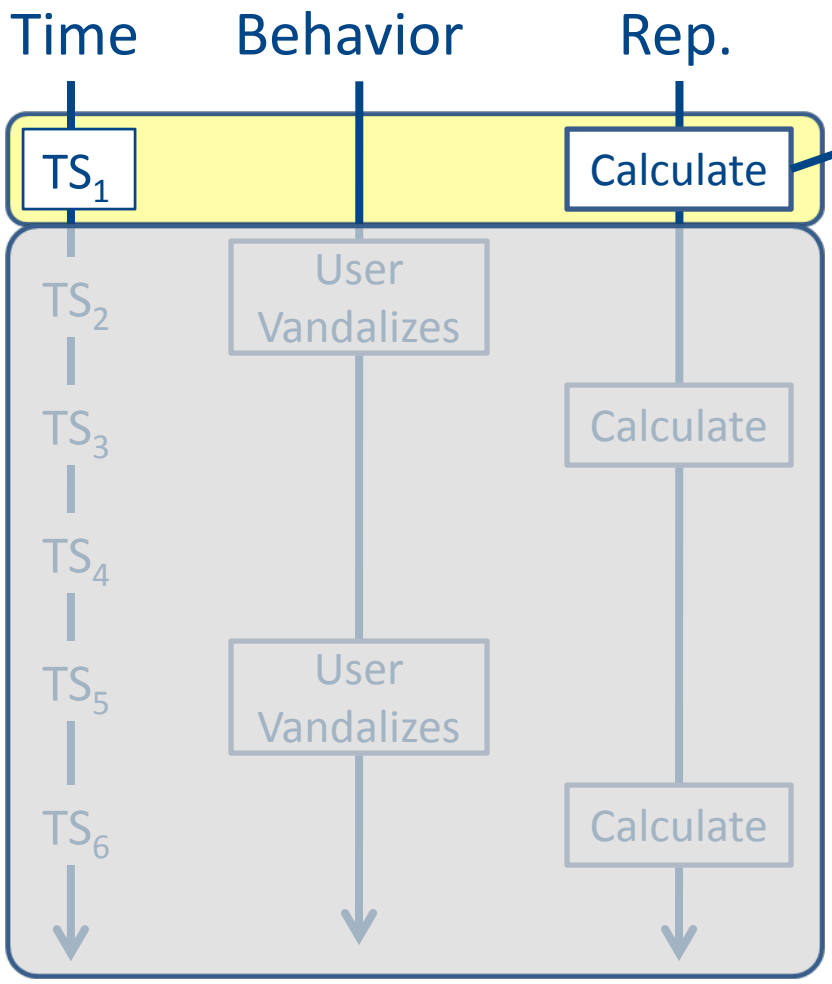
PreSTA [5]: Model for ST-rep:

$$\text{Rep}(group) = \sum \frac{time\_decay\,(TS_{vandalism})}{size(group)}$$

Timestamps (TS) of vandalism incidents by *group* members

- Grouping functions (spatial) define memberships
- Observations of misbehavior form feedback – and observ-ations are decayed (temporal)

Example Reputation



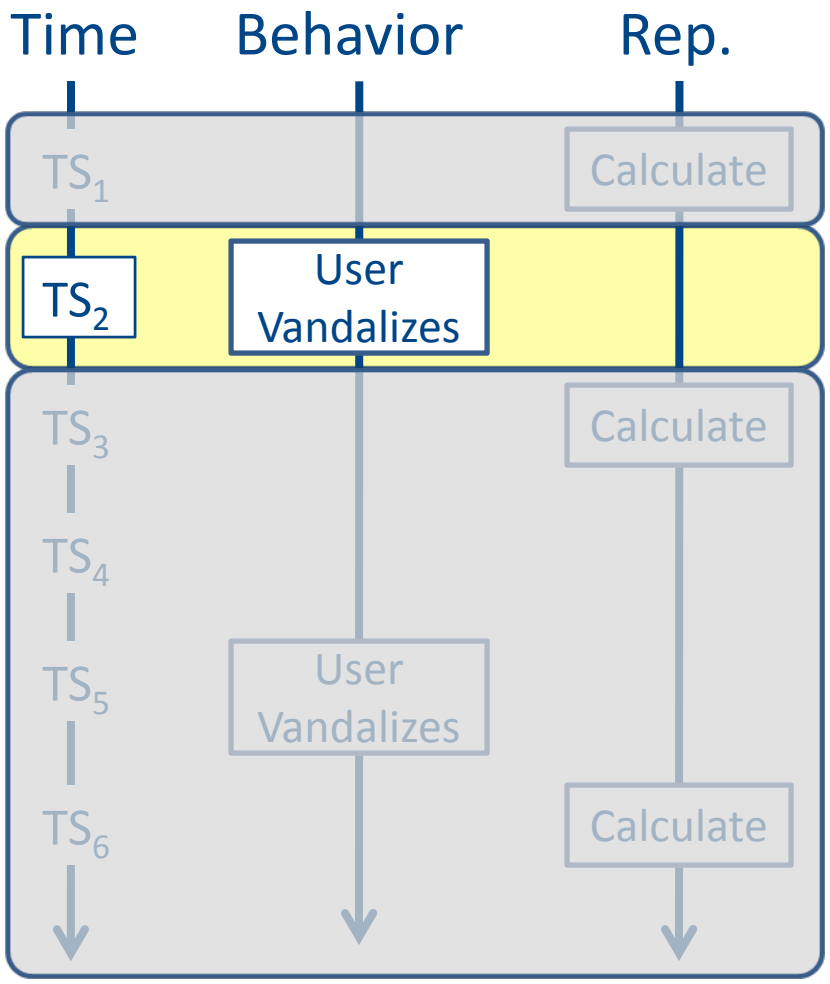
Rep. Calculation

No history?

Reputation = 0.0

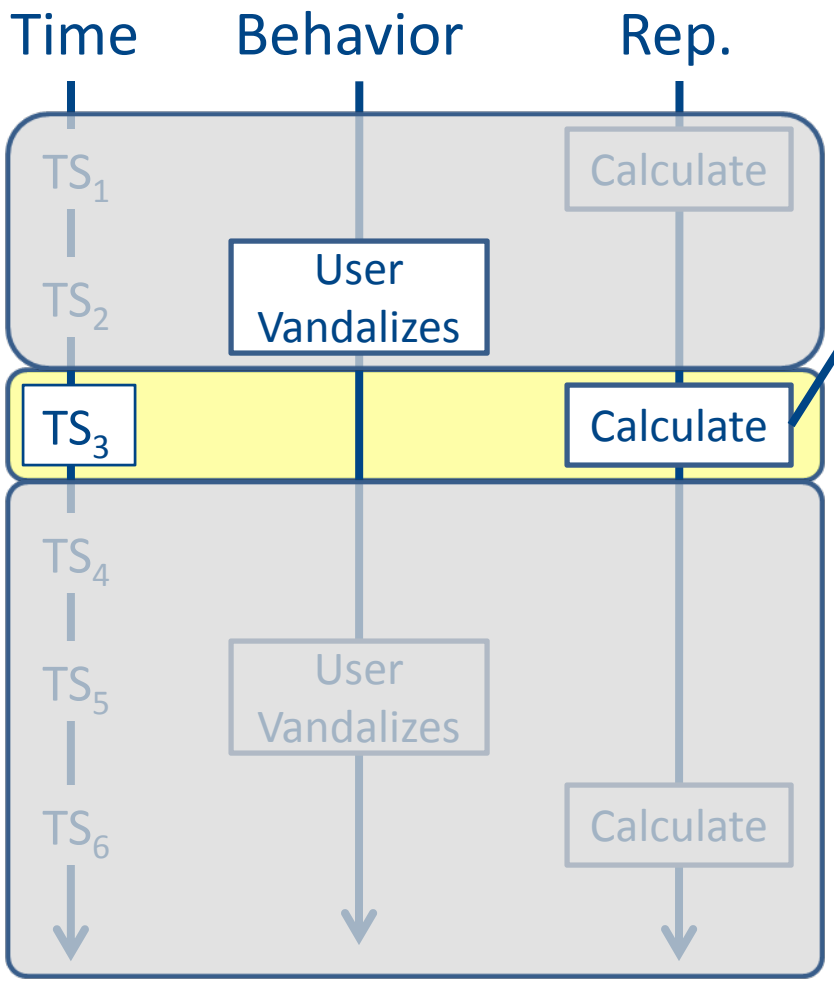
Completely Innocent!

Example Reputation



Rep. Calculation

Example Reputation



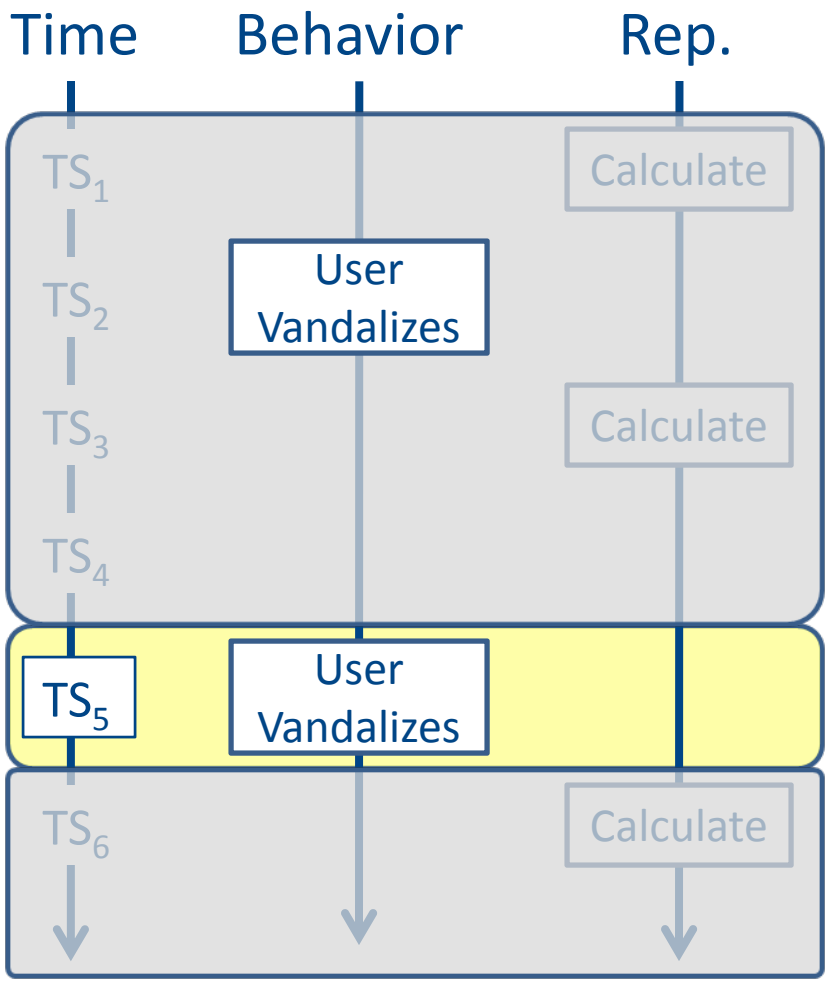
Rep. Calculation

One incident in history

Reputation: *decay*(TS_3 - TS_2) = 0.95

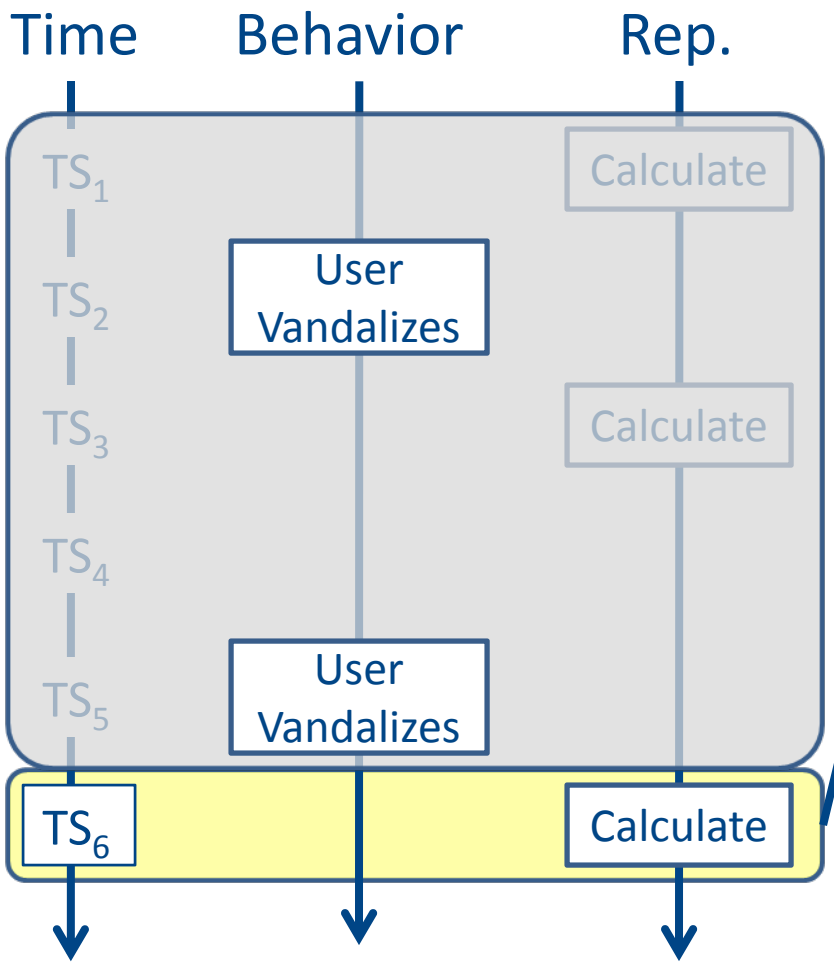
decay() returns values on [0,1]

Example Reputation



Rep. Calculation

Example Reputation



Rep. Calculation

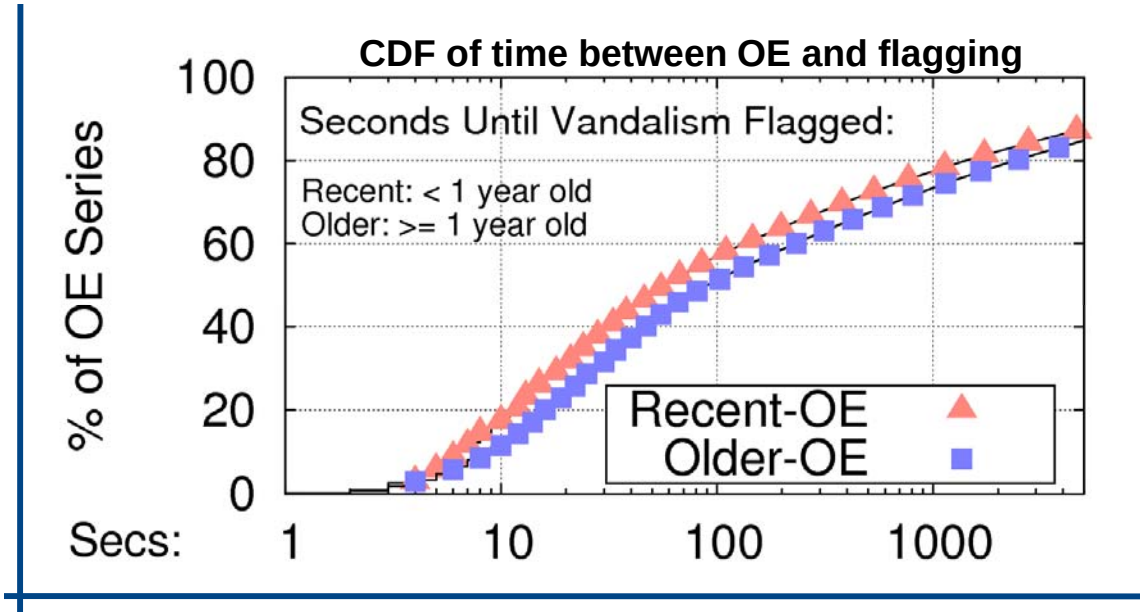
Two incidents in history

Reputation:
$decay(TS_6 - TS_2) + decay(TS_6 - TS_5) = 0.50 + 0.95 = 1.45$

Values are relative

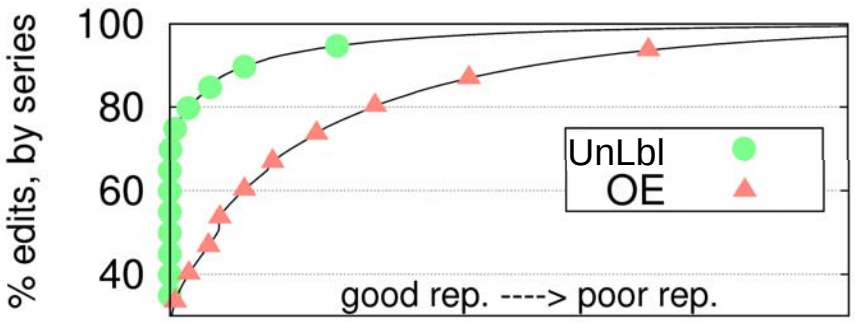
Rollback as Feedback

Use rollbacks (OEs) as neg. feedbacks for entities



- Key notion: A bad edit is not part of reputation until ($TS_{flag} > TS_{vandalism}$). Thus, vandalism must be flagged quickly so reputations are not latent.
  - Fortunately, median time-to-rollback: ≈80 seconds

Article Reputation



CDF of Article Reputation

ARTICLE	#OEs
George W. Bush	6546
Wikipedia	5589
Adolph Hitler	2612
United States	2161
World War II	1886

Articles w/most OEs

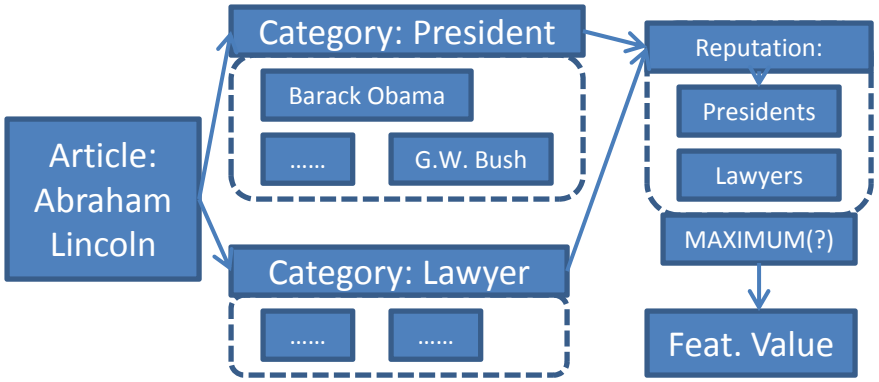
- Intuitively some topics are contro-versial and likely targets for vandalism (or temporally so).
- Trivial spatial grouping (size=1)
- 85% of OEs have non-zero rep (just 45% of random)

Category Reputation

- Category = spatial group over articles
- Wiki provides cats. /memberships – use only topical ones
- *size*() = Number of category members
- Overlapping grouping
- 97% of OEs have non-zero reputation (85% in article case)

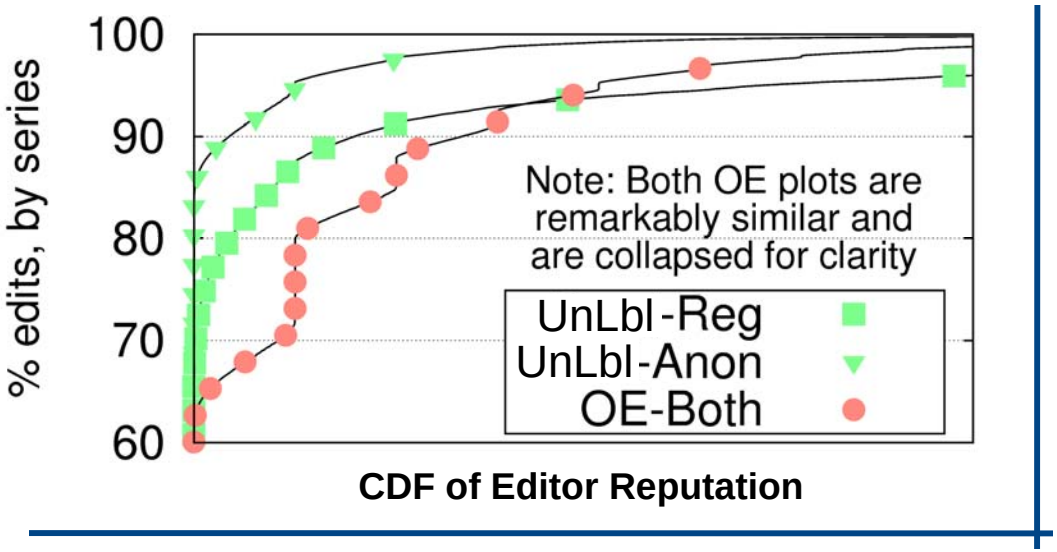
CATEGORY (with 100+ members)	PGs	OEs/PG
World Music Award Winners	125	162.27
Characters of Les Miserables	135	146.88
Former British Colonies	145	141.51

Categories with most OEs



Example of Category Rep. Calculation

Editor Reputation

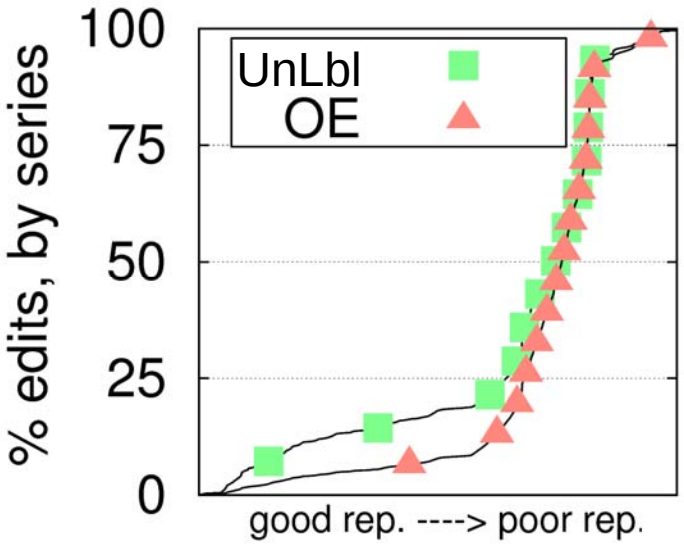


- Straightforward use of the *rep()* function, one-editor groups

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

Country Reputation

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



CDF of Country Reputation

RANK	COUNTRY	%-OEs
1	Italy	2.85%
2	France	3.46%
3	Germany	3.46%
…	…	…
12	Canada	11.35%
13	United States	11.63%
14	Australia	12.08%

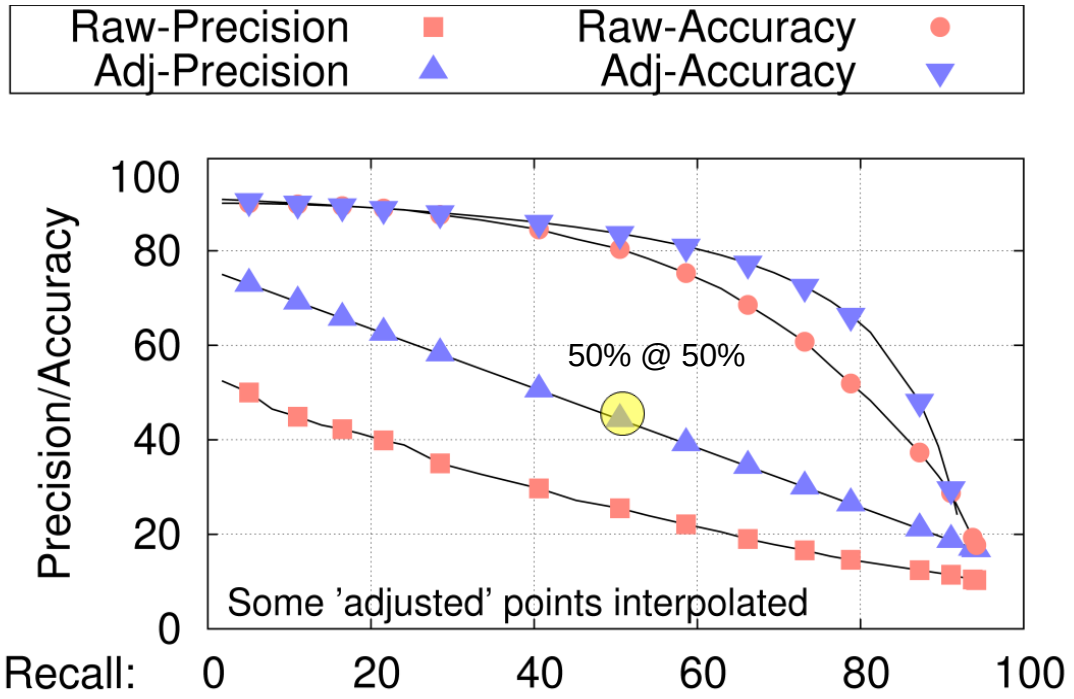
OE-rate (normalized) for countries with 100k+ edits

Off-line Performance

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

Recall: % total OEs classified correctly

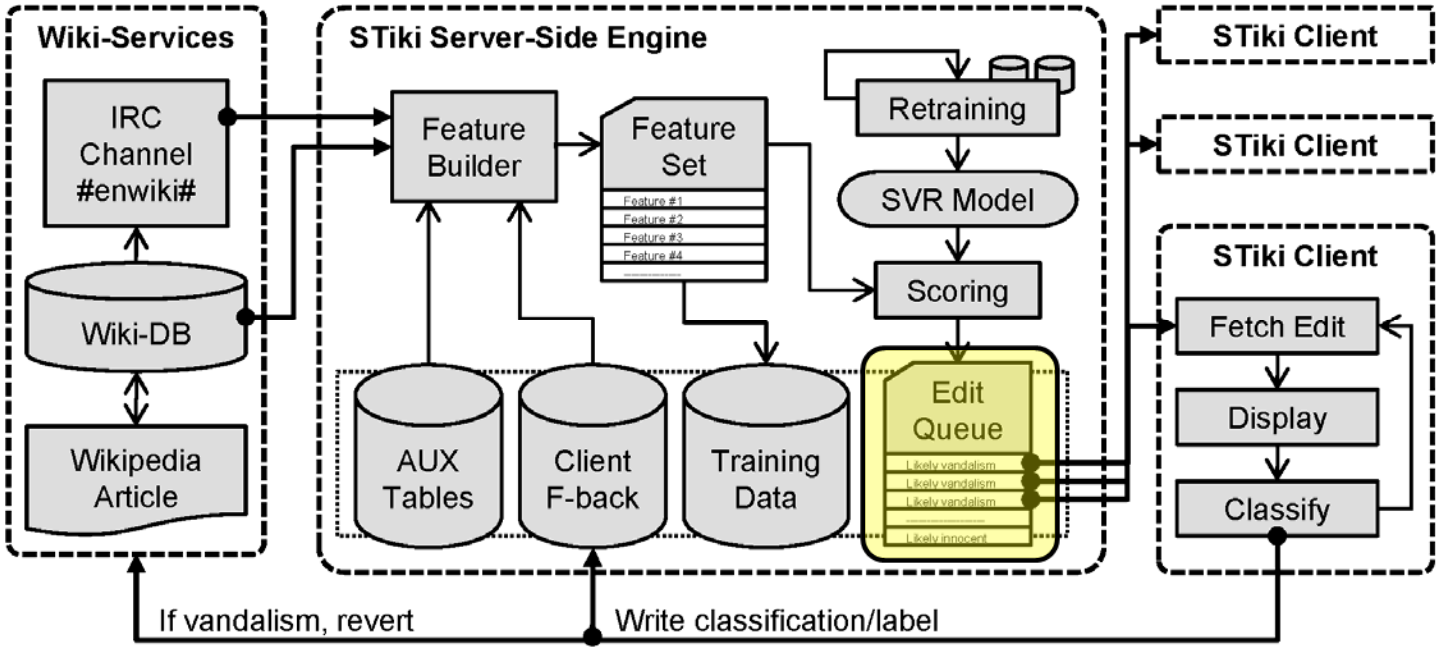
Precision: % of edits classified OE that are vandalism



STiki

STiki [4]: A real-time, on-Wikipedia implementation of the technique

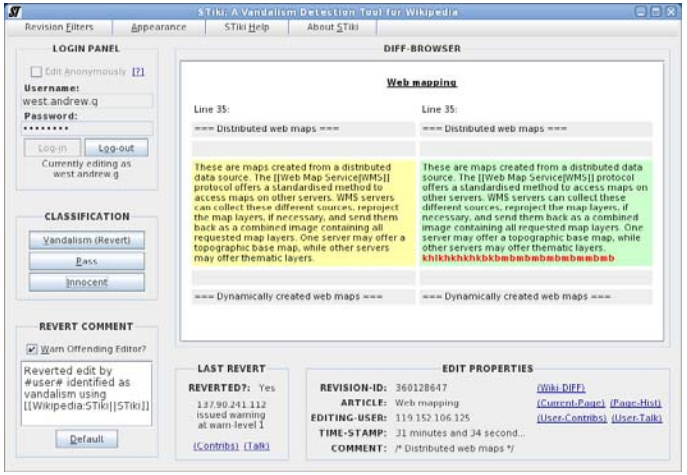
STiki Architecture



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

STiki Performance

- 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
    - 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 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

References

[1] S. Hao, N.A. Syed, N. Feamster, A.G. Gray, and S. Krasser. **Detecting spammers with SNARE: Spatiotemporal network-level automated reputation engine**. In *18th USENIX Security Symposium*, 2009

[2] M. Potthast, B. Stein, and R. Gerling. **Automatic vandalism detection in Wikipedia**. In *Advances in Information Retrieval*, 2008.

[3] R. Priedhorsky, J. Chen, S.K. Lam, K. Achier, L. Terveen, and J. Riedl. **Creating, destroying, and restoring value in Wikipedia**. In *GROUP `07, 2007.*

[4] A.G. West. **STiki: A vandalism detection tool for Wikipedia**. *http://en.wikipedia.org/wiki/Wikipedia:STiki*. Software, 2010.

[5] A.G. West, A.J. Aviv, J. Chang, and I. Lee. **Mitigating spam using spatio-temporal reputation**. *Technical report UPENN-MS-CIS-10-04*, Feb. 2010.

[6] A.G. West, S. Kannan, and I. Lee. **Detecting Wikipedia Vandalism via Spatio-Temporal Analysis of Revision Metadata**. In *EUROSEC `10*, April 2010.