7-2010

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

Andrew G. West  
*University of Pennsylvania, westand@cis.upenn.edu*

Sampath Kannan  
*University of Pennsylvania, kannan@cis.upenn.edu*

Insup Lee  
*University of Pennsylvania, lee@cis.upenn.edu*

Follow this and additional works at: [http://repository.upenn.edu/cis_papers](http://repository.upenn.edu/cis_papers)

Part of the [Computer Sciences Commons](http://repository.upenn.edu/cis_papers)

---

**Recommended Citation**


6th International Symposium on Wikis and Open Collaboration (WikiSym '10), Gdańsk, Poland, July 7-9, 2010.

This paper is posted at ScholarlyCommons, [http://repository.upenn.edu/cis_papers/453](http://repository.upenn.edu/cis_papers/453)  
For more information, please contact repository@pobox.upenn.edu.
STiki: An Anti-Vandalism Tool for Wikipedia Using Spatio-Temporal Analysis of Revision Metadata

Abstract
STiki is an anti-vandalism tool for Wikipedia. Unlike similar tools, STiki does not rely on natural language processing (NLP) over the article or diff text to locate vandalism. Instead, STiki leverages spatio-temporal properties of revision metadata. The feasibility of utilizing such properties was demonstrated in our prior work, which found they perform comparably to NLP-efforts while being more efficient, robust to evasion, and language independent.

STiki is a real-time, on-Wikipedia implementation based on these properties. It consists of, (1) a server-side processing engine that examines revisions, scoring the likelihood each is vandalism, and, (2) a client-side GUI that presents likely vandalism to end-users for definitive classification (and if necessary, reversion on Wikipedia). Our demonstration will provide an introduction to spatio-temporal properties, demonstrate the STiki software, and discuss alternative research uses for the open-source code.

Keywords
Wikipedia, collaborative applications, information security, intelligent routing, spatio-temporal processing

Disciplines
Computer Sciences | Physical Sciences and Mathematics

Comments
6th International Symposium on Wikis and Open Collaboration (WikiSym ’10), Gdańsk, Poland, July 7-9, 2010.

This other is available at ScholarlyCommons: http://repository.upenn.edu/cis_papers/453
STiki: An Anti-Vandalism Tool for Wikipedia using Spatio-Temporal Analysis of Revision Metadata

Andrew G. West  
University of Pennsylvania  
Philadelphia, PA, USA  
westand@cis.upenn.edu

Sampath Kannan  
University of Pennsylvania  
Philadelphia, PA, USA  
kannan@cis.upenn.edu

Insup Lee  
University of Pennsylvania  
Philadelphia, PA, USA  
lee@cis.upenn.edu

ABSTRACT
STiki is an anti-vandalism tool for Wikipedia. Unlike similar tools, STiki does not rely on natural language processing (NLP) over the article or diff text to locate vandalism. Instead, STiki leverages spatio-temporal properties of revision metadata. The feasibility of utilizing such properties was demonstrated in our prior work, which found they perform comparably to NLP-attacks while being more efficient, robust to evasion, and language independent.

STiki is a real-time, on-Wikipedia implementation based on these properties. It consists of: (1) a server-side processing engine that examines revisions, scoring the likelihood each is vandalism, and (2) a client-side GUI that presents likely vandalism to end-users for definitive classification (and if necessary, reversion on Wikipedia). Our demonstration will provide an introduction to spatio-temporal properties, demonstrate the STiki software, and discuss alternative research uses for the open-source code.

Categories and Subject Descriptors
H.5.3 [Group and Organization Interfaces]: collaborative computing, computer-supported cooperative work; K.6.5 [Management of Computing and Information Systems]: Security and Protection

General Terms
Design, Management, Human Factors, Security

1. VANDALISM DETECTION & STIKI

We informally define Wikipedia vandalism to be any revision that is non-value adding, offensive, or destructive in its removal of content. The detrimental impact of vandalism is large, with one source [1] estimating the number of damaged page-views to be in the hundreds of millions. Detecting vandalism is difficult; it has many varied and subtle forms.

To this end, our prior research [4] investigated the spatio-temporal properties of edit metadata as an alternative means of detection, complementing techniques based on natural language processing. The metadata of an edit includes: the (1) edit time-stamp, (2) article being edited, (3) user-name or IP of the editor, and (4) the revision comment. Meanwhile, temporal properties are a function of the time at which an event occurs and spatial properties are appropriate wherever a distance or membership function can be defined.

Our prior work [4] identifies ten spatio-temporal properties (see Tab. 1) that are effective in locating malicious edits. Simple features include the edit time-of-day, revision comment length, etc. Aggregate features combine time-decayed behavioral observations (feedback) to create reputations [3] for single entities and spatial groupings thereof.

STiki [2] exploits these features, processing edits in real-time and enabling on-Wikipedia reversion. It consists of:

- Server-Side Engine: Listens on an IRC channel for Wikipedia edits. When one is made, the associated metadata is fetched. Combined with auxiliary data (e.g., geolocation), this is sufficient to compute the feature-set. A machine-learning technique called Support Vector Regression (SVR) assigns the edit a real-value vandalism score. SVR is trained over older edits labeled via, (1) automatic parsing of administrative reverts called rollbacks, and, (2) user-provided feedback from STiki clients.

- Client-Side GUI: Presents likely vandalism to users, displaying intuitively-colored edit diffs (see Fig. 1). Edits identified as vandalism are reverted on Wikipedia. In either case, feedback improves future server-side scoring.

A detailed STiki system workflow diagram is provided in Fig. 2. STiki is platform-independent (Java). Both the GUI executable and full source-code are available at [2].

2. PRESENTER & AUDIENCE BENEFIT

The presenter(s) wish to solicit feedback from casual users and vandalism experts regarding STiki’s ease-of-use and methodology. Further, exposure will result in a larger user-base – critical given the nature of the feedback loop.

Meanwhile, the audience will be introduced to an innovative line of Wiki-relevant research. They will be invited not only to become STiki users, but to become contributors by extending the feature-set, improving GUI functionality, or interfacing with our tool. Lastly, we will discuss how STiki code can be modified to support alternative research goals.
References


---

Figure 1: STiki GUI displaying a revision exhibiting vandalism (nonsense).

Figure 2: Simplified STiki workflow diagram.

Table 1: STiki features [4].