Detecting Wikipedia Vandalism via Spatio-Temporal Analysis of Revision Metadata

Andrew G. West

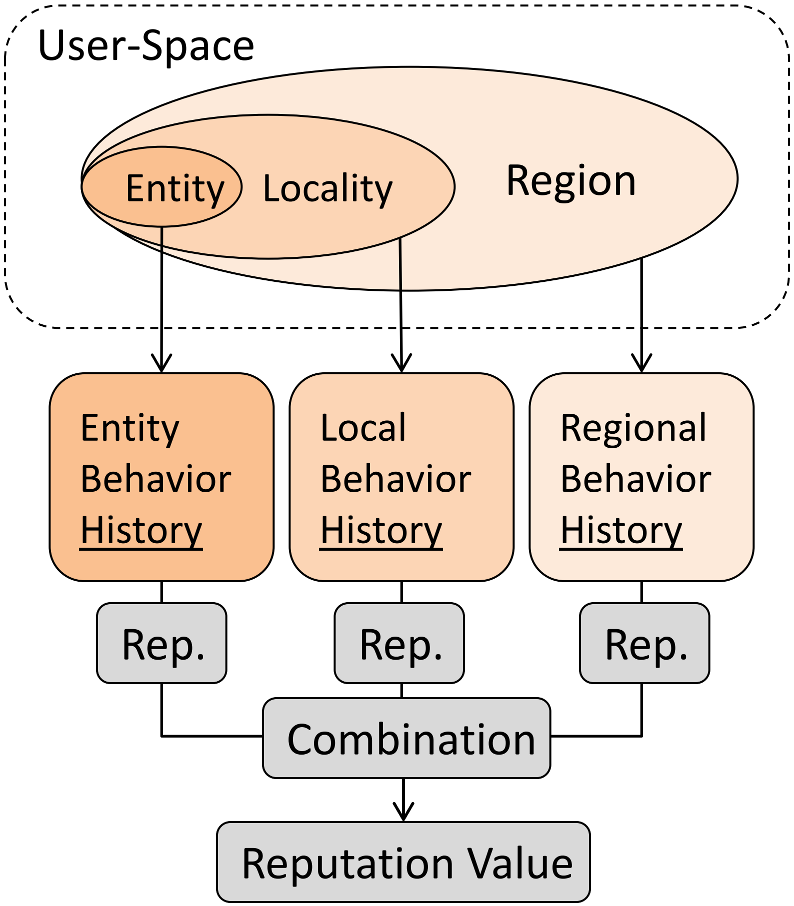
June 10, 2010

ONR-MURI Presentation

Where we left off….

FROM THE LAST MURI REVIEW

Spatio-Temporal Reputation

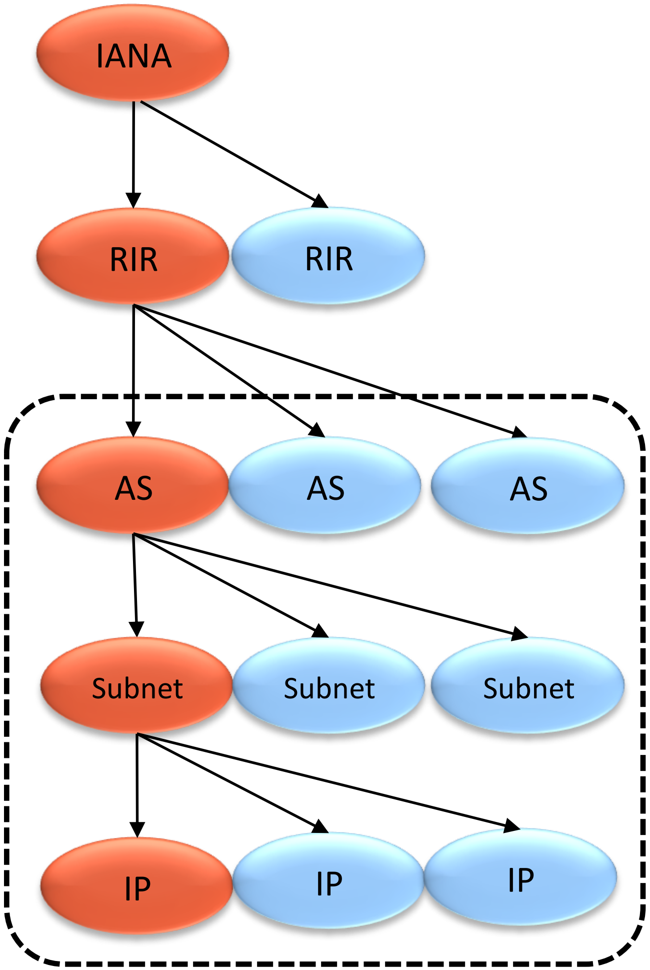


- Single-entity reputation values are the status quo
  - Issue: Sybil attacks (*e.g.*, spam botnets)
- Spatial reputation:
  - No entity-specific data? Use broader groupings
  - Exploit homophily
  - Clarity in borderline classification cases

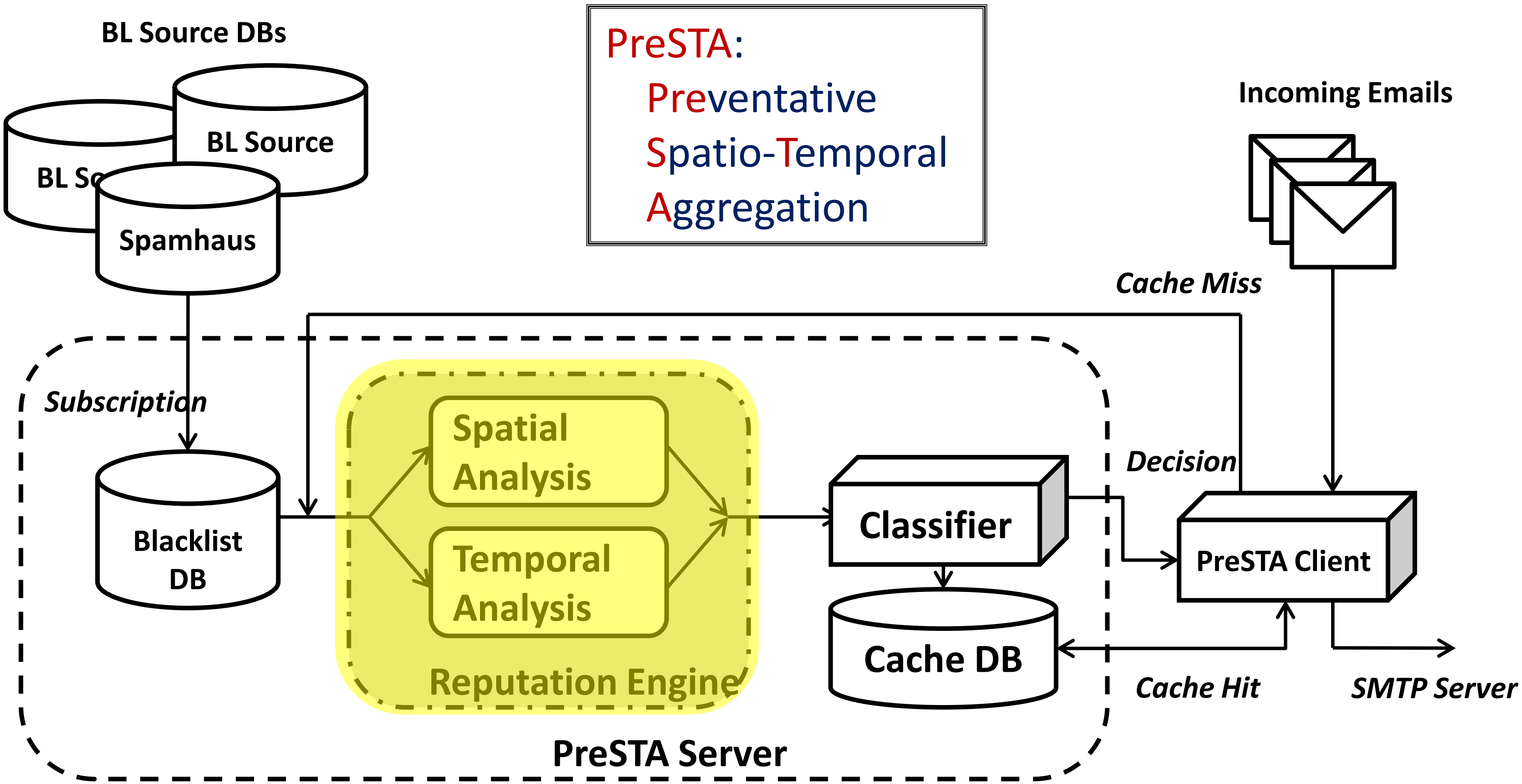
Hierarchical Groupings = TDG = QTM

- 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



New Contributions…

APPLYING SPATIO-TEMPORAL PROPERTIES TO WIKIPEDIA

Vandalism

- 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

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:
    - <u>Simple features</u>: Straightforward metadata analysis
    - <u>Aggregate features</u>: 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
  - Motivation: PreSTA [5] reputation algorithm
  - Article rep., editor rep., spatial reputations…
- Classifier performance
- STiki [4] (a real-time implementation)

Metadata

Wikipedia provides metadata via DB-dumps:

#	METADATA ITEM	NOTES
(1)	**Timestamp** of edit	In GMT locale
(2)	**Article** being edited	Able to deduce namespace from title
(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

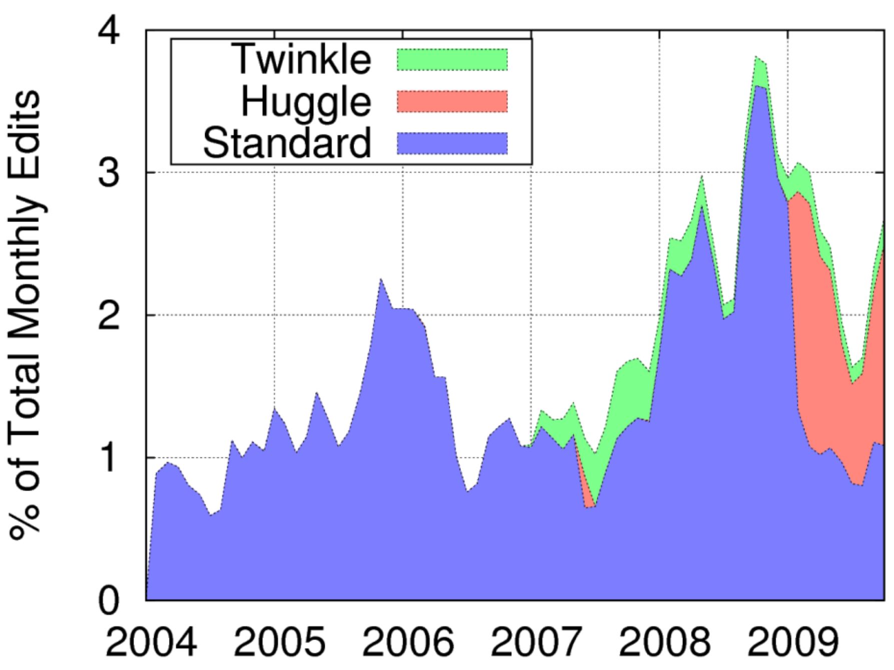
"Reversion" (*i.e.*, undo)

- 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*"

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



Prevalence/Source of Rollbacks

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

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

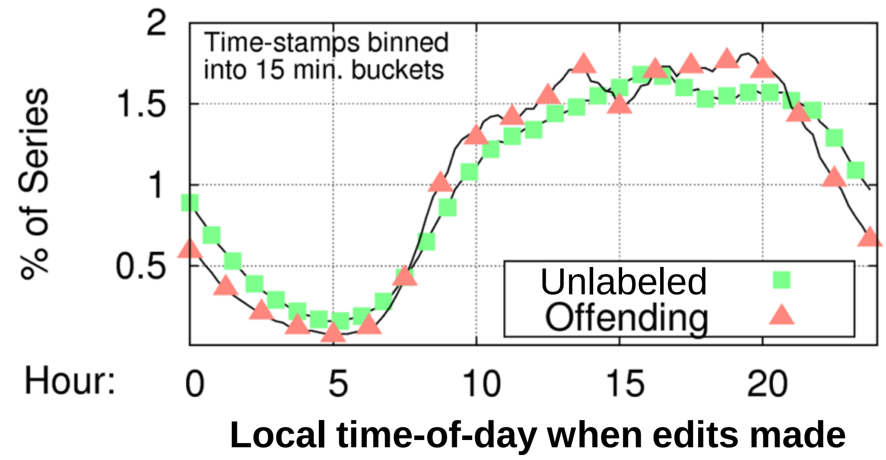
---

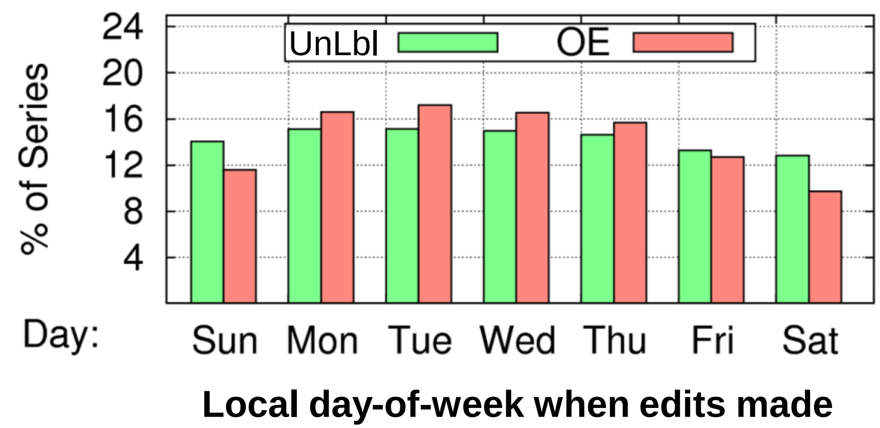
Motivating work: SNARE [1]

- 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





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%

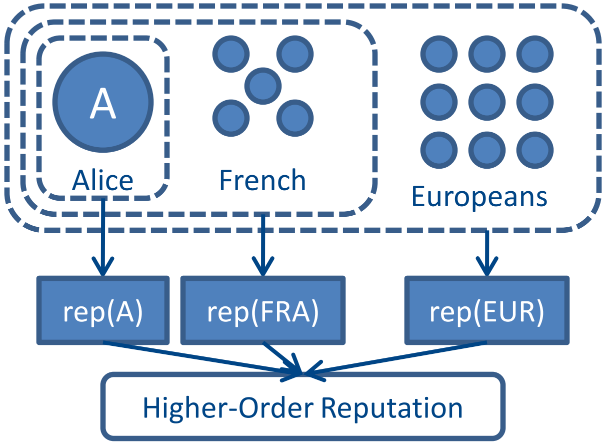
- Revision comment length
  - Vandals leave shorter comments (lazy-ness? or just minimizing bandwidth?)
- Privileged editors (and bots)
  - Huge contributors, but rarely vandalize

Aggregate Features

AGGREGATE FEATURES

PreSTA Algorithm

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



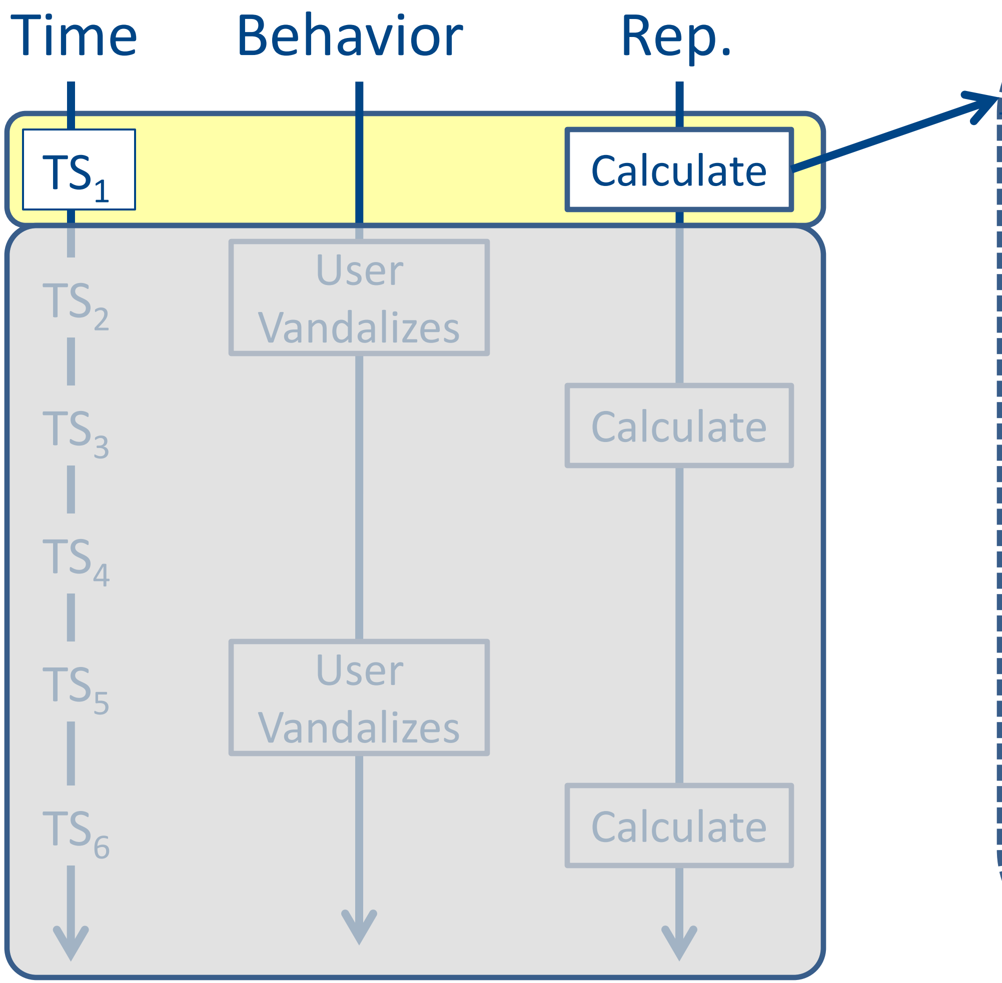
PreSTA [5]: Model for ST-rep:

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

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

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

Example Reputation



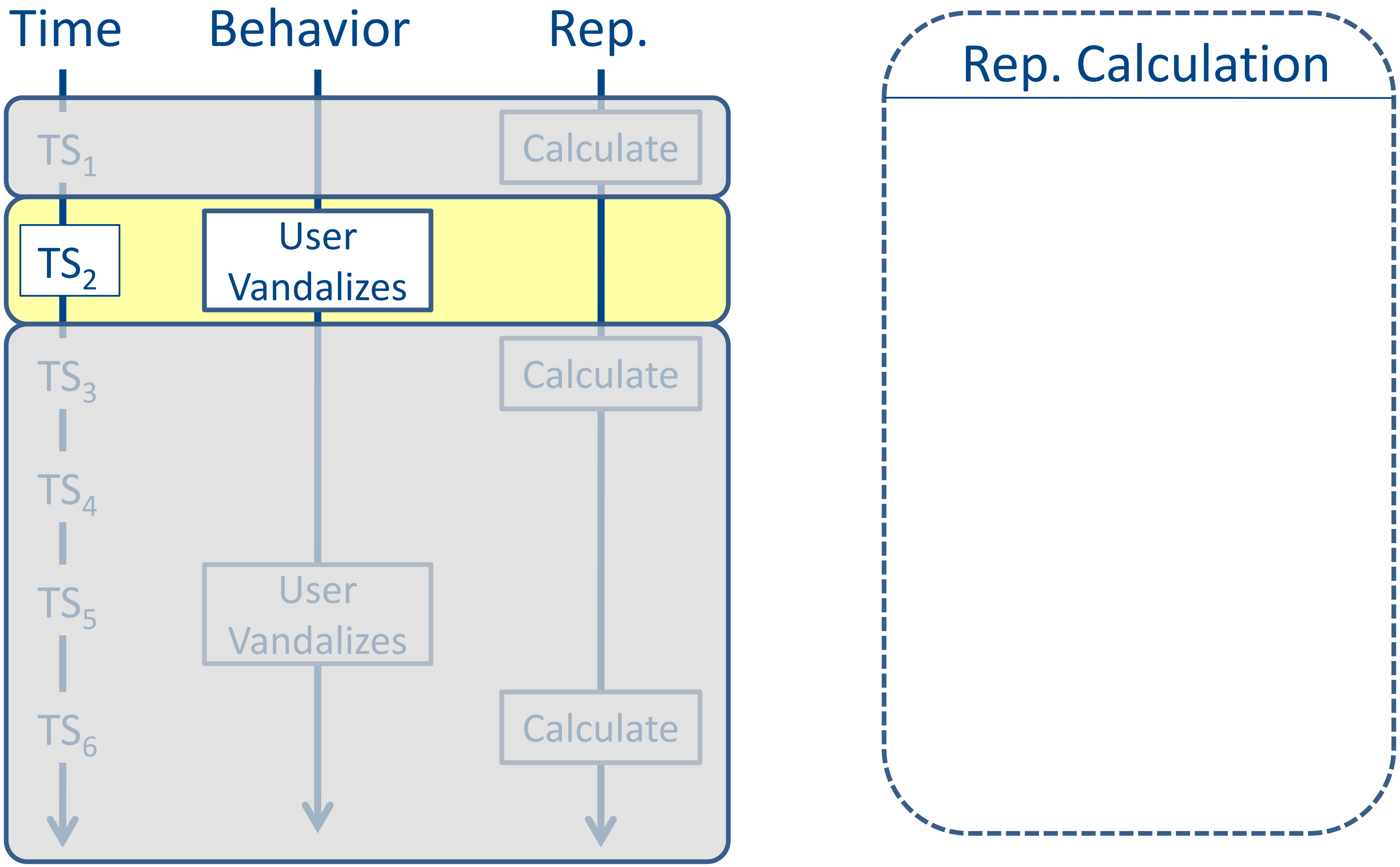
Rep. Calculation

No history?

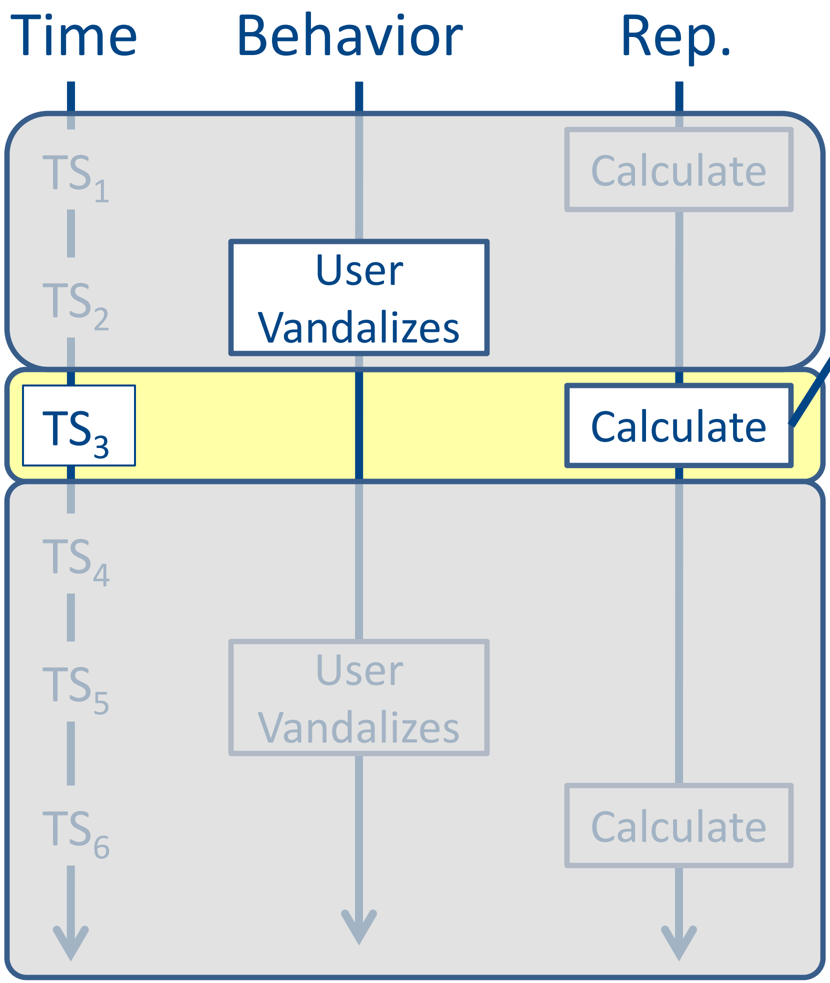
Reputation = 0.0

Completely Innocent!

Example Reputation



Example Reputation



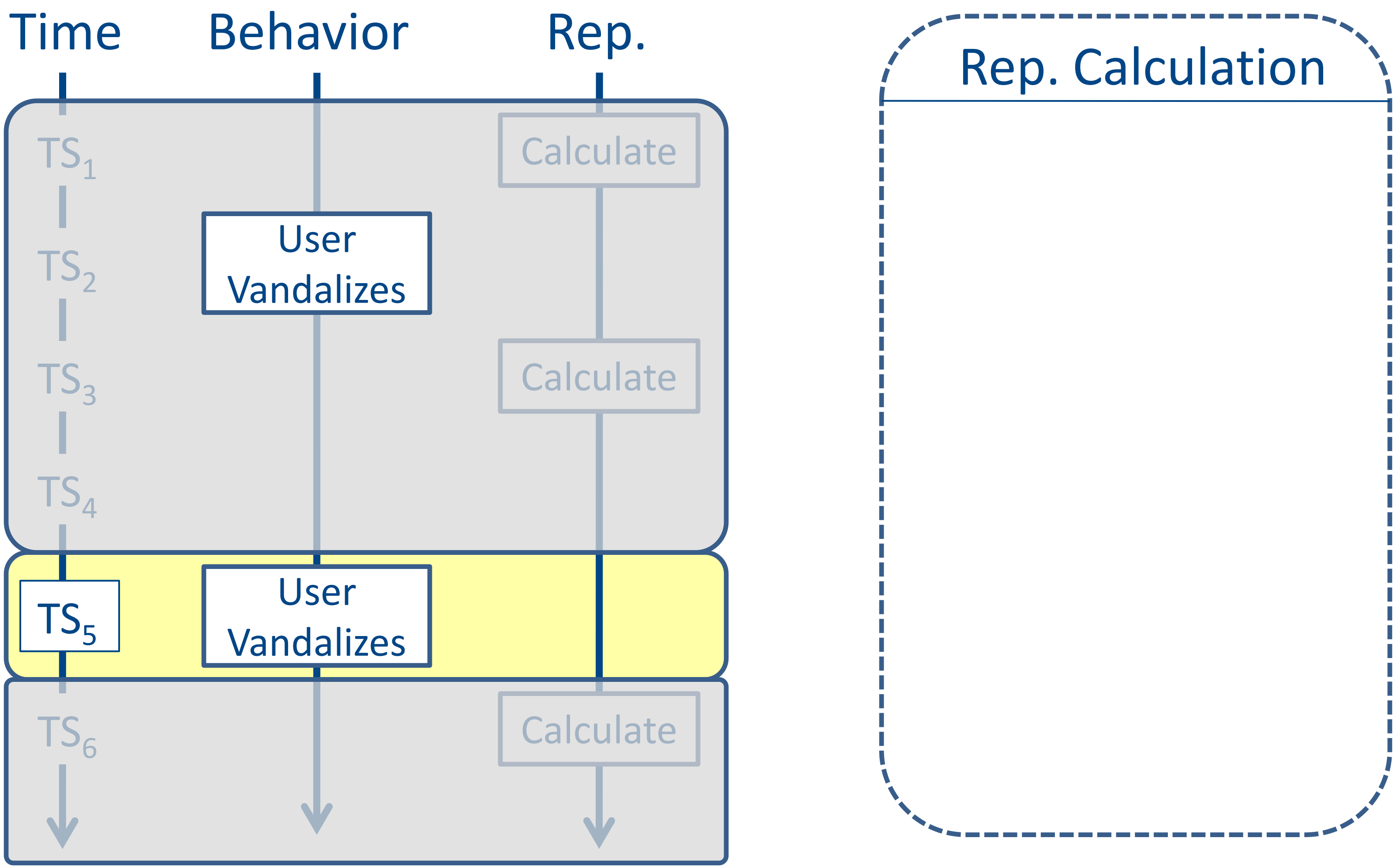
Rep. Calculation

One incident in history

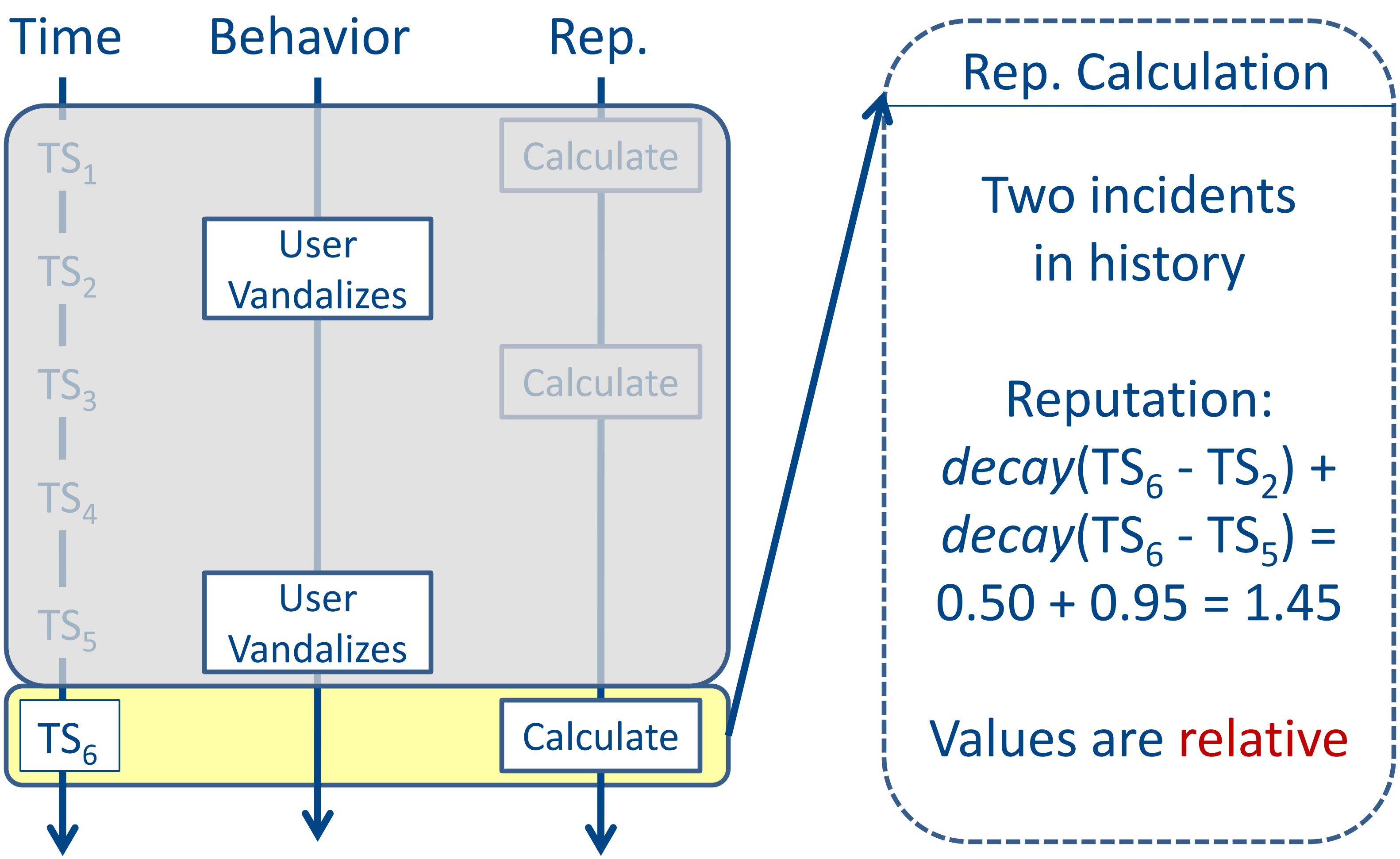
Reputation: *decay*(TS_3 - TS_2) = 0.95

decay() returns values on [0,1]

Example Reputation

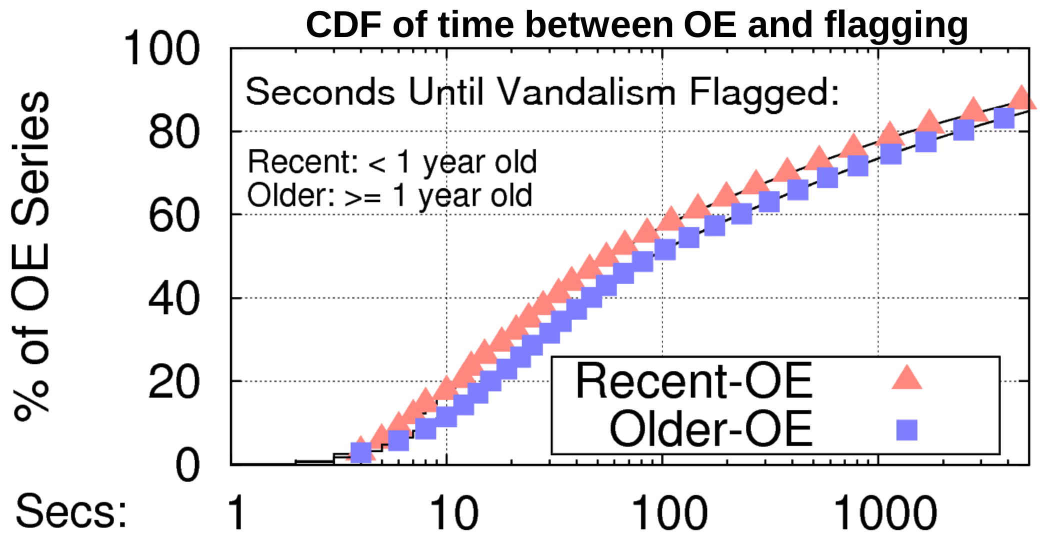


Example Reputation



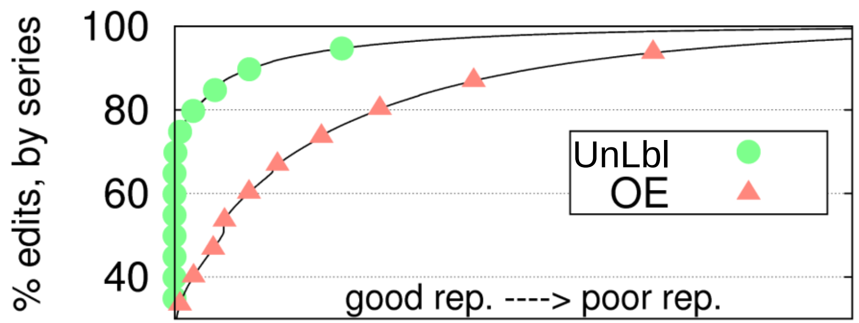
Rollback as Feedback

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

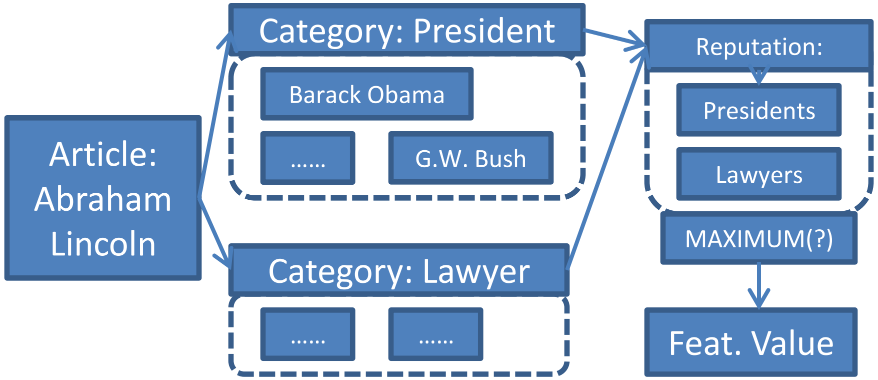
- Intuitively some topics are controversial and likely targets for vandalism (or temporally so).
- Trivial spatial grouping (size=1)
- 85% of OEs have non-zero rep (just 45% of random)

Category 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)

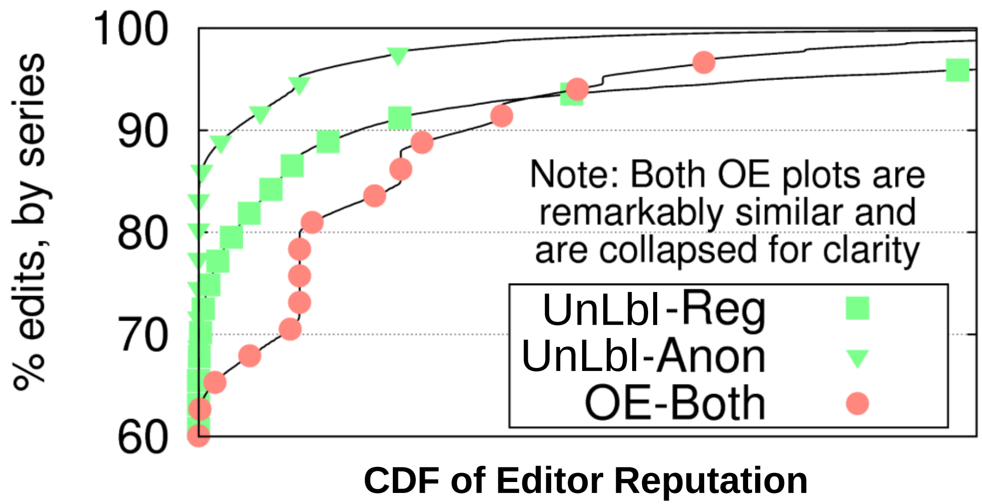
CATEGORY (with 100+ members)	PGs	OEs/PG
World Music Award Winners	125	162.27
Characters of Les Miserables	135	146.88
Former British Colonies	145	141.51

Categories with most OEs



Example of Category Rep. Calculation

Editor Reputation

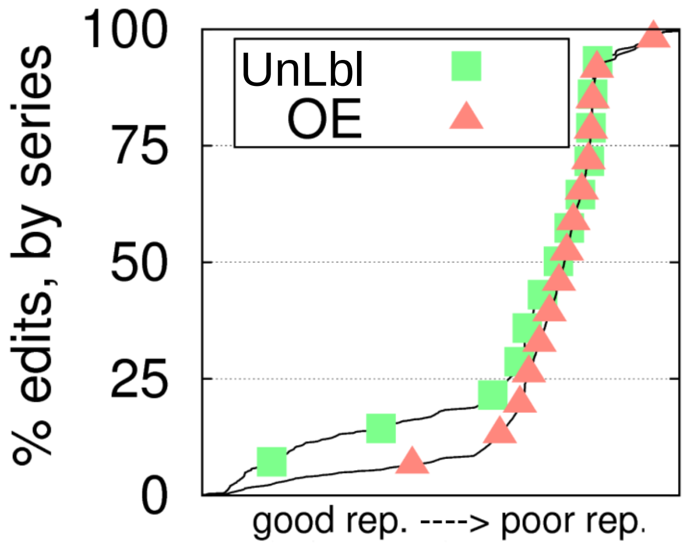


- Straightforward use of the *rep()* function, one-editor groups

- 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

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).

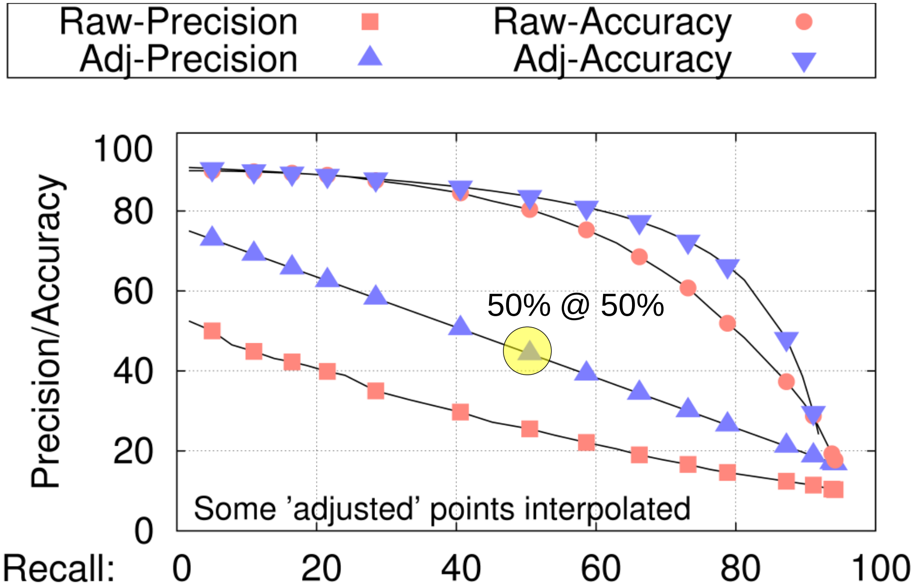
#	FEATURE
1	Edit time-of-day
2	Edit day-of-week
3	Time-since page edited
4	Time-since user reg.
5	Time-since last user OE
6	Rev. comment length
7	Article reputation
8	Category reputation
9	Editor reputation
10	Country reputation

Review of features used (only IP-editors)

Performance

- 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



Precision-recall trade-off

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?
  - Email spam: SNARE [1] and PreSTA [5]
  - This work shows it also works for Wikipedia

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*, pp. 663-668, 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: The 2007 ACM Conference on Supporting Group Work*, pp. 259-268, 2007.

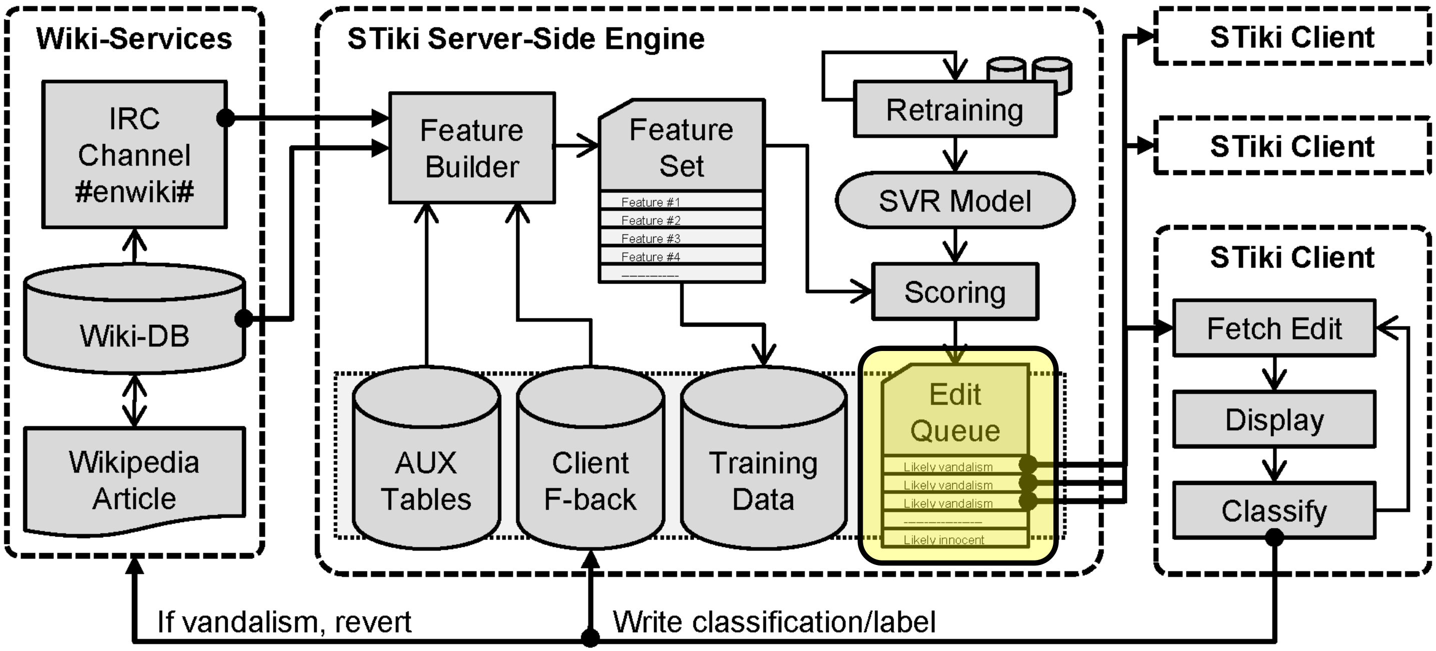
[4] A.G. West. **STiki: A vandalism detection tool for Wikipedia**. http://en.wikipedia.org/wiki/Wikipedia:STiki. *Software*, 2010.

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

STiki

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

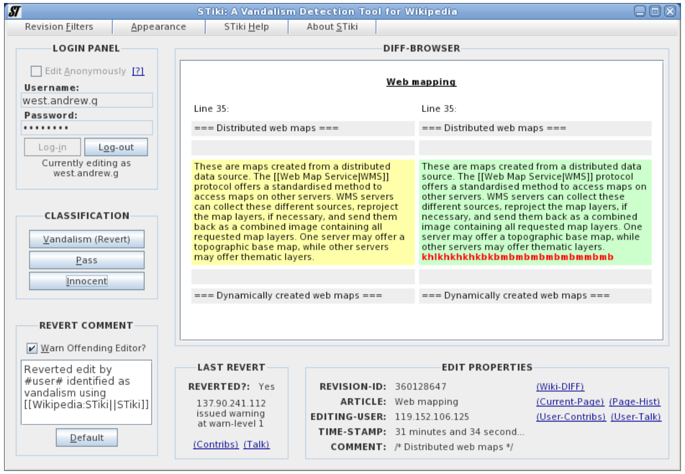
STiki Architecture



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 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?