2-2011

Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features

B. Thomas Adler
University of California, Santa Cruz, thumper@soe.ucsc.edu

Luca de Alfaro
University of California, Santa Cruz -- Google, luca@dealfaro.com

Santiago M. Mola-Velasco
Universidad Politcnica de Valencia, smola@dsic.upv.es

Paolo Rosso
Universidad Politcnica de Valencia, rosso@dsic.upv.es

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

Recommended Citation

B. Thomas Adler, Luca de Alfaro, Santiago M. Mola-Velasco, Paolo Rosso, and Andrew G. West, "Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features", *Lecture Notes in Computer Science: Computational Linguistics and Intelligent Text Processing* 6609, 277-288. February 2011. http://dx.doi.org/10.1007/978-3-642-19437-5_23

CICLing '11: Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics, Tokyo, Japan, February 20-26, 2011.

Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features

Abstract

Wikipedia is an online encyclopedia which anyone can edit. While most edits are constructive, about 7% are acts of vandalism. Such behavior is characterized by modifications made in bad faith; introducing spam and other inappropriate content. In this work, we present the results of an effort to integrate three of the leading approaches to Wikipedia vandalism detection: a spatio-temporal analysis of metadata (STiki), a reputation-based system (WikiTrust), and natural language processing features. The performance of the resulting joint system improves the state-of-the-art from all previous methods and establishes a new baseline for Wikipedia vandalism detection. We examine in detail the contribution of the three approaches, both for the task of discovering fresh vandalism, and for the task of locating vandalism in the complete set of Wikipedia revisions.

Keywords

Wikipedia, wiki, collaboration, vandalism, machine learning, metadata, natural-language processing, reputation

Disciplines

Other Computer Sciences

Comments

CICLing '11: Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics, Tokyo, Japan, February 20-26, 2011.

Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features

B. Thomas Adler[1], Luca de Alfaro[2], Santiago M. Mola-Velasco[3], Paolo Rosso[3], and Andrew G. West[4] *

[1] University of California, Santa Cruz, USA – *thumper@soe.ucsc.edu*
[2] Google and UC Santa Cruz, USA – *luca@dealfaro.com*
[3] NLE Lab. - ELiRF - DSIC. Universidad Politcnica de Valencia. Camino de Vera s/n 46022 Valencia, Spain – {*smola,prosso*}*@dsic.upv.es*
[4] University of Pennsylvania, Philadelphia, USA – *westand@cis.upenn.edu*

Abstract. Wikipedia is an online encyclopedia which anyone can edit. While most edits are constructive, about 7% are acts of vandalism. Such behavior is characterized by modifications made in bad faith; introducing spam and other inappropriate content.

In this work, we present the results of an effort to integrate three of the leading approaches to Wikipedia vandalism detection: a spatio-temporal analysis of metadata (STiki), a reputation-based system (WikiTrust), and natural language processing features. The performance of the resulting joint system improves the state-of-the-art from all previous methods and establishes a new baseline for Wikipedia vandalism detection. We examine in detail the contribution of the three approaches, both for the task of discovering fresh vandalism, and for the task of locating vandalism in the complete set of Wikipedia revisions.

1 Introduction

Wikipedia [1] is an online encyclopedia that anyone can edit. In the 10 years since its creation, 272 language editions have been created, with 240 editions being actively maintained as of this writing [2]. Wikipedia's English edition has more than 3 million articles, making it the biggest encyclopedia ever created. The encyclopedia has been a collaborative effort involving over 13 million registered users and an indefinite number of anonymous editors [2]. This success has made Wikipedia one of the most used knowledge resources available online and a source of information for many third-party applications.

The open-access model that is key to Wikipedia's success, however, can also be a source of problems. While most edits are constructive, some are *vandalism*, the result of attacks by pranksters, lobbyists, and spammers. It is estimated that about 7% of the edits to Wikipedia are vandalism [3]. This vandalism is removed by a number of dedicated individuals who patrol Wikipedia articles looking for

* Authors appear alphabetically. Order does not reflect contribution magnitude.

such damage. This is a daunting task: the English Wikipedia received 10 million edits between August 20 and October 10, 2010[5], permitting the estimation that some 700,000 revisions had to be reverted in this period.

Wikipedia vandalism also creates problems beyond the effort required to remove it. Vandalism lends an aura of unreliability to Wikipedia that exceeds the statistical extent of the damage. For instance, while Wikipedia has the potential to be a key resource in schools at all levels due to its breadth, overall quality, and free availability – the risk of exposing children to inappropriate material has been an obstacle to adoption [4, 5]. Likewise, the presence of vandalism has made it difficult to produce static, high-quality snapshots of Wikipedia content, such as those that the Wikipedia 1.0 project plans to distribute in developing countries with poor Internet connectivity[6].

For these reasons, autonomous methods for locating Wikipedia vandalism have long been of interest. The earliest such attempts came directly from the user community, which produced several *bots*. Such bots examine newly-created revisions, apply hand-crafted rule sets, and detect vandalism where appropriate. Over time, these approaches grew more complex, using a vast assortment of methods from statistics and machine learning. Feature extraction and machine-learning, in particular, have proven particularly adept at the task – capturing the top spots at the recent PAN 2010 vandalism detection competition[7].

In this paper, we present a system for the automated detection of Wikipedia vandalism that constitutes, at the time of writing, the best-performing published approach. The set of features includes those of the two leading methodologies in PAN 2010: the Mola-Velasco system [6] (NLP) and the WikiTrust system [7] (reputation). Further, the features of the STiki system [8] (metadata) are included, which has academic origins, but also has a GUI frontend [9] enabling actual on-Wikipedia use (and has become a popular tool on English Wikipedia).

Since the systems are based largely on non-overlapping sets of features, we show that the combined set of features leads to a markedly superior performance. For example, 75% precision is possible at 80% recall. Moreover, fixing precision at 99% produces a classifier with 30% recall – perhaps enabling autonomous use.

Most importantly, we investigate the relative merit of different classes of features of different computational and data-gathering costs. Specifically, we consider (1) metadata, (2) text, (3) reputation, and (4) language features. *Metadata* features are derived from basic edit properties (*e.g.,* timestamp), and can be computed using straightforward database processing. *Text* features are also straightforward, but may require text processing algorithms of varying sophistication. *Reputation* features refer to values that analyze the behavior history of some entity involved in the edit (*e.g.,* an individual editor).Computing such reputations comes with a high computational cost, as it is necessary to analyze large portions of Wikipedia history. Finally, *language* features are often easy to compute for specific languages, but require adaptation to be portable.

[5] `http://en.wikipedia.org/wiki/User:Katalaveno/TBE`
[6] `http://en.wikipedia.org/wiki/Wikipedia:Wikimedia_School_Team`
[7] Held in conjunction with CLEF 2010. See `http://pan.webis.de`

Moreover, we consider two classes of the vandalism detection problem: (1) the need to find *immediate* vandalism, (*i.e.*, occurring in the most recent revision of an article), and (2) *historical* vandalism, (*i.e.*, occurring in any revision including past ones). Immediate vandalism detection can be used to alert Wikipedia editors to revisions in need of examination. The STiki tool [9], whose features are included in this work, has been successfully used in this fashion to revert over 30,000 instances of vandalism on the English Wikipedia.

Historical vandalism detection can be used to select, for each article, a recent non-vandalized revision from the entire article history. The WikiTrust system (whose features are also included in this work) was recently used to select the revisions for the Wikipedia 0.8 project, a static snapshot of Wikipedia intended to be published in DVD form[8]. We consider historical detection to be an interesting variation of the standard Wikipedia vandalism detection problem, as it has the potential to use *future* information in edit analysis.

Combining the feature-vectors of the three systems, our meta-detector produces an area under the precision-recall curve (AUC-PR) of 81.83% for *immediate* vandalism detection. This is a significant improvement over the performance achieved from using any two of the systems in combination (performance ranges between 69% and 76%). Moreover, the meta-detector far exceeds the best known system in isolation (whose features are included), which won the PAN 2010 competition with 67% AUC-PR. Similar improvements were seen when performing the *historical* detection task. In a 99% precision setting, the meta-system could revert 30% of vandalism without human intervention.

The remainder of the work is structured as follows: Section 2 overviews related work. Section 3 describes our features and their categorization. Section 4 presents results. Finally, we conclude in Section 5.

2 A Brief History of Wikipedia Vandalism Detection

Given the damage that vandalism causes on Wikipedia, it is no surprise that attempts to locate vandalism automatically are almost as old as Wikipedia itself. The earliest tools consisted of *bots* that would labeled vandalism using handcrafted rule systems – encoding heuristic vandalism patterns. Examples of such bots include [10–14]. Typical rules were narrowly targeted, including: the amount of text inserted or deleted, the ratio of capital letters, the presence of vulgarisms detected via regular expressions, *etc.*.

Given the community's low tolerance for accidentally categorizing a legitimate edit as vandalism, such systems operated with high precision, but low recall. For instance, ClueBot was found to have 100% precision in one study, but fairly low recall: below 50% for any vandalism type, and below 5% for insertions [15]; a different study confirmed this low recall [16].

The idea that an edit's *textual content* is a likely source of indicative features has been investigated by several different research groups [15–19]. Casting the

[8] `http://blogs.potsdam.edu/wikipediaoffline/2010/10/26/wikipedia-version-0-8-is-coming/`

problem as a machine-learning binary classification problem, Potthast *et al.* [15] used manual inspection to inspire a feature set based on metadata and content-level properties and built a classifier using logistic regression. Smets *et al.* [16] used Naïve Bayes applied to a bag-of-words model of the edit text. Chin *et al.* [19] delve deeper into the field of natural language processing by constructing statistical language models of an article from its revision history.

A different way of looking at edit content is the intuition that appropriate content somehow "belongs together." For example, *cohesion* can be measured via compression rates over consecutive editions of an article [16, 18]. If inappropriate content is added to the article, then the compression level is lower than it would be for text which is similar to existing content. A drawback of this approach is that it tends to label as vandalism any large addition of material, regardless of its quality, while overlooking the small additions of insults, racial epithets, pranks, and spam that comprise a significant portion of vandalism.

The idea of using reputation systems to aid in vandalism detection was advanced in [20–22]. West *et al.* [8] apply the idea of reputations to editors and articles, as well as spatial groupings thereof — including geographical regions and topical categories.

Many previous works have some small dependence on metadata features [15, 17, 23], but only as far as it encoded some aspect of human intuition about vandalism. Drawing inspiration from email spam research, West *et al.* [8] demonstrated that the broader use of metadata can be very effective, suggesting that there are more indicators of vandalism than are apparent to the human eye.

The first systematic review and organization of features was performed by Potthast *et al.* [24] as part of the vandalism detection competition associated with PAN 2010. Potthast *et al.* conclude their analysis by building a meta-classifier based on all nine competition entries, and finds it significantly outperforms any single entry. As our own work will confirm, a diverse array of features is clearly beneficial when attacking the vandalism detection problem. Our work extends that of Potthast by concatenating entire feature vectors (not just the single variable output) and by analyzing the effectiveness of unique feature classes.

3 Vandalism Detection

On Wikipedia, every article is stored as a sequence of revisions in chronological order. Visitors to Wikipedia are shown the latest revision of an article by default; if they so choose, they can edit it, producing a new revision. Some of these revisions are *vandalism*. Vandalism has been broadly defined as any edit performed in bad faith, or with the intent to deface or damage Wikipedia. In this work, we do not concern ourselves with the definition of vandalism; rather, we use the PAN-WVC-10 corpus as our ground-truth. The corpus consists of over 32,000 edits (some 2,400 vandalism), each labeled by 3 or more annotators from Amazon Mechanical Turk. See [24] for additional details.

In order to detect vandalism, we follow a classical architecture: feature extraction, followed by data-trained classification. Features can be obtained from:

(1) the revision itself, (2) from comparison of the revision against another revision (*i.e.,* a `diff`), or (3) from information derived from previous or subsequent revisions. For instance, the ratio of uppercase to lowercase characters inserted is one feature, as is the edit distance between a revision and the previous one on the same article. The feature vectors are then used to train and classify. As a classifier, we use the Random Forest[9] model [26]. We perform evaluation using 10-fold cross-validation over the entire PAN-WVC-10 corpus.

We consider two types of vandalism detection problem: immediate and historic. *Immediate* vandalism detection is the problem of detecting vandalism in the most recent revision of an article; *historic* detection is the problem of finding vandalism in any past revision. For immediate vandalism detection, one can only make use of the information available at the time a revision is committed. In particular, in immediate vandalism detection, information gathered from *subsequent* revisions cannot be used to decide whether a particular revision is vandalism or not. In contrast, historical vandalism detection permits the use of any feature. We propose one such possible feature: the implicit judgements made by later editors in deciding whether to keep some or all text previously added.

We divide our features into classes, according to the complexity required to compute them, and according to the difficulty of generalizing them across multiple languages. These classes are: Metadata, Text, Reputation, and Language, abbreviated as **M**, **T**, **R**, and **L**, respectively. Our work is based directly on the previous works of [6, 7] and [8, 9]. What follows is a discussion of representative features from each class. For a complete feature listing, see Table 1.

3.1 Metadata

Metadata (M) refers to properties of a revision that are immediately available, such as the identity of the editor, or the timestamp of the edit. This is an important class of features because it has minimal computational complexity. Beyond the properties of each revision found directly in the database (*e.g.* whether the editor is anonymous, used by nearly every previous work), there are some examples that we feel expose the unexpected similarities in vandal behavior:

- **Time since article last edited** [8]. Highly-edited articles are frequent targets of vandalism. Similarly, quick fluctuations in content may be indicative of edit wars or other controversy.
- **Local time-of-day** and **day-of-week** [8]. Using IP geolocation, it is possible to determine the *local* time when an edit was made. Evidence shows vandalism is most prominent during weekday "school/office hours."
- **Revision comment length** [6–8]. Vandals decline to follow community convention by leaving either very short revision comments or very long ones.

[9] We used the Random Forest implementation available in the Weka Framework 3.7 [25], available at`http://www.cs.waikato.ac.nz/ml/weka/`.

3.2 Text

We label as *Text* (T) those language-independent features derived from analysis of the edit content. Therefore, very long articles may require a significant amount of processing. As the content of the edit is the true guide to its usefulness, there are several ideas for how to measure that property:

- **Uppercase ratio** and **digit ratio** [6, 8]. Vandals sometimes will add text consisting primarily of capital letters to attract attention; others will change only numerical content. These ratios (and similar ones [6]) create features which capture behaviors observed in vandals.
- **Average** and **minimum edit quality** [7] (Historic only). Comparing the content of an edit against a future version of the article provides a way to measure the Wikipedia community's approval of the edit [17, 22]. To address the issue of edit warring, the comparison is done against several future revisions. This feature uses edit distance (rather than the blunt detection of reverts) to produce an implicit quality judgement by later edits; see [22].

3.3 Language

Similar to text features, *Language* (L) features must inspect edit content. A distinction is made because these features require expert knowledge about the (natural) language. Thus, these features require effort to be re-implemented for each different language. Some of the features included in our analysis are:

- **Pronoun frequency** and **pronoun impact** [6]. The use of first and second-person pronouns, including slang spellings, is indicative of a biased style of writing discouraged on Wikipedia (non-neutral point-of-view). *Frequency* considers the ratio of first and second- person pronouns relative to the size of the edit. *Impact* is the percentage increase in first and second-person pronouns that the edit contributes to the overall article.
- **Biased** and **bad words** [6]. Certain words indicate a bias by the author (*e.g.* superlatives: "coolest", "huge"), which is captured by a list of regular expressions. Similarly, a list of bad words captures edits which appear inappropriate for an encyclopedia (*e.g.* "wanna", "gotcha") and typos (*e.g.* "seperate"). Both these lists have corresponding frequency and impact features that indicate how much they dominate the edit and increase the presence of biased or bad words in the overall article.

3.4 Reputation

We consider a feature in the *Reputation* (R) category if it necessitates extensive historical processing of Wikipedia to produce a feature value. The high cost of this computational complexity is sometimes mitigated by the ability to build on earlier computations, using incremental calculations.

– **User reputation** [7] (Historic only[10]) User reputation as computed by WikiTrust [22]. The intuition is that users who have a history of good contributions, and therefore high reputation, are unlikely to commit vandalism.
– **Country reputation** [8]. For anonymous/IP edits, it is useful to consider the geographic region from which an edit originates. This feature represents the likelihood that an editor from a particular country is a vandal, by aggregating behavior histories from that same region. Location is determined by geo-locating the IP address of the editor.
– **Previous** and **current text trust histogram** [7]. When high-reputation users revise an article and leave text intact, that text accrues reputation, called "trust" [7]. Features are, (1) the histogram of word trust in the edit, and (2) the difference between the histogram before, and after, the edit.

4 Experimental Results

In this section, we present results and discussion of our experiments using different combinations of meta-classifers. Table 2 summarizes the performance of these subsets per the experimental setup described in Section 3. We present the results in terms of area under curve[11] (AUC) for two curves: the precision-recall curve (PR), and the receiver operating characteristics (ROC) curve. The results in terms of AUC-ROC are often presented for binary classification problem (which vandalism detection is), but AUC-PR better accounts for the fact that vandalism is a rare phenomenon [27], and offers a more discriminating look into the performance of the various feature combinations.

In Figure 1 we show precision-recall curves for each system, distinguishing between immediate and historic vandalism cases. Only [7] considers features explicitly for the historic cases. We find a significant increase in performance when transitioning from immediate to historical detection scenarios.

Analysis of our feature taxonomy, per Figure 2, leads to some additional observations in a comparison between immediate and historic vandalism tasks:

– Most obvious is the improvement in the performance of the Language (L) set, due entirely to the **next comment revert** feature. The feature evaluates

[10] In a live system, user reputation is available at the time a user makes an edit, and therefore, user reputation is suitable for immediate vandalism detection. However, since WikiTrust only stores the current reputation of users, *ex post facto* analysis was not possible for this study.

[11] `http://mark.goadrich.com/programs/AUC/`

[12] Note that performance numbers reported for [6] and [7] differ from those reported in [24] due to our use of 10-fold cross validation over the entire PAN2010 corpus and differences in ML models (*e.g.,* ADTree vs. Random Forest). We do not list the performance of the PAN 2010 Meta Detector because it was evaluated with an unknown subset of the PAN 2010 corpus, and is therefore not precisely comparable.

[13] Note that statistics for the "West *et al.*" system are strictly the metadata ones described in [8], and not the more general-purpose set used in the online tool [9].

Table 1. Comprehensive listing of features used, organized by class. Note that features in the "!Z" (not zero-delay) class are those that are only appropriate for historical vandalism detection.

FEATURE	CLS	SRC	DESCRIPTION
IS_REGISTERED	M	[6–8]	Whether editor is anonymous/registered (boolean)
COMMENT_LENGTH	M	[6–8]	Length (in chars) of revision comment left
SIZE_CHANGE	M	[6–8]	Size difference between prev. and current versions
TIME_SINCE_PAGE	M	[7, 8]	Time since article (of edit) last modified
TIME_OF_DAY	M	[7, 8]	Time when edit made (UTC, or local w/geolocation)
DAY_OF_WEEK	M	[8]	Local day-of-week when edit made, per geolocation
TIME_SINCE_REG	M	[8]	Time since editor's first Wikipedia edit
TIME_SINCE_VAND	M	[8]	Time since editor last caught vandalizing
SIZE_RATIO	M	[6]	Size of new article version relative to new one
PREV_SAME_AUTH	M	[7]	Is author of current edit same as previous? (boolean)
REP_EDITOR	R	[8]	Reputation for editor via behavior history
REP_COUNTRY	R	[8]	Reputation for geographical region (editor groups)
REP_ARTICLE	R	[8]	Reputation for article (on which edit was made)
REP_CATEGORY	R	[8]	Reputation for topical category (article groups)
WT_HIST	R	[7]	Histogram of text trust distribution after edit
WT_PREV_HIST_N	R	[7]	Histogram of text trust distribution before edit
WT_DELT_HIST_N	R	[7]	Change in text trust histogram due to edit
DIGIT_RATIO	T	[6]	Ratio of numerical chars. to all chars.
ALPHANUM_RATIO	T	[6]	Ratio of alpha-numeric chars. to all chars.
UPPER_RATIO	T	[6]	Ratio of upper-case chars. to all chars.
UPPER_RATIO_OLD	T	[6]	Ratio of upper-case chars. to lower-case chars.
LONG_CHAR_SEQ	T	[6]	Length of longest consecutive sequence of single char.
LONG_WORD	T	[6]	Length of longest token
NEW_TERM_FREQ	T	[6]	Average relative frequency of inserted words
COMPRESS_LZW	T	[6]	Compression rate of inserted text, per LZW
CHAR_DIST	T	[6]	Kullback-Leibler divergence of char. distribution
PREV_LENGTH	T	[7]	Length of the previous version of the article
VULGARITY	L	[6]	Freq./impact of vulgar and offensive words
PRONOUNS	L	[6]	Freq./impact of first and second person pronouns
BIASED_WORDS	L	[6]	Freq./impact of colloquial words w/high bias
SEXUAL_WORDS	L	[6]	Freq./impact of non-vulgar sex-related words
MISC_BAD_WORDS	L	[6]	Freq./impact of miscellaneous typos/colloquialisms
ALL_BAD_WORDS	L	[6]	Freq./impact of previous five factors in combination
GOOD_WORDS	L	[6]	Freq./impact of "good words"; wiki-syntax elements
COMM_REVERT	L	[7]	Is rev. comment indicative of a revert? (boolean)
NEXT_ANON	!Z/M	[7]	Is the editor of the *next* edit registered? (boolean)
NEXT_SAME_AUTH	!Z/M	[7]	Is the editor of *next* edit same as current? (boolean)
NEXT_EDIT_TIME	!Z/M	[7]	Time between current edit and *next* on same page
JUDGES_NUM	!Z/M	[7]	Number of later edits useful for implicit feedback
NEXT_COMM_LGTH	!Z/M	[7]	Length of revision comment for *next* revision
NEXT_COMM_RV	!Z/L	[7]	Is *next* edit comment indicative of a revert? (boolean)
QUALITY_AVG	!Z/T	[7]	Average of implicit feedback from judges
QUALITY_MIN	!Z/T	[7]	Worst feedback from any judge
DISSENT_MAX	!Z/T	[7]	How close QUALITY_AVG is to QUALITY_MIN
REVERT_MAX	!Z/T	[7]	Max reverts possible given QUALITY_AVG
WT_REPUTATION	!Z/R	[7]	Editor rep. per WikiTrust (permitting future data)
JUDGES_WGHT	!Z/R	[7]	Measure of relevance of implicit feedback

Table 2. Performance of all meta-classifier combinations

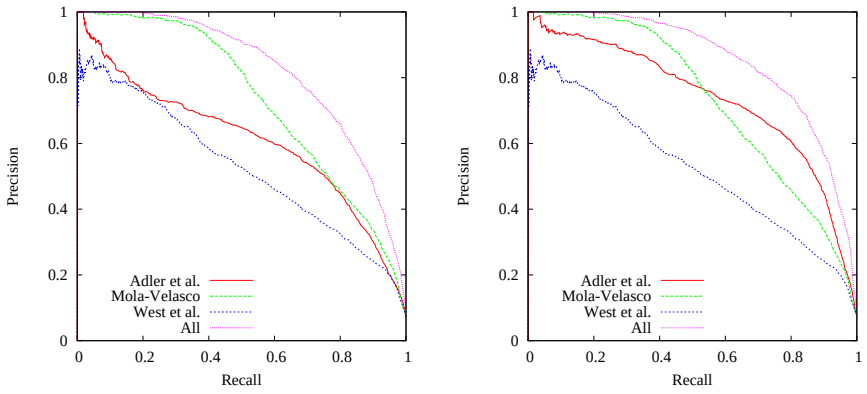
	Immediate		Historic	
Features	**AUC-PR**	**AUC-ROC**	**AUC-PR**	**AUC-ROC**
Adler *et al.*[12]	0.61047	0.93647	0.73744	0.95802
Mola-Velasco[12]	0.73121	0.94567	0.73121	0.94567
West *et al.*[13]	0.52534	0.91520	0.52534	0.91520
Language	0.42386	0.74950	0.58167	0.86066
Metadata	0.43582	0.89835	0.66180	0.93718
Reputation	0.59977	0.92652	0.64033	0.94348
Text	0.51586	0.88259	0.73146	0.95313
M+T	0.68513	0.94819	0.81240	0.97121
M+T+L	0.76124	0.95840	0.85004	0.97590
M+T+R	0.76271	0.96315	0.81575	0.97140
All	0.81829	0.96902	0.85254	0.97620

whether the revision comment for the next edit contains the word "revert" or "rv," which is used to indicate that the prior edit was vandalism [7].

- Both Metadata (M) and Text (T) show impressive gains in going from the *Immediate* task to the *Historic* task. For Metadata, our investigation points to `NEXT_EDIT_TIME` as being the primary contributor, as pages more frequently edited are more likely to be vandalized. For Text, the set of features added in the *historic* task all relate to the implicit feedback given by later editors, showing a correlation between negative feedback and vandalism.
- A surprise in comparing the feature sets is that the predictive power of [M+T] and [M+T+R] are nearly identical in the historic setting. That is, once one knows the future community reaction to a particular edit, there is much less need to care about the past performance of the editor. We surmise that bad actors quickly discard their accounts or are anonymous, so reputation would be useful in the *immediate* detection case, but is less useful in *historic* detection.

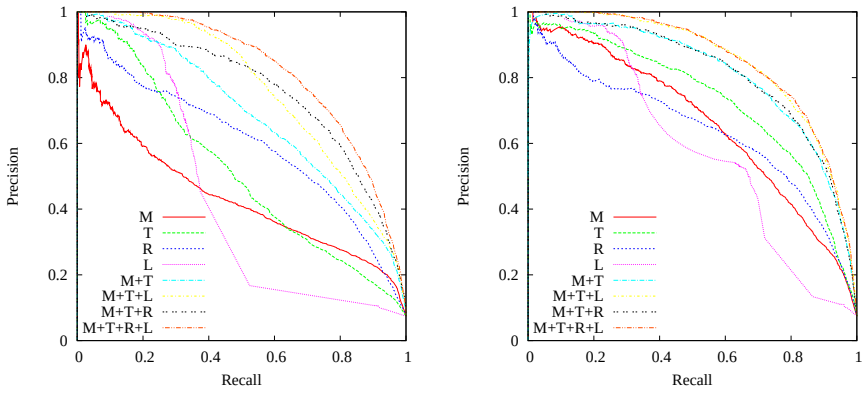
One of the primary motivations for this work was to establish the significance of Language (L) features as compared to other features, because language features are more difficult to generate and maintain for each language edition of Wikipedia. In the case of immediate vandalism detection, we see the interesting scenario of the AUC-PR for [M+T+L] being nearly identical to that of [M+T+R]. That is, the predictive power of Language (L) and Reputation (R) features is nearly the same when there are already Metadata (M) and Text (T) features present. The improvement when all features are taken together is indicative of the fact that Language (L) and Reputation (R) features capture different behavior patterns which only ocassionally overlap.

We chose to use the features of [6] as being representative of a solution focused on Language (L) features due to its top-place performance in the PAN 2010 competition [24]. Yet Figure 2 visualizes that the Language (L) class of features performs only marginally well. Inspection of Table 2 shows that Language (L) features have the worst PR-AUC, but the combined features of [6] have the



(a) Immediate Task (b) Historic Task

Fig. 1. Precision-Recall curves for the three systems and their combination.



(a) Immediate Task (b) Historic Task

Fig. 2. Precision-Recall curves for feature categories.

highest performance. This suggests that the key to the performance beyond the that portion Language (L) features can detect lies in metadata and text features.

5 Conclusions

The success of a machine learning algorithm depends critically on the selection of features that are inputs to the algorithm. Although the previous works on the problem of Wikipedia vandalism detection utilize features from multiple categories, each work has individually focused predominantly on a single category.

We proposed that solving the vandalism detection problem requires a more thorough exploration of the available feature space. We combined the features of

three previous works, each representing a unique dimension in feature selection. Each feature was categorized as either metadata, text, reputation, or language, according to the nature of how they are computed and roughly corresponding to their computational complexity.

We discovered that language features only provide an additional 6% of performance over the combined efforts of language-independent features. This has important ramifications for the development of vandalism detection tools across the other Wikipedia language editions. Moreover, our results outperform the winning system of the PAN 2010 competition, showing that the feature combination explored in this work considerably improves the state of the art (67% vs. 82% AUC). Finally, our meta-classifier could be suitable for the autonomous reversion of *some* bad edits – in a 99% precision setting, 30% recall was achieved.

Acknowledgments. The authors would like to thank Ian Pye of CloudFlare Inc. as well as Insup Lee and Sampath Kannan of the University of Pennsylvania. These contributors were integral in the development of the original/component systems. Additionally, Martin Potthast deserves acknowledgment for his development of the vandalism corpus and for generating interest in the vandalism detection problem. The authors from Universidad Politécnica de Valencia thank also the TIN2009-13391-C04-03 research project. UPenn contributions were supported in part by ONR MURI N00014-07-1-0907. This research was partially supported by award 1R01GM089820-01A1 from the National Institute Of General Medical Sciences, and by ISSDM, a UCSC-LANL educational collaboration. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute Of General Medical Sciences or the National Institutes of Health.

References

1. Wikimedia Foundation: Wikipedia (2010) [Online; accessed 29-December-2010].
2. Wikimedia Foundation: Wikistats (2010) [Online; accessed 29-December-2010].
3. Potthast, M.: Crowdsourcing a Wikipedia Vandalism Corpus. In: Proc. of the 33rd Intl. ACM SIGIR Conf. (SIGIR 2010), ACM Press (Jul 2010)
4. Gralla, P.: U.S. senator: It's time to ban Wikipedia in schools, libraries. `http://blogs.computerworld.com/4598/u_s_senator_its_time_to_ban_wikipedia_in_schools_libraries` [Online; accessed 15-Nov-2010].
5. Olanoff, L.: School officials unite in banning Wikipedia. Seattle Times (Nov. 2007)
6. Mola-Velasco, S.M.: Wikipedia Vandalism Detection Through Machine Learning: Feature Review and New Proposals. In Braschler, M., Harman, D., eds.: Notebook Papers of CLEF 2010 LABs and Workshops, 22-23 September, Padua, Italy. (2010)
7. Adler, B., de Alfaro, L., Pye, I.: Detecting Wikipedia Vandalism using WikiTrust. In Braschler, M., Harman, D., eds.: Notebook Papers of CLEF 2010 LABs and Workshops, 22-23 September, Padua, Italy. (2010)
8. West, A.G., Kannan, S., Lee, I.: Detecting Wikipedia Vandalism via Spatio-Temporal Analysis of Revision Metadata. In: EUROSEC'10: Proceedings of the Third European Workshop on System Security. (2010) 22–28

9. West, A.G.: STiki: A Vandalism Detection Tool for Wikipedia. http://en.wikipedia.org/wiki/Wikipedia:STiki (2010)
10. Wikipedia: User:AntiVandalBot – Wikipedia. http://en.wikipedia.org/wiki/User:AntiVandalBot (2010) [Online; accessed 2-Nov-2010].
11. Wikipedia: User:MartinBot – Wikipedia. http://en.wikipedia.org/wiki/User:MartinBot (2010) [Online; accessed 2-Nov-2010].
12. Wikipedia: User:ClueBot – Wikipedia. http://en.wikipedia.org/wiki/User:ClueBot (2010) [Online; accessed 2-Nov-2010].
13. Carter, J.: ClueBot and Vandalism on Wikipedia. http://www.acm.uiuc.edu/~carter11/ClueBot.pdf (2008) [Online; accessed 2-Nov-2010].
14. Rodríguez Posada, E.J.: AVBOT: detección y corrección de vandalismos en Wikipedia. NovATIca (203) (2010) 51–53
15. Potthast, M., Stein, B., Gerling, R.: Automatic Vandalism Detection in Wikipedia. In: ECIR'08: Proceedings of the 30th European Conference on IR Research. Volume 4956 of LNCS., Springer-Verlag (2008) 663–668
16. Smets, K., Goethals, B., Verdonk, B.: Automatic Vandalism Detection in Wikipedia: Towards a Machine Learning Approach. In: WikiAI'08: Proceedings of the Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy, AAAI Press (2008) 43–48
17. Druck, G., Miklau, G., McCallum, A.: Learning to Predict the Quality of Contributions to Wikipedia. In: WikiAI'08: Proceedings of the Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy, AAAI Press (2008) 7–12
18. Itakura, K.Y., Clarke, C.L.: Using Dynamic Markov Compression to Detect Vandalism in the Wikipedia. In: SIGIR'09: Proc. of the 32nd Intl. ACM Conference on Research and Development in Information Retrieval. (2009) 822–823
19. Chin, S.C., Street, W.N., Srinivasan, P., Eichmann, D.: Detecting Wikipedia Vandalism with Active Learning and Statistical Language Models. In: WICOW '10: Proc. of the 4th Workshop on Information Credibility on the Web. (Apr 2010)
20. Zeng, H., Alhoussaini, M., Ding, L., Fikes, R., McGuinness, D.: Computing Trust from Revision History. In: Intl. Conf. on Privacy, Security and Trust. (2006)
21. McGuinness, D., Zeng, H., da Silva, P., Ding, L., Narayanan, D., Bhaowal, M.: Investigation into Trust for Collaborative Information Repositories: A Wikipedia Case Study. In: Proc. of the Workshop on Models of Trust for the Web. (2006)
22. Adler, B., de Alfaro, L.: A Content-Driven Reputation System for the Wikipedia. In: WWW 2007: Proceedings of the 16th International World Wide Web Conference, ACM Press (2007)
23. Belani, A.: Vandalism Detection in Wikipedia: a Bag-of-Words Classifier Approach. Computing Research Repository (CoRR) **abs/1001.0700** (2010)
24. Potthast, M., Stein, B., Holfeld, T.: Overview of the 1st International Competition on Wikipedia Vandalism Detection. In Braschler, M., Harman, D., eds.: Notebook Papers of CLEF 2010 LABs and Workshops, 22-23 September, Padua, Italy. (2010)
25. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA Data Mining Software: An Update. SIGKDD Explorations **11**(1) (2009)
26. Breiman, L.: Random Forests. Machine Learning **45**(1) (2001) 5–32
27. Davis, J., Goadrich, M.: The relationship between Precision-Recall and ROC curves. In: ICML 2006: Proc. of the 23rd Intl. Conf. on Machine Learning. (2006)