Detecting Wikipedia Vandalism with Active Learning and Statistical Language Models

Si-Chi Chin
Interdisciplinary Graduate Program in Informatics (IGPI)
The University of Iowa
Iowa City, IA 52242, USA
si-chi-chin@uiowa.edu

W. Nick Street
Management Sciences Department & IGPI
The University of Iowa
Iowa City, IA 52242, USA
nick-street@uiowa.edu

Padmini Srinivasan
Computer Science Department & IGPI
The University of Iowa
Iowa City, IA 52242, USA
padmini-srinivasan@uiowa.edu

David Eichmann
Institute of Clinical and Translational Science & IGPI
The University of Iowa
Iowa City, IA 52242, USA
david-eichmann@uiowa.edu

ABSTRACT

This paper proposes an active learning approach using language model statistics to detect Wikipedia vandalism. Wikipedia is a popular and influential collaborative information system. The collaborative nature of authoring, as well as the high visibility of its content, have exposed Wikipedia articles to vandalism. Vandalism is defined as malicious editing intended to compromise the integrity of the content of articles. Extensive manual efforts are being made to combat vandalism and an automated approach to alleviate the laborious process is needed.

This paper builds statistical language models, constructing distributions of words from the revision history of Wikipedia articles. As vandalism often involves the use of unexpected words to draw attention, the fitness (or lack thereof) of a new edit when compared with language models built from previous versions may well indicate that an edit is a vandalism instance. In addition, the paper adopts an active learning model to solve the problem of noisy and incomplete labeling of Wikipedia vandalism. The Wikipedia domain with its revision histories offers a novel context in which to explore the potential of language models in characterizing author intention. As the experimental results presented in the paper demonstrate, these models hold promise for vandalism detection.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms

Experimentation, Human Factors, Languages, Security

Keywords

Wikipedia, vandalism, statistical language models, active learning, classification

WICOW'10, April 27, 2010, Raleigh, North Carolina, USA.

1. INTRODUCTION

Wikipedia, the online encyclopedia, is a popular collaborative information system. As a collaborative space for any individual to edit articles, Wikipedia is also prone to malicious editing – vandalism. Wikipedia defines vandalism as "any addition, removal, or change of content made in a deliberate attempt to compromise the integrity of Wikipedia[1]." Measures to combat vandalism are extensively discussed on Wikipedia and individual task forces and studies were created for this purpose . Wikipedia has taken many measures to address the challenges of vandalism, such as restricting the privileges of anonymous users, adopting "article validation" and using an "abuse filter" to control user activities by reacting automatically to suspicious user behaviors.

Currently active tools to fight vandalism include ClueBot and VoABot II. The two anti-vandal bots provided an automatic solution to detect and revert vandalism edits. There, however, exists opportunity for improvement. Research [14, 11] has shown that the current bots were limited in their extensibility as well as in their effectiveness at detecting instances of committed vandalism. Therefore, exploring additional automated measures to improve the accuracy of the vandalism detection carries numerous benefits. First, it helps alleviate manual effort required for cleaning vandalism edits. Second, it helps identify automated solutions to address the weakness of the current tools. Finally, an effective anti-vandalism tool could prevent or correct future

[1]http://en.wikipedia.org/wiki/Wikipedia:Vandalism

malicious editing – thus protecting the integrity of Wikipedia articles.

In this paper, we use the outputs of statistical language models as predictive features, and use them to train classifiers in an active learning framework to classify potential vandalism instances. Moreover, we develop a Wikipedia editing taxonomy and provide a more comprehensive categorization of vandalism instances. We work on the complete revision history of the "Microsoft" and "Abraham Lincoln" articles, which are listed among the most vandalized articles on Wikipedia. In addition, we manually inspect and label experiment results to provide a more complete and accurate description of vandalism instances. Contributions of this paper are:

- We provide a taxonomy structure for editing actions on Wikipedia, and categorize types of vandalism.
- We analyze the technical difficulties of identifying vandalism instances in each category.
- We explore a novel application of language models, i.e., to indicate vandalism instances. We build statistical language models using the CMU-toolkit [3] and test the models with a both the newly edited revision and the difference between consecutive revisions (*diff*).
- We build content-based classifiers using the statistics generated from language models as features to provide an accurately ranked list of potential vandalism instances. The classifier is effective without using information regarding contributors.
- We adopt active learning models to learn iteratively the most probable vandalism instances, minimizing the manual efforts to annotate a training set.

The paper is structured as follows. In Section 2, we describe the taxonomy of Wikipedia actions and the categorization of vandalism. In Section 3, we describe the data sets used for our experiments, and detail the implementation of the system, including the system framework, statistical language models and classification approach. In Section 4 we present our experimental results. In Section 5, we review previous academic research on Wikipedia vandalism and revision history. In Section 6, we conclude the paper and discuss the opportunities for future work.

2. TYPES OF VANDALISM

In this section, we present a preliminary categorization of vandalism based on the action taxonomy of Wikipedia. The basic actions include delete, insert, change, and revert. A revert occurs to correct vandalism, edits without proper reference, edits for testing, or due to the development of edit wars. Actions of delete, insert, and change involve the content and the formatting of articles. The content class includes text, images, and links; the formatting includes HTML tags or CSS, and Wikipedia templates. Figure 1 illustrates our current taxonomy of actions on Wikipedia.

Previous research has identified many common types of vandalism. Viégas et al. [15] identified five common types of vandalism: mass deletion, offensive copy, phony copy, phony redirection, and idiosyncratic copy. Priedhorsky et al. [12] categorized Wikipedia damaged edits[2] into seven types: misinformation, mass delete, partial delete, offensive, spam, nonsense, and other. The categories proposed in these papers were not developed systematically, and can be made more comprehensive. Potthast et al. [11] organized vandalism edits according to the "Edit content" (text, structure, link, and media) and the "Editing category" (insertion, replacement, and deletion). This organization does not consider the scale of editing, which correlates with the difficulty level of vandalism detection. Usually, as a Wikipedia article evolves and stabilizes, a large-scale editing is likely to be malicious and will be reverted by the following revision.

To categorize vandalism instances, we introduce a systematic taxonomy (Figure 1) that differs from the previous works in the following ways:

- The categorization is systematically organized based on the four primary actions (change, insert, delete, and revert) and types of change (format and content).
- The scale of editing is assessed in the taxonomy. It is more intuitive to detect a malicious large-scale editing (e.g. massive deletion of content) as opposed to a minor revision (e.g. removal of a character from a word).
- The taxonomy illustrates the relationship between a vandalism instance and a legitimate action.

In this paper, we elaborate on common types of vandalism (Table 1), building classifiers based on language model statistics to detect vandalism instances of type "Blanking," "Large-scale Editing," "Graffitti," and "Misinformaiton." We elaborate on technical challenges and the limit of rule-based systems for these categories elaborated as follows.

Blanking and Large-scale Editing can be detected by examining the *diff* result of two consecutive revisions. In our experiments, we categorized a revision as an instance of blanking if the new revision was at least 90% smaller than the average length of the page. We defined a revision as an instance of large-scale editing if the size of new edits (insertion and change) was twice as large as the median value[3] of the length of all edits in the previous diff history.

Graffiti is an insertion of unproductive, irrelevant, random, or unintelligible text. Some instances of graffiti can be identified with manually crafted rules such as setting a threshold to the ratio of upper-case letters or the maximum length of a word in the new edit. Hand-crafted rules that detect a large portion of upper case letters could filter out vandalism instances such as "I LOVE MAC!"; similarly, rules that threshold the maximum word length may detect a long sequence of meaningless letters such as "daewiatlgkjdflkgsyhgfaw". However, graffiti such as "I like eggs." or "John loves Jane" would not be discovered by these rules or captured by the sixteen features used in the work of Potthast et al. [11].

[2]Although damage edits were not referred to as vandalism in their work, they were in fact in line with the definition of Wikipedia vandalism.

[3]Thresholds of 90% for blanking and twice the median for large-scale editing were chosen empirically based on the authors' experience. Further empirical studies may determine more discriminating values.

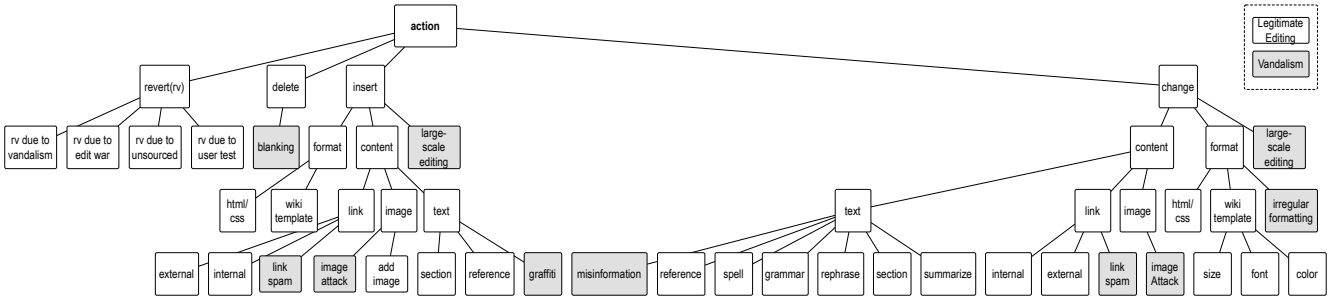

Figure 1: Wikipedia Action Taxonomy

Table 1: Types of Vandalism

Type	Action Taxonomy	Definition	Example
Blanking	Delete(massive)	Delete the entire article or a massive amount of existing content.	
Large-scale Editing	Insert (massive) Change (massive)	Add a massive amount of malicious text to lengthen the article to slow the loading speed or change a massive portion of the existing content.	Replace all the occurrences of "Microsoft" to "Microshaft".
Graffiti	Insert–Text	Insert completely irrelevant, random, or unintelligible text, including the usage of profanity or other vocabulary or phrases to express anger, and adding contents only remotely related to the subject or fruitless comments that undermine the quality of the article.	• *I like eggs!* • *dfdfefefd jaaaei #$%&@@#* • *John Smith loves Jane Doe.* • *This ***king program is EVIL!!!* • *Buying their computers is totally a waste of your money.*
Misinformation	Change–Text	Replace existing content with false information such as changing named entities (e.g. personal names, locations, and product names etc..) It usually occurs when vandals attack the information box (brief summary box on the left of the page). The changes often appear in a nearly indiscernible manner, such as changing the spelling of words, deleting one or more digits for numbers, or inverting a positive statement to negative.	• *Key Person: John Lennon* (on Microsoft page) • *4,600 million → 4,000 million* • *This is true → This is not true*
Image Attack	Insert–Image Change–Image	Replace existing image with an irrelevant one, or insert one to many images, so as to damage the page.	Replace Microsoft logo with a picture of a kitten.
Link Spam	Insert–Link Change–Link	Insert external or internal links which are irrelevant to the article	• *http://www.wierdspot.com Abe's Personal Diary*
Irregular Formatting	Insert–Format Change–Format	Insert HTML or CSS format tags that are not standard to the editing guideline; change the format of existing texts or images to damage the appearance of the article.	• Inappropriate use of Wikimarkup such as *{{nonsense}}*

While one may generate more rules to detect a list of commonly-used vandalism language vocabulary (e.g. profanity, slang, unintelligible words etc.), it is difficult to maintain the list as the vandalism language may evolve over time. Moreover, some usage of profanity vocabulary is justifiable based on the context. For example, the phrase "VISTA IS AN EVIL SOFTWARE!!" would be a vandalism instance on the Microsoft article; however, the sheer occurrence of the word "evil" is not a good indicator for detecting vandalism, as in another context such as "good or evil, it would depend on users", it becomes part of a valid edit. Therefore, a rule-based filtering system to detect this type of vandalism is neither extensible nor easy to maintain.

MISINFORMATION usually involves changing the existing named entities (e.g. personal names, locations, and product names), the spelling of words, deleting one or more digits for numbers, or inverting a positive statement to negative. Vandalism in this category conducts changes at a micro level to deceive human perceptions. It usually occurs when vandals attack the information box (brief summary box on the left of the page). Although the automated *diff* processing would identify subtle revisions, it is challenging to create rules to differentiate a valid correction of typos or grammar from a malicious subtle revision. A rule-based system may compile a list of known named entities, using an automated named entity recognizer (NER) to track the occurrence of unforeseen named entities. However, maintaining such a list is a non-trivial task and its effect would not be evident if the NER has limited performance.

3. VANDALISM CLASSIFICATION

3.1 Data and Experimental Setup

We worked with the Wikipedia page history archive from February 24th, 2009[4]. Our corpus includes complete revision histories (note this aspect is unique to our research) for two Wikipedia articles: Abraham Lincoln (8,816 revisions), Microsoft (8,220 revisions). These articles are acknowledged to be among the most vandalized pages[5]. The reason for choosing the most vandalized pages is to acquire an extensive amount of vandalism instances for the analysis. We intentionally chose one article from the "Computing and Internet" category and one article from the "History" to demonstrate the similarity and differences of the vandalism pattern across categories.

Figure 2 illustrates the system structure and preprocessing of the revision history. We extracted the two articles from the Wikipedia Dump file and parsed them into individual revisions with the SAX parser. Information such as revision comments, contributors, and timestamp are also extracted from the XML file. We used the Java BreakIterator class to preprocess the revision history. Each revision was processed into one sentence per line to enable diff processing at the sentence level.

We used the CMU-toolkit [3] to build bigram statistical language models for each revision of a page. Moving through the sequence of revisions we adopt the following process. Assuming we are at revision n we compute the diff between it and the previous version *n-1*. This diff is directional in that we record only the new data that is in version n as compared to version *n-1*. The diff data for revision n and the full revision n are then tested using the built model.

[4]http://download.wikimedia.org/enwiki/latest/

[5]http://en.wikipedia.org/wiki/Wikipedia:Most_vandalized_pages

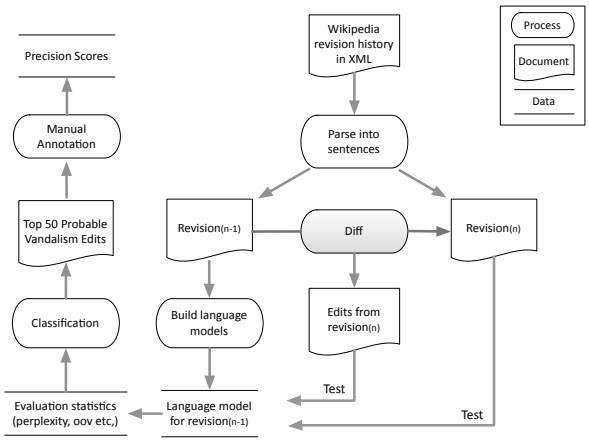

Figure 2: Flowchart of experiments.

Each test yields a set of values: perplexity, number of words, number of words that are out of vocabulary, percentage of words that are out of vocabulary, number of bigrams hits and unigram hits, and percentage of bigram and unigram hits.

As vandalism often involves the use of unexpected vocabulary (the "out-of-vocabulary" number from CMU-toolkit *evallm* process) to draw attention, an instance of vandalism would produce high surprise factor when compared with the previous version, i.e., it would produce high perplexity when assessed using the language model of the previous version. Since we built a language model for every individual revision, including vandalized revisions, a follow up revision to revert a vandalism would also have high perplexity compared to the previous vandalism instance. To address the challenge and to identify a non-vandalized revision for the evaluation, we evaluate each diff result and the new added revision n against three language models: the model built from the revision *n-1*, the revision *n-5*, and the revision *n-10*. [6] We would expect an instance of vandalism to have three large out-of-vocabulary results, and a revert to have only one large out-of-vocabulary number. Therefore, from the three results, we select the one with the lowest out-of-vocabulary number, so as to avoid mistaking a legitimate revision for a vandalism instance.

3.2 Statistical Language Models and Classification

Statistical language modeling (SLM) [13] computes the distribution of tokens in natural language text and assigns a probability to the occurrence of a string S or a sequence of m words. SLM is commonly applied to many natural lan-

[6]The choice of *n-5* and *n-10* is based on authors' experiences. It is not uncommon that a vandalism action occurs consecutively. If a vandalism occurs at the revision *n-1*, it is likely that the revision *n-2* or *n-3* is also a vandalism instances. Meanwhile, as the language evolves over time, we want to use an old revision that is still similar enough to the current revision. Experiences showed that using the revision *n-5* and *n-10* demonstrated an adequate results.

Table 2: Definition of Features

Feature	Definition
word_num(d)	Number of known words (from *diff*)
perplex(d)	Perplexity value (from *diff*)
entropy(d)	Entropy value (from *diff*)
oov_num(d)	Number of unknown words (from *diff*)
oov_per(d)	Percentage of unknown words (from *diff*)
bigram_hit(d)	Number of known bigrams (from *diff*)
bigram_per(d)	Percentage of known bigrams (from *diff*)
unigram_hit(d)	Number of known unigrams (from *diff*)
unigram_per(d)	Percentage of known unigrams (from *diff*)
ratio_a	Ratio of added text from previous revision
ratio_c	Ratio of changed text from previous revision
ratio_d	Ratio of deleted text from previous revision

Table 3: Classification Comparison

Classifier	Precision	Recall	F-measure
Baseline	0.860	0.845	0.850
Boosting J48	0.945	0.854	0.897
Logistic	0.876	0.774	0.822
SVMs	0.869	0.774	0.819

guage processing tasks such as speech recognition , machine translation , text summarization , information retrieval , and web spam detection [10, 2]. The CMU SLM toolkit [3] allows construction and testing of n-gram language models. The *evallm* tool evaluates the language model dynamically, providing statistics such as perplexity, number of n-grams hits, number of OOV (out of vocabulary), and the percentage of OOV from a given test text. In our experiments, we built bigram language models with the Good-Turning smoothing method [3].

We used two sets of *evallm* statistics results that were generated separately from the diff data for the new revision and the full new revision to build classifiers. In addition to the 18 attributes (9 for each set) generated from SLM, three features: ratio of insertion, ratio of change, and ration of deletion, were added to the set of attributes. We summarize features for the classification in Table 2.

We used the Weimar data from Potthast et al. [11] as the baseline to evaluate our features and classification methods. This data includes pairs of consecutive edits from different articles, some of which are vandalism instances. All instances are labeled, allowing a full evaluation of classification accuracy. We used Weka to train classifiers and evaluated them with 10-fold cross-validation. As shown in Table 3, boosting with J48 decision trees using our features dramatically outperformed the baseline performance from [11], and both logistic regression and SVMs also achieved better precision than the baseline. The results demonstrate the effectiveness of our features and the potential of three classification methods. However, although boosted decision trees achieved the best performance, the method fails to provide an adequate probability distribution to rank the results. Conversely, both logistic regression and SVMs provide satisfactory probability distributions to allow for an accurate ranked list. Therefore, we used logistic regression and SVMs to in our experiments with Wikipedia revision history.

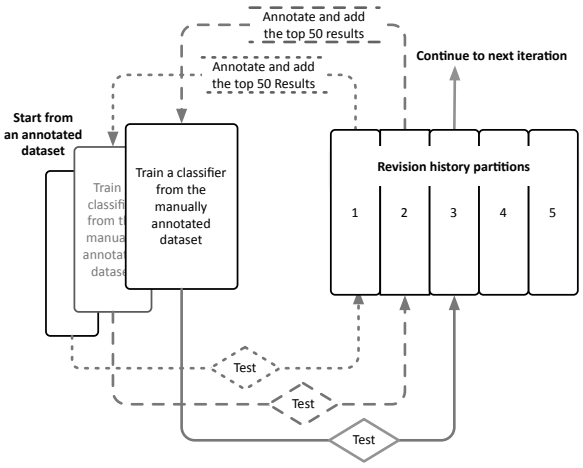

Figure 3: Active Learning Models

3.3 Active Learning Models and Annotation

Vandalism instances are not systematically archived by Wikipedia. Previous research [7, 12] typically uses regular expressions matched against revision comments to label vandalism, matching any form of the word "vandal" and "rvv" ("revert due to vandalism"). Studies using this labeling approach showed that vandalism only composed a small portion of edits (1-2%) and was fixed relatively quickly (the mean survival time was 2.1 days, with a median of 11.3 minutes). However, matching against comments is insufficient as vandalism is usually corrected without comments. Moreover, in the case of dual vandalism, in which a user vandalized two or more consecutive revisions and reverted only the last vandalism revision to mislead stewards that the vandalism had been corrected, revision comments were no longer accurate indicators for vandalism instances. Hand-labeling thousands of Wikipedia revisions to obtain an accurate training data is labor intensive. We use a supervised active learning model to address this challenge.

Research [9] has shown that supervised active learning benefited situations in which labeled training data is sparse and obtaining labels is expensive. In our experiments, we iteratively built classifiers that incorporated the highest-ranked samples from the Wikipedia revision history to detect and rank future vandalism instances. We started with the annotated data provided by Potthast et al. [11] and used it as the baseline dataset. We then divide a revision history into five partitions chronologically. In the first iteration, we built a classifier using the baseline data and tested it on the first partition. The classifier produced a ranked list, and the top 50 results were annotated and added to the existing training pool to build a new classifier for the next iteration. Figure 3 illustrates three iterations of active learning.

The annotation process involved labeling whether a revision is a vandalism instance and which type of vandalism it is. An annotator is provided a ranked list of 50 probable vandalism revision identifiers. The annotation interface linked each retrieved identifier to a diff view provided by Wikipedia[7]. An annotator judged from the newly edited content to determine if it is a vandalism instance. An an-

[7]http://en.wikipedia.org/w/index.php?diff=prev&oldid=(id)

Table 4: Logistic and SVM Overlap Ratio

	Iteration				
Data	**1**	**2**	**3**	**4**	**5**
Microsoft	0.44	0.29	0.33	0.39	0.54
Lincoln	0.22	0.47	0.25	0.55	0.14

notator also made the judgement by examining whether the revision was reverted by the next revision[8].

4. EXPERIMENT RESULTS

4.1 Classifiers Performance

Our aim is to classify vandalism instances, providing an accurate ranked list of potential vandalism occurrences. We used a supervised active learning model, learning from the best samples for each five iterations, to minimize manual effort for the annotation. We used the average precision at 50 revisions that were ranked by classifiers as the most probable vandalism instances to evaluate the performance. Our experiments used two classifiers: logistic regression and SVMs, and worked on two revision histories: "Microsoft" and "Abraham Lincoln".

Figure 4.1 shows that logistic regression achieved the highest average precision of 0.81 at the 4th iteration for the "Microsoft" dataset and at the 3rd iteration for the "Abraham Lincoln" dataset. SVMs achieved .68 and .76 respectively to "Microsoft" and "Abraham Lincoln" at the third iteration. Both datasets exhibit a climb of average precision from the first to third iteration for either logistic regression or SVMs. The non-monotonic results imply that the underlying distribution of vandalism instances and types varied as a Wikipedia article evolved. One explanation for the decline of the average precision in the last two iteration is the introduction of new templates, Wikimarkups, and language links in the later revisions. For example, the insertion and deletion of tags such as {{sprotect}}, {{toolong}}, and {{spilt}} occurred more frequently as the Wikipedia article evolved. Inserting any unseen new tags would increase the perplexity of the current revision and consequently create more false positive instances. Another possibility is that the actual number of vandalism instances decreased in the later revisions.

Our experimental results show that logistic regression and SVMs identified different vandalism instances. Table 4 is a tabular view of the overlapping ratio (the intersection over the union) of the two classifiers. This characteristic is most evident at the third iteration for both "Microsoft" and "Abraham Lincoln" data. While both classifiers achieved equivalently high performance, they only overlapped for 0.33 and 0.25 respectively to "Microsoft" and "Abraham Lincoln" data. This, along with the boosting tree results, points to the potential of using ensemble methods for this task.

We observe that classifiers trained from the baseline data can achieve satisfactory performance on the "Microsoft" and "Abraham Lincoln" data. It indicates the potential of training classifiers from heterogeneous sources to use on data from other domains.

[8]http://en.wikipedia.org/w/index.php?diff=next&oldid=(id)

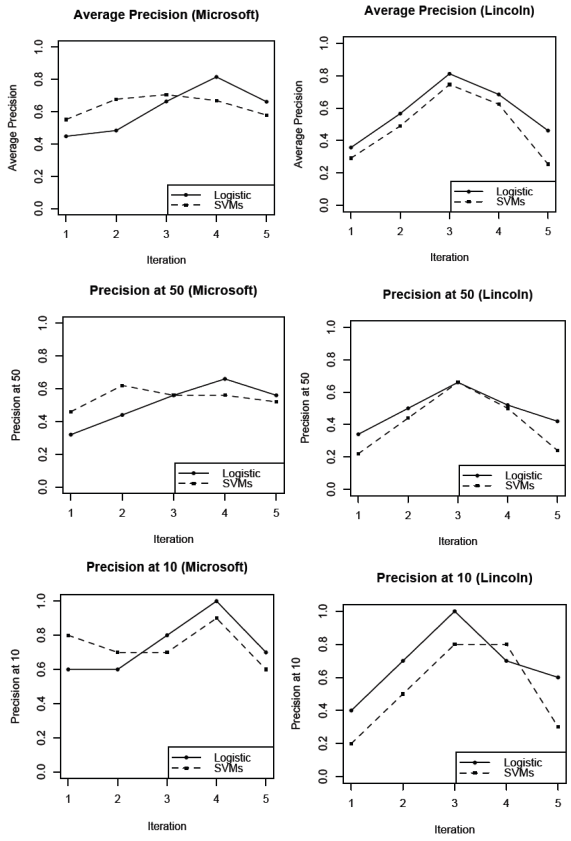

Figure 4: Experimental Results for Active Learning

4.2 Vandalism Types Analysis

Tables 5 show the distribution of types of vandalism for two classification methods. Results show that both logistic regression and SVMs are robust in detecting small-scale graffiti vandalism instances, while maintaining the capability of capturing large-scale editing and deletion (blanking). Our classifiers successfully identified various graffiti instances. Examples include short unintelligible sequence of letters ("gjfjkhkfllh"), irrelevant text ("Peter likes PANCAKES!"), unproductive comment ("Microsoft are without doubt a premier company, now lets find out about them here!"), and angry expressions ("BILL GATES IS VERY RICH!!! I HATE HIM!!!").

Our classifiers also successfully identified numerous misinformation vandalism occurrences. Examples include subtle changes in numbers (change Microsoft's global annual sales from 41.36 Billion to $1), replacing existing named entities with irrelevant names (change "Mary Todd Lincoln" to "Brayson Kondracki"), removing a letter from a personal name (change "Abraham" to "Abraha"), changing date information (change Lincoln's birthday from March 4 to March 14).

Experimental results show that our approach can identify both large-scale and small-scale vandalism instances and is strong in filtering out various types of graffiti and misinformation instances. Our approach also identified some image

Table 5: Vandalism Type Distribution (M:Microsoft; L:Lincoln)

Classifier	Type	M(%)	L(%)
Logistic	Graffiti	52	52
	Large-scale Editing	22	5
	Misinformation	13	27
	Blanking	2	11
	Link Spam	1	2
	Image Attack	2	3
SVMs	Graffiti	37	55
	Large-scale Editing	38	5
	Link Spam	9	0
	Misinformation	6	22
	Blanking	9	14
	Image Attack	1	4

attack and link spam instances. This result indicates the possibility of tuning a classifier with language model statistics features to classify these types of vandalism.

4.3 Error Analysis

The two primary difficulties with our approach include separating the reverted revisions from the actual vandalized revision and identifying new legitimate Wikipedia markups and language links.

The majority of errors occurred when a new language interlink or a new Wikimarkup is introduced. For example, because an interlink such as [[be: Абрахам Лінкальн]] for the "Abraham Lincoln" article did not exist in the constructed language models, the system would identify it as an irregular editing. Also, the use of new markups and legitimate template updates have the same effect to the system. One solution to address this technical issue would be filtering out existing template tags or Wikimarkups when the language model is built.

Another common error is that the system often retrieved both the vandalized revision and the revision that reverted it. This error results from our attempt to identify the nearest legitimate edit with an automated selector described in Section 3.1. Our automated selection criteria aim to identify a previous revision that is most similar to the current revision, the revision that is least likely to be a vandalism instance. However, in the case of vandalism in very small scale (e.g. the replacement of one named entity), the automated selector selected the vandalized *n-1* revision, as it resembled the current revision n the most among the three revisions: *n-1*, *n-5*, and *n-10*. Therefore, the new revision that reverted the previous small-scale misinformation vandalism instance exhibited the same statistical feature as the vandalism revision and thus became a false positive instance. However, in the case that the nearest legitimate revision is known, that is, the use of automated selector is no longer necessary, this error can be eliminated.

5. RELATED WORK

Previous research has used Wikipedia's revision history to assess the quality and trustworthiness of Wikipedia articles [6, 18, 12, 7, 4, 8, 5, 1]. Lim et al. [8] proposed a mutual reinforcement principle to model the quality of Wikipedia articles. The authors proposed two mutual reinforcement models: the basic model and peer review model. The basic model depended on the authority of contributors and the peer review model depended on the authority of reviewers. Hu et al. [4] also modeled the dependency between Wikipedia articles and the authority of their authors to measure article quality. They assumed that if content survives through the review of high-authority reviewers it suggests an endorsement from the reviewers, thus implying the survived content has high quality. Priedhorsky et al. [12] introduced the persistent word view (PWV) – the number of times a word in an edit is viewed – to measure the impact of an edit. The PWV was based on the notion that if a contribution is viewed many times without being altered, it is likely to be a valuable edit.

Adler et al. [1] proposed a content-driven reputation system, using the knowledge of contributing authors and the trustworthiness of a word to indicate the reliability of Wikipedia articles. Zeng et al. [17] applied a Dynamic Bayesian network (DBN) to model the trustworthiness of revision history, implementing "trust view" to visualize the trustworthiness of text fragments. Javanmardi and Lopes [5] built the Wiki Trust Model (WTM) based on Hidden Markov Models. The model was to assess the reputation of Wikipedia contributors and infer reliability of article content dynamically. Their empirical study compared the evolution of the reputation of admin users and vandal users, demonstrating the capability for the WTM to identify vandal users. Vuong et al. [16] introduced three models to automatically identify controversial articles in Wikipedia. Rather than interpreting actual article contents, the authors used interaction among contributors obtained from edit history to construct these models.

However, assessments of quality and trustworthiness of articles are not direct indicators of vandalism occurrences because a poor quality edit does not necessarily imply vandalism. Contributors without adequate training or domain knowledge may produce poor quality content; however, the edits are still well-intentioned as opposed to malicious. In such cases, determining intent is likely a hard problem. Moreover, using the survival time of words as an indicator can only detect potential vandalism edits retrospectively. Pragmatically, a useful vandalism detection tool needs to identify a vandalism instance as it occurs. In addition, some vandalism edits are difficult to detect and are likely to survive through numerous reviews.[9] Therefore, the survival time and the review frequency from users may not be sufficient to identify an instance of vandalism. Methods for article quality assessment are not beneficial for detecting vandalism.

A few recent articles directly addressed the detection of vandalism on Wikipedia. Potthast et al. [11] manually crafted 16 features, using logistic regression to classify vandalism instances. The authors organized vandalism edits according to the the "Edit content" (text, structure, link, and media) and the "Editing category" (insertion, replacement, and deletion). Smets et al. [14] used the Prediction by Partial Match (PPM) compression model to classify revisions occurring in one hour from the Wikipedia main namespace.

Compared to the work of Potthast et al., we use a novel approach – based on language models – to classify poten-

[9]In our studies, we discovered an image of a tree in the article of "Abraham Lincoln" replacing the portrait of Lincoln. The tree image was named "Lincoln.jpg" and survived through 4,000 revisions for nearly two years (2004 – 2006).

tial vandalism instances. Moreover, we develop a Wikipedia editing taxonomy and provide more comprehensive categorization of vandalism instances. Compared to the work of Smets et al., we apply our method to a much larger dataset. We work on the complete revision histories of two articles which are listed among the most vandalized articles on Wikipedia. In addition, we manually inspect and label more data to provide a more complete and accurate description of vandalism instances.

6. CONCLUSIONS AND FUTURE WORK

This paper explores the use of SLM statistics as features to classify and provide an accurate ranked list of potential instances of vandalism. We categorize vandalism into seven major types that are based on a basic taxonomy of Wikipedia actions. To minimize the manual effort to annotate training set, we used a supervised active learning model, learning from the best samples in five iterations. Our work demonstrated the effectiveness and utility of our approach to detect vandalism instances. Experimental results showed that our classifiers outperformed the baseline and thrived in detecting vandalism instances of types of graffiti and misinformation, while maintaining the capability of detecting large-scale editing and deletion. Experimental results also showed that classifiers built from a small annotated data (the baseline data) can be used on data from a different domain ("Microsoft" and "Abraham Lincoln"). It implies the potential to generalize classifiers using SLM statistics as features to other Wikipedia articles.

Future empirical studies may include more data to increase the size and diversity of the pool to assess the robustness of classifiers using SLM statistics as features. Improvement may also be achieved by refining the language model by using different sets of tuning and smoothing techniques, generating trigram or n-gram models to enlarge the feature set. Additional n-gram models may capture a different set of features to improve the detection accuracy. As shown in our experiments results, the overlapping ratio between logistic regression and SVMs is low. Future studies may use ensemble methods, comprising multiple classifiers to detect specific type of vandalism. In addition, building customized classifiers to detect targeted types of vandalism may further improve accuracy , providing the capability to prioritize the anti-vandal task based on the type of vandalism.

7. ACKNOWLEDGMENTS

This publication was made possible by Grant Number UL1RR024979 from the National Center for Research Resources (NCRR), a part of the National Institutes of Health (NIH). Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the CTSA or NIH. Our special thanks go to Information Retrieval Research Group and Data Mining Iowa Group (DMIG) for their valuable comments.

8. REFERENCES

[1] B. T. Adler, J. Benterou, K. Chatterjee, L. de Alfaro, I. Pye, and V. Raman. Assigning trust to Wikipedia content. Technical report, School of Engineering, University of California, Santa Cruz, 2007.

[2] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, page 430. ACM, 2007.

[3] P. Clarkson and R. Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. In *Fifth European Conference on Speech Communication and Technology*, pages 2707—2710, September 1997.

[4] M. Hu, E. Lim, A. Sun, H. W. Lauw, and B. Vuong. Measuring article quality in Wikipedia: Models and evaluation. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pages 243–252, Lisbon, Portugal, 2007. ACM.

[5] S. Javanmardi and C. Lopes. Modeling trust in collaborative information systems. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 299–302, 2007.

[6] A. Kittur, B. Suh, and E. H. Chi. Can you ever trust a wiki?: Impacting perceived trustworthiness in Wikipedia. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 477–480, San Diego, CA, USA, 2008. ACM.

[7] A. Kittur, B. Suh, B. A. Pendleton, and E. H. Chi. He says, she says: Conflict and coordination in Wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 453–462, San Jose, California, USA, 2007. ACM.

[8] E. Lim, B. Vuong, H. W. Lauw, and A. Sun. Measuring qualities of articles contributed by online communities. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 81–87. IEEE Computer Society, 2006.

[9] A. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of ICML-98, 15th International Conference on Machine Learning*, pages 350–358, 1998.

[10] G. Mishne, D. Carmel, and R. Lempel. Blocking blog spam with language model disagreement. In *1st International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[11] M. Potthast, B. Stein, and R. Gerling. Automatic vandalism detection in Wikipedia. In *Advances in Information Retrieval*, pages 663–668. Springer Berlin / Heidelberg, 2008.

[12] R. Priedhorsky, J. Chen, S. T. K. Lam, K. Panciera, L. Terveen, and J. Riedl. Creating, destroying, and restoring value in Wikipedia. In *Proceedings of the International ACM Conference on Supporting Group Work*, pages 259–268, Sanibel Island, Florida, USA, 2007. ACM.

[13] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000.

[14] K. Smets, B. Goethals, and B. Verdonk. Automatic vandalism detection in Wikipedia: Towards a machine learning approach. In *AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy*, pages 43–48, 2008.

[15] F. B. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 575–582, Vienna, Austria, 2004. ACM.

[16] B. Vuong, E. Lim, A. Sun, M. Le, and H. W. Lauw. On ranking controversies in Wikipedia: Models and evaluation. In *Proceedings of the International Conference on Web Search and Web Data Mining*, pages 171–182, Palo Alto, California, USA, 2008. ACM.

[17] H. Zeng, M. Alhossaini, R. Fikes, and D. L. McGuinness. Mining revision history to assess trustworthiness of article fragments. In *Proc. of the 2nd Intl. Conf. on Collaborative Computing: Networking, Applications, and Worksharing*, 2006.

[18] H. Zeng, M. A. Alhossaini, L. Ding, R. Fikes, D. L. McGuinness, and Stanford Univ. CA Knowledge Systems Lab. *Computing Trust from Revision History*. Defense Technical Information Center, 2006.