Predicting Bad Edits to Wikipedia Pages

Atul Mirajkar MAT7@UMBC.EDU
UMBC

Abstract

According to the PAN 2010 Wikipedia Vandalism Detection training corpus , about 7% of all revisions were vandalized. This is a significant problem for Wikipedia, because the readers can never be sure of the quality of available information, unless they verify it from other sources. The problem at hand is to identify whether a particular edit made to a wikipedia page is legitmiate and not an attempt to compromise the integrity of the page.

1. Introduction

1.1. Forms of vandalism

Wikipedia being like a free encyclopedia is open to all for edits. And one of its biggest con is edit vandalism. While some edits like huge text removal or addition can be easily itentified other edits like unrelated or garbage information is difficult to spot. Examples of typical vanadalisms are adding irrelevant obscenities, crude jokes which could be humorous at times but definiely need to be reverted back. Types of Vandalism:

- Abuse of Tags: Placing tags such as afd- article for deletion, delete, sprotected etc which do not meet page criteria.
- Link Vandalism: Changing internal links to point to disruptive content.
- Image Vandalism: Uploading ,disruptive images.
- Page Lengthening: Adding large anount of garbage data increasing the load time of a page.
- Spam external links: Adding links to advertising pages.
- Blanking: Removing all important parts of the original page.
- Edit summary Vandalism: Edit summaries are difficult to undo and require admin rights.
- Hidden Vandalism: Any form of vandalism that is not visible in the final document but is embedded in the markup and visible during edits.
- Silly Vandalism: Adding profanity, graffiti or nonsense to pages.

1.2. Problem Idea

PAN 2010 has put forth a corpus of vandalism cases. The corpus compiles 32452 edits on 28468 Wikipedia articles, among which 2391 vandalism edits have been identified. To annotate the corpus Amazon's Mechanical Turk was used by PAN 2010 oraganizers; 753 workers were identified in such a way that each edit was reviewed by at least 3 annotators. The achieved level of agreement was analyzed in order to label an edit as "regular" or "vandalism." Approach is to use supervised learning algorithm to learn a classifier. Features that are used are mainly of 3 classes: Metadata, Text and Language.

2. Related Work

Wikipedia currently has few existing methods for vandalism detection. They are basically of two types.

- First Generation: Regular Expressions, Heuristics, User/IP Blacklist
- Second Generation: Natural Language Processing simple filters

These methods are called bots and are able to detect only upto 30% of vandalism attempts only. Deliberate research in the field of machine learning is taking place in the recent years.

Currently the most effective bots are ClueBot and VoABot. These use regular expressions and lists of

database users and IP addresses blocked to revent vandalism.

Efficient results have been achieved by combining detection rules of STiki, ClueBot, WikiTrust and URL Spam Detection.

Potthast et al.[4] cast the problem as a machine learning one-class classication problem and manually inspect 301 cases of vandalism to create a set of features based in the text and metadata of the edit. These features include the uppercase ratio, frequency of vulgarisms and personal pronouns, size change in the article, whether the editor is anonymous or not, among others.

A completely different approach are reputation systems. West et al.[7] applied the idea of reputation to editors and articles, as well as countries where the editors are. Adler et al. demonstrated that a mixture of user and text reputation and simple metadata features results in good performance. WikiTrust system is used to predict labels based on user reputation features. User reputation increases or decreases depending on the quality of their article revisions (edits). It assumes that quality is directly proportional to the amount of the change that was retained in subsequent revisions. The algorithm also considers reviewer reputation scores, and if the author was anonymous as features (amongst others).

The first systematic review and organization of features appears by Potthast et al.[4] as part of the PAN 2010 Evaluation Lab. The authors conclude their analysis by building a classier using the predictions of the nine participants in the competition. This classier performed significantly better than the best single participants, which suggests that the success in vandalism detection relies on the combination of a wide variety of features from all approach: content, metadata and reputation.

3. Proposed Method

An article from its birth undergoes a series of revision edits and is called as article history. A single revision is a state of the article at a given time in its history and is composed of the textual content, markup and metadata describing the transition from the previous history. A revision metadata contains the user who performed the edit, a comment explaining why the edit was made and describing the changes made to the article. A revision edit can be considered as a tuple containing the previous edit, the new edit and its corresponding metadata. The goal of the classifier is to output whether a particular edit made is good (no vandalism) or bad(vandalism).

The proposed method is to preprocess both (previous and new) revision edits pertaing to a single edit to remove the extra wiki markup and get pure text. Get the difference in the pure texts and compute metadata, text and language features which are then fed to the classifier.

3.1. Pre-Processing

Wikipedia edits and formatting is done using mark ups. These mark ups are not only HTML tags but wiki specific formatting. Special formatting is used to denote links, image URLs, tables, section headers, references, alingnment information. Steps taken to get plain text from which fetaures can be extracted.

1. Replace HTML tags with spaces
2. Remove mark up used for fonts ””’/””””
3. Remove mark up used for headings ”==/===/====...”
4. Remove mark up used for numbered Lists ”#/##” , simple Lists ”*/**...” indented text ”:/::..”
5. Remove mark up used for pre formatted text ”..”, tables ”——”
6. Replace all URLs with ”URL:*url*”
7. Replace all alphanumeric characters with ”ALPHANUMERIC:*alphanumeric*”
8. Replace all numbers with ”NUMBER:*number*”

Steps 6,7,8 were necessary for feature extractions based on URLs and non language statistics.

3.2. Features

The proposed method is to extend Potthast et al.[4] approach and define and a set of language dependent and independent features.Approach is to use supervised learning algorithm to learn a classifier. Features that will be used are mainly of 3 classes: Metadata, Text and Language. Proposed features that can be used are stated below.

3.2.1. Metadata Features

- ISREGISTERED: marks if the author of the edit has a Wikipedia account.

- COMMENTLENGTH: the length of the edit revision.
- SIZECHANGE: length difference between the new and old revisions.
- SIZERATIO: ratio between the new and old revisions text length.
- PREVSAMEAUTH: if the old revision has the same author as the new one.

3.2.2. Text Features:

Extraction of features based on the inserted edit text. These features help in detecting random word inserts, silly vandalism.

- DIGITRATIO: the frequency of digits in the new revision.
- ALPHANUMRATIO: the frequency of alphanumeric characters in the new revision.
- UPPERRATIO: the frequency of upper case characters in the new revision.
- UPPERLOWERRATIO: ratio between the upper case and lower case characters in the new revision.
- LONGCHARSEQ: longest single character sequence length.
- LONGWORD: longest word length.
- COMPRESSLZW: compression ratio of added words .
- PREVLENGTH: the text length of the previous revision.
- WORDDIFFCOUNT: vector of difference in counts of all words before and after edit .

3.2.3. Language Features:

Language features helps the classifier in understanding some sort of semantics. Example using a dictionary of "Pronouns" helps detect self promotion which is a strong indicator of vandalism. External dictionaries assist the use the following features.

- VULGARITY: the frequency of vulgar words.
- PRONOUNS: the frequency of first and second person pronouns.
- BIASEDWORDS: the frequency of high bias words.
- MISCBADWORDS: the frequency of any other words with negative meaning (or not suitable for an encyclopedia)
- ALLBADWORDS: the frequency of all bad words (vulgar, pronouns, biased, sexual and miscellaneous)
- GOODWORDS: the frequency of words that are not bad
- COMMREVERT: if the new revision comment marks that previous changes were reverted to an earlier state.

3.2.4. Intuition

Metadata features help detect authencity of the editor. Most vandalisms come from anonymous users and so this feature plays an important role. Also other metadata features like size change of the revision help identify vandalism. Another feature to look at is the comment length. Most of the times if the intentions are bad, the editor would be least concerned of the comment length and on an average the comment length observed for bad edits is small. This class of features is also important because the feature computation is least expensive since the data is instantly available.

Text features help extract information of the inserted text. Vandalism edits like repeating characters, entering special characters, entering large sequence of meaningless characters can be detected using text features. Text features also try extract information like character diversity for an edit. These features help identify silly vandalisms as they are the most difficult ones to identify. Computation wise this class of features is expensive because it involves getting the difference between the revisions and also requires iterating on the diffed text.

Language features help classifier understand semantics of the added text. Vulgar terms and slang words feature help in detecting abusive language and personal attacks. The pronoun list feature tries to identify self promotion which is another kind of vandalism hard to detect. This class of features is also computationally expensive as it requires iterating over the natural language dictionaries to get counts of slang/vulgar/pronoun words.

3.3. Experiments and Results

The corpus contains 32452 edits among which 2391 are vandalism edits. For a classification setting the plan is to use 10 fold cross validation on training data and report precision, recall, F-measure, area under precision recall curve, area under receiver operating characteristic curve. As the dataset is highly skewed estimating the performance on the basis of accuracy is not a good idea.

Performance measures such as precision, recall and F-measure are computed from the following values.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$FP - Rate = \frac{FP}{FP + TN}$$

Where: TP : Number of edits that are correctly identified as vandalism edits (True Positives).
FP : Number of edits that are wrongly identified as vandalism edits (False Positives).
TN : Number of edits that are correctly identified as non vandalism edits (True Negatives).
FN : Number of edits that are wrongly identified as non vandalism edits (True Positives).

Area under the precision recall curve will be used as a true performance measure for a given classifier.

3.3.1. Evaluation for vandalism class with different classifiers

Cl	P	R	F	AUCPR	AUCROC
NB	0.303	0.433	0.357	0.316	0.836
KNN	0.586	0.254	0.354	0.403	0.846
SVM	0.857	0.1	0.177	0.425	0.549
ANN	0.711	0.151	0.249	0.428	0.870
RF	0.719	0.503	0.592	0.606	0.879

Cl - Classifier
P - Precision
R - Recall
AUCPR - Area under Precision Recall curve
AUCROC - Area under Receiver Operating Characteristic curve
NB - Naive Bayes
KNN- K Nearest Neighbors
SVM - Support Vector Machine
ANN - Artificial Neural Network
RF - Random Forest

- Naive Bayes:
  As we can see from the comparison table, area under Precision Recall Curve for Naive Bayes is the least. The main reason for the low AUC-PR is attributed to the dataset being skewed. The prior probability of an edit not being vandalized is almost 93% and so values of precision nad recall are very low.
- Support Vectors Machines and Artificial Neural Networks:
  Most of the edits were blanks or the difference in the previous and new revision was very small. As a result values for most feature values were very small. This resulted in wrong classification and low values for precision and recall for NB and ANN but values were better than that for Naive Bayes.
- k Nearest Neighbors:
  KNN gave better values for precision and recall compared to NB and SVM.
- Random Forest
  The precision and recall values for Random Forest were the best among all classifiers. The reason is bagging approaches tend to work well when the dataset is skewed. Multiple random trees participate in the classification. Also as the number of trees increase, the correlation between different features is better learnt and hence the accuracy is the best.

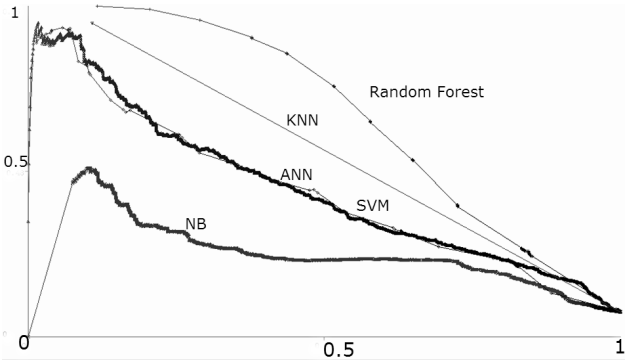

Fig:*Area Under Precision Recall Curve for different classifiers*

3.3.2. Different No. Of Trees for Random Forest

As the best performance is achieved using Random Forest we further evaluate its performance using different number of Random Forests.

Class	P	R	F	AUCPR	AUCROC
10	0.719	0.503	0.592	0.606	0.879
20	0.776	0.506	0.612	0.643	0.901
30	0.794	0.491	0.607	0.659	0.910
50	0.810	0.484	0.606	0.672	0.918
70	0.821	0.491	0.614	0.679	0.923

As we increase the number of trees the precision goes on increasing but the recall goes on decreasing. The best performance is provided by random forest with 20 trees.

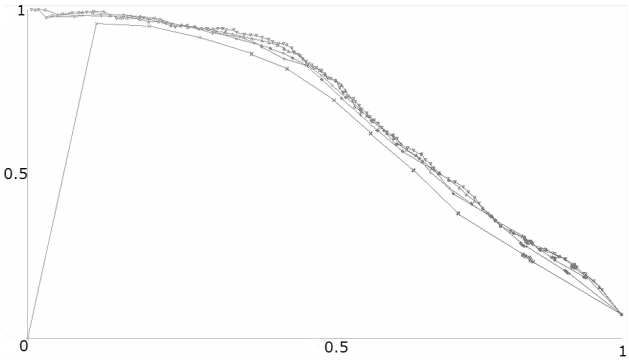

Fig:*Area Under Precision Recall Curve for different Random Forests with different trees*

3.3.3. Feature Set Evaluation on Random Forest with 20 trees

Further we evaluate the features which provide the most information towards classification of vandalism edit.

Class	P	R	F	AUCPR	AUCROC
Meta(M)	0.546	0.344	0.422	0.412	0.828
Text(T)	0.578	0.296	0.391	0.386	0.774
Language	0.655	0.187	0.291	0.322	0.719
M+T	0.707	0.461	0.558	0.582	0.896
All	0.776	0.506	0.612	0.643	0.901

3.4. Softwares to be tested on

- Weka 3: Data Mining Software in Java
- LIBSVM – A Library for Support Vector Machines

4. Conclusion and Future Work

Language Features provide the least information gain. It is expected that language features would provide the maximum information gain. But the problem is if anyone wants to vandalize a page, he or she would not care to spell the words correctly and so in most cases vulgar/slang dictionaries fall short identifying the bad words.

If we run Random Forest Classifier with 20 trees on different classes of features individually, it is seen that metadata features provide the maximum information gain. Metadata featuires like comment length and anonymous user have the most information gain towards classification. But the maximum performance is provided using all classes of features together.

Bagging methods like Random Forest are a better approach for classification of unbalanced classes or skewed classes. Random forests do a better job identifying correlation between features.

Till now we have only considered features that can only be extracted from the corpus. Also we have seen that metadata features provide a better information gain compared to other classes of features. Other metadata features (not from corpus) need to be identified that can be used and experimented to improve accuracy.

- Reputation of the editor: The information of the editor of a particular revision can be used. Information like what kind of articles the editor has edited in the past. Frequency of edits the editor has made in the past.
- GeoLocation of the editor: Geographical location of the editor and the article class to which the edit was made can be learnt to extract geo-spatial information
- Comment Revert: If a particular edit comment was a "Revert", we are pretty sure the previous edit was a bad edit

5. References

1. B Adler et al, Detecting Wikipedia Vandalism using Wikitrust, Padua, 2010. In M. Braschler and D. Harman, editors, Notebook Papers of CLEF 2010 LABs and Workshops, 22-23 September, Padua, Italy, Sept. 2010.
2. Si-Chi Chin et al, Detecting Wikipedia Vandalism with Active Learning and Statistical Language Models, in WICOW'10, Raleigh, NC, USA, 2010.
3. Chin, Si-Chi, W. Nick Street, Padmini Srinivasan, and David Eichmann. Detecting Wikipedia van-

dalism with active learning and statistical language models." Proceedings of the 4th workshop on Informa- tion credibility. ACM, 2010.

4. Potthast, Martin. Crowdsourcing a Wikipedia Vandalism Corpus. In Hsin-Hsi Chen, Efthimis N.Efthimiadis, Jaques Savoy, Fabio Crestani, and Stephane Marchand-Maillet, editors, 33rd International ACM Conference on Research and Development in Information Retrieval (SIGIR 10), pages 789-790, July 2010. ACM. ISBN 978-1-4503-0153-4.

5. Velasco, Santiago M. Mola. Wikipedia vandalism detection through machine learning: Feature review and new proposals." Lab Report for PAN-CLEF 2010 (2010).

6. Smets, K., Goethals, B., Verdonk, B.: Automatic Vandalism Detection in Wikipedia: Towards a Machine Learning Approach. Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy (WikiAI08) (2008)

7. West, A. G., Kannan, S., Lee, I.: Detecting Wikipedia Vandalism via Spatio-Temporal Analysis of Revision Metadata. Technical Reports (CIS), University of Pensylvania, Department of Computer and Information Science. (2010)

8. Adler, B., de Alfaro, L., Mola-Velasco, S. M., Rosso, P., West, A.: Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features.Computational Linguistics and Intelligent Text Processing, University of California, Santa Cruz, USA (2011)

9. Adler, B. T., and de Alfaro, L. 2007. A Content-Driven Reputation System for the Wikipedia. In Proceedings of the 16th International World Wide Web Conference

10. Smets, K., Goethals, B., Verdonk, B.: Automatic Vandalism Detection in Wikipedia: Towards a Machine Learning Approach. Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy (WikiAI08) (2008)

11. Gabrilovich, E., and Markovitch, S. 2007. Computing Semantic Relatedness using Wikipedia-Based Explicit Semantic Analysis. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI), 16061611.

12. Cluebot. NG, Wikipedia user ClueBot NG, 2012. [Online]. Available: http://en.wikipedia.org/wiki/User:ClueBot_NG.

13. A. G. West. STiki: A vandalism detection tool for Wikipedia. http://en.wikipedia.org/wiki/Wikipedia: STiki, 2010. (Software tool applying, in real-time, the spatiotemporal features described herein).

14. Vandalism, Wikipedia, 2012. [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:Vandalism#Types_of_vandalism.

15. University of Waikato, Weka.

16. Wikipedia. http://www.wikipedia.org.