Using Language Models to Detect Wikipedia Vandalism

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

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

W. Nick Street
Management Sciences
Department & IGPI
The University of Iowa
Iowa City, IA 52242, USA
nick-street@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 explores a statistical language modeling approach for detecting 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, 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 essential.

This paper offers first a categorization of Wikipedia vandalism types and identifies technical challenges associated with detecting each category. In addition, 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. 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.5.3 [**Information Interfaces**]: Group and Organization Interfaces—*Collaborative computing, Computer-supported cooperative work, Web-based interaction*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; K.4.3 [**Computers and Society**]: Organizational Impacts—*Computer-supported collaborative work*

Keywords

Wikipedia, vandalism, statistical language model

1. INTRODUCTION

1.1 Motivation

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[2] [3]. Wikipedia has taken many measures to address the challenges of vandalism, such as restricting the privileges of anonymous users, adopting "article validation"[4] and using an "abuse filter"[5] to control user activities by reacting automatically to suspicious user behaviors.

Currently active tools to fight vandalism include ClueBot[6] and VoABot II[7]. The two anti-vandal bots provided an automatic solution to detect and revert vandalism edits. However, research [16, 12] 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.

[1] http://en.wikipedia.org/wiki/Wikipedia:Vandalism
[2] http://meta.wikimedia.org/wiki/Anti-vandalism_ideas
[3] http://en.wikipedia.org/wiki/Wikipedia:Cleaning_up_vandalism
[4] http://meta.wikimedia.org/wiki/Article_validation
[5] http://en.wikipedia.org/wiki/Wikipedia:AF
[6] http://en.wikipedia.org/wiki/User:ClueBot
[7] http://en.wikipedia.org/wiki/User:VoABot_II

- 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 malicious editing – thus protecting the integrity of Wikipedia articles.

1.2 Paper Organization

The paper is structured as follows. In Section 2, we review previous academic research on Wikipedia vandalism and revision history. In Section 3, we describe the taxonomy of Wikipedia actions and the categorization of vandalism. In Section 4, we describe the implementation of the system, including the system framework and the data set used for the experiment. In addition, we present the statistical language model in the same section. In Section 5, we present experiment results. In Section 6, we conclude the paper and discuss the opportunity for future work.

2. RELATED WORK

Previous research has used Wikipedia's revision history to assess the quality and trustworthiness of Wikipedia articles [9, 20, 13, 10, 7, 11, 8, 1]. Lim et al. [11] 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. [7] 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. [13] 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. [19] 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 [8] 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. [18] 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 is 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. Methods for article quality assessment are not beneficial for detecting vandalism. In addition, some vandalism edits are difficult to detect and are likely to survive through numerous reviews.[8] Therefore, the survival time and the review frequency from users may not be sufficient to identify an instance of vandalism.

A few recent articles directly addressed the detection of vandalism on Wikipedia. Potthast et al. [12] 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. [16] used the Prediction by Partial Match (PPM) compression model to classify revisions occurring in one hour from the Wikipedia main namespace[9]. Compared to the work of Potthast et al., we use a novel method –based on language models – to identify potential 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 three 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. Contributions of this paper are:

- We provide a taxonomy structure for editing actions on Wikipedia, categorizing also types of vandalism.
- We inspect the distribution of vandalism instances in each vandalism type manually, demonstrating that articles in different categories are attacked by vandals differently.
- We analyze the technical difficulties of identifying vandalism instances in each category.
- We study three articles that are in the list of most vandalized pages on Wikipedia[10]. We choose two articles from the category of "Computing and Internet" – Apple Inc. and Microsoft – and one article from the category of "History" – Abraham Lincoln. Our experiments use full revision histories of the three articles.
- We explore a novel application of language models, i.e., to indicate vandalism instances. We build statistical language models using the CMU-toolkit [4] and test the models with the *diff* (unix command) results between consecutive revisions. Our indicator functions built upon perplexity values, word numbers, number of words that are out of vocabulary, and percentage of

[8] 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).

[9] http://en.wikipedia.org/wiki/Main_namespace

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

words that are out of vocabulary; they are shown to be effective indicators for identifying vandalism.

- We develop a system that strives to provide an accurate ranked list of potential vandalism instances for Wikipedia administrators and stewards.

3. TYPES OF VANDALISM

In this section, we present a 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, links, references, and sections; the formatting includes HTML tags or CSS, and Wikipedia templates. Figure 1 illustrates the taxonomy of actions on Wikipedia.

Previous research has identified many common types of vandalism. Viégas et al. [17] visualized revision histories in their "history flow" and demonstrated that mass deletions are easy to spot on the history flow visualization. The authors identified five common types of vandalism: mass deletion, offensive copy, phony copy, phony redirection, and idiosyncratic copy. Priedhorsky et al. [13] 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 authors also categorized Wikipedia damaged edits to seven types: misinformation, mass delete, partial delete, offensive, spam, nonsense, and other. Although damage edits were not referred as vandalism in their work, they were in fact in line with the definition of Wikipedia vandalism, which is described as "any addition, removal, or change of content made in a deliberate attempt to compromise the integrity of Wikipedia". Potthast et al. [12] organized vandalism edits according to the the "Edit content" (text, structure, link, and media) and the "Editing category" (insertion, replacement, and deletion).

Compared to existing works on vandalism categorization, we use a systematic taxonomy to categorize vandalism instances. We elaborate on commonly known types of vandalism and analyze the technical challenges of each category. A summary of vandalism categorization is shown in Table 1. The technical challenges of each category are described as follows.

Blanking is a deletion of the entire article or massive amount of existing content. Detecting instances of blanking is straightforward since it is self-evident that if a new revision contains zero or very few words, it is an instance of blanking. In our experiments, we catogorized a revision as an instance of blanking if the new revision was at least 90% smaller than the average length of the page.

Large-scale Editing is an insertion or a change of massive extent. Detecting instances of large-scale editing is effortless since they are manifest on the diff result of two consecutive revisions. We defined a revision as an instance of large-scale editing if the size of new edits (insertion and change) was twice larger than the median value[11] of the length of all edits in the previous diff history.

Graffiti is an insertion of completely irrelevant, random, or unintelligible text. Some instances of graffiti can be identified by examining the ratio of upper-case letters or the maximum length of a word in the new edit. These rules may filter out vandalism edits that contain a large portion of upper case letters (e.g. "I LOVE MAC! APPLE COMPUTER RULZ!!") and a long sequence of meaningless letters(e.g. "daewiatlgkjdflkgsyhgfawerekfhgdslkgdajaeef"). However, graffiti such as "I like eggs." or "John loves Jane" would not be discovered by these rules. One can generate more rules to catch more graffiti instances but such a rule-based filtering system is neither extensible nor easy to maintain.

Angry Expression is an insertion of profanity or other vocabulary or phrases to express strong anger. Although it is possible to filter out edits that contain words from a list of profanity vocabulary, it is difficult to maintain the list as the profanity vocabulary 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 system would only be able to achieve limited effect.

Misinformation is an insertion of false information or change of named entities such as personal names, locations, and product names. It usually occurs when vandals attack the information box (brief summary box on the left of the page). 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 as list is a non-trivial task and its effect would not be evident if the NER has limited performance.

Image Attack is a replacement of existing image with a irrelevant one, or an insertion of one or more images to damage the page. Although a scan with regular expression searching for image file extension (e.g. ".jpg", ".tiff" etc.) would note an insertion or change of image, it is challenging to decide the validity of image from the name of the image file. Resizing images is a very common and valid edit on Wikipedia. Labeling any revision of images as vandalism would only generate large number of false positives. Examining the name of image may discover some instances of Image Attach, such as noting a "batman.jpg" on the article of Abraham Lincoln. However, one can easily change the title of an image to make it appear like a valid image from the text, for example, naming an image of trees as "Lincoln.jpg".

Link Spam in an insertion of external or internal links that are irrelevant to the article. Identifying an insertion of links can be done by using a regular expression to match patterns of http links. Although one may use some heuristics to differentiate the quality of links by analyzing the domain of the link, a quality link can come from any domain such as personal blogs. It requires further analysis on the content of the link to determine its relevance to the article.

Unproductive Comments are insertions of contents only remotely related to the subject or fruitless comments that undermine the quality of the article. As vandalism in this category varies from article to article, it is difficult to deduce general rules to identify unproductive comments.

[11] 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.

Table 1: Types of Vandalism

Type	Action Taxonomy	Definition	Example
Blanking	Delete(massive)	Delete the entire article or massive amount of existing content.	

Large-scale Editing	Insert(massive)
	Change(massive)
		Add massive amount of malicious text to lengthen the article to slow the loading speed or change massive portion of the existing content.	Replace all the occurrences of "Apple" to "Orange" in the article of "Apple Inc."

Graffiti	Insert–Text	Insert completely irrelevant, random, or unintelligible text.	• *I like eggs!* • *dfdfefefd jaaaei #$%&@@#* • *John Smith loves Jane Doe.*
Angry Expression	Insert–Text	Usage of profanity or other vocabulary or phrases to express strong anger.	• *Microsoft Suxx!!* • *I HATE MAC!!!!* • *This ***king program is EVIL!!!*
Misinformation	Insert–Text Change–Text	Insert false information or change named entities such as 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).	• *Key Person: John Lennon* (on Microsoft page) • *CEO: Bruce Wayne* (on Apple Inc. page)
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	Insert external or internal links which are irrelevant to the article	• *http://www.wierdspot.com Abe's Personal Diary*
Unproductive Comments	Insert–Text	Insert contents only remotely related to the subject or fruitless comments that undermine the quality of the article.	• *To what end is only apparent to themselves* • *Buying their computers is totally a waste of your money.* • *Mac is for people who are too slow to use PC or Unix.*
Subtle Revision	Change–Text	Change the existing content in a nearly indiscernible manner. It usually involves changing the spelling of words, deleting one or more digits for numbers, or inverting a positive statement to negative.	• *propaganda of the deed* → *propaganda of the dead* • *4,600 million* → *4,000 million* • *This is true* → *This is not true*
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.	• *<font size=6>Bill Gates</font>*

SUBTLE REVISION is a change of the existing content in a nearly indiscernible manner. It usually involves changing 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. Although the automated diff processing would identify any subtle revision, it is challenging to create rules to differentiate a valid correction of typos or grammar from a malicious subtle revision.

IRREGULAR FORMATTING is a change or an insertion of HTML or CSS format tags that are not standard to the editing guideline; changing the format of existing texts or images to damage the appearance of the article. Vandalism using irregular formatting is usually detectable by both human inspection and an automated diff process. However, it requires numerous rules to distill out invalid formatting without depriving users' flexibility in editing.

To address the technical challenges of identifying instances of vandalism, we built statistical language models and used the result of model evaluation (e.g., perplexity, number of words that are out of vocabulary, etc.) as an indicator for vandalism. In Section 4, we describe the model in detail and demonstrate the effectiveness of the approach.

4. EXPERIMENTS

4.1 Dataset Description

We worked with the Wikipedia page history archive from February 24th, 2009[12]. Our corpus includes complete revision histories (note this aspect is unique to our research) for three Wikipedia articles: Abraham Lincoln, Apple Inc., and Microsoft. These articles are acknowledged to be among the most vandalized pages[13]. The reason for choosing the most vandalized pages is to acquire an extensive amount of vandalism instances for the analysis. We intentionally chose two articles from the "Computing and Internet" and one article from the "History" to demonstrate the similarity and differences of the vandalism pattern across categories. Table 2 shows the summary of the dataset.

Since vandalism instances are not systematically archived by Wikipedia, previous research[10, 13] typically use regular expressions matched against revision comments to label vandalism, matching any form of the word "vandal" and "rvv" ("revert due to vandalism"). The regular expressions we used were "∼ .*vandal.*" and "∼ .*rvv.*" The revisions being reverted are labeled as vandalism instances. 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.

Smets et al. used the reference of any revert actions in revision comments to automatically label the data. The downside of this approach is that it produced false positive instances because a revert may result from causes such as the lack of proper reference or the edit wars between users. The authors reported that the false positive rate was 8% in the worst case. However, the rate would be underestimated for many controversial articles, where large-scale edit wars occurred. 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, this labeling approach still cannot exclude the possibility of false negatives in the dataset.

This paper aims to provide a ranked list having a high positive rate to alert stewards of Wikipedia articles of vandalism instances. We therefore labeled the data using the regular expression to match explicit notions of vandalism in revision comments to eliminate false positives. This labeling method identified 330 instances of vandalism from 8,816 revisions of the article "Abraham Lincoln;" 229 instances of vandalism from 6,715 revisions of the article "Apple Inc.;" and 335 instances of vandalism from 8,220 revisions of the article "Microsoft." This rate of vandalism instance is higher than that reported in previous studies since we specifically chose highly vandalized articles. Key to note that we also identify additional instances by manually annotating top ranked revisions (using our methods) if they are instances of vandalism.

4.2 Experiment Setup

Figure 2 illustrates the system structure and preprocessing of the revision history. We extracted the three 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 Java BreakIterator class to preprocess the revision history. Each revision was processed into one sentence per line to enable diff process at the sentence level.

We used CMU-toolkit [4] to build statistical language models for each revision of a page. Moving through the sequence of reversions we adopt the following process. Assuming we are at revision (n+1) we compute the unix diff between it and the previous version (n). This diff is directional in that we record only the new data that is in the n+1 version as compared to version n. Meanwhile we also build a set of language models for the n page version, another set for the n-4 version, and yet another for the n-9 version. Each set consists of a bigram model a trigram model, each with or without Wikimarkup. The diff data for the n+1 revision is then tested using each of these prior models. Each test yields a set of values pertaining to: perplexity, number of words, number of words that are out of vocabulary, and percentage of words that are out of vocabulary. Table 3 demonstrates an example of diff and highlights its representation within each language model. In the example, the newer revision replaced "two-room" with "one-room," deleted the internal link to "log cabin"[14], and inserted "348 acre". Thus for example, if the newer version contains no words that exists in the previous revision, the number of bigram or trigram hits would be zero. In addition, it is noted that models with or without Wikimarkup rendered different results. Bigram model with Wikimarkup discovered more new bigrams from the newer revision, detecting the deletion of link markups. However, the bigram model without Wikimarkup was less sensitive to changes of templates and markups but more robust in detecting changes of actual content.

As vandalism often involves the use of unexpected vocabulary to draw attention, an instance of vandalism would produce high surprise factor when compared with the previous

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

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

[14] The symbol "[[]]" is a Wikimarkup for links.

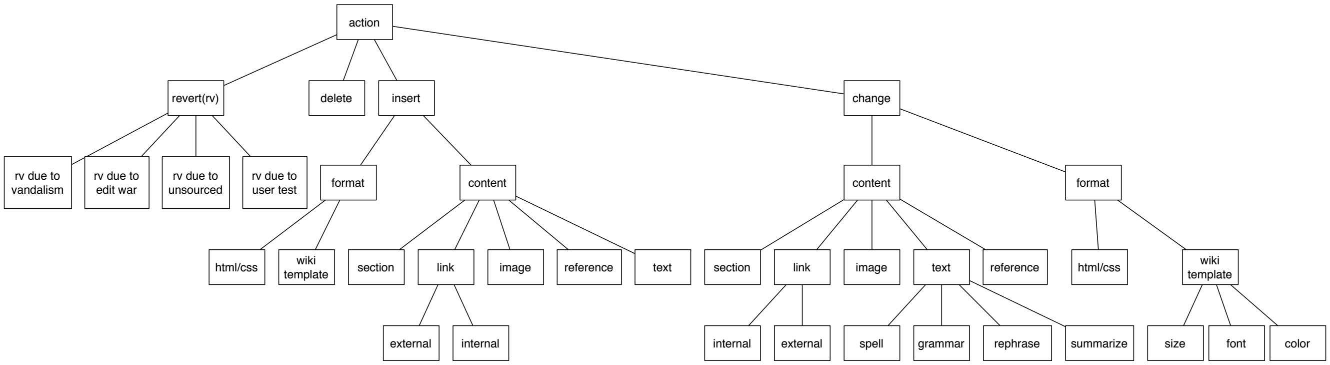

Figure 1: Wikipedia Action Taxonomy

Table 2: Dataset Overview

Article	# of Revisions	# of Vandalisms	# of Users	Avg. # of Sentences	Time Span
Abraham Lincoln	8,816	330 (3.7%)	1,668	565	2002/02/24 - 2008/03/13
Apple Inc.	6,715	229 (3.4%)	1,475	546	2002/02/09 - 2008/03/10
Microsoft	8,220	335 (4%)	1,720	870	2002/02/14 - 2008/03/14

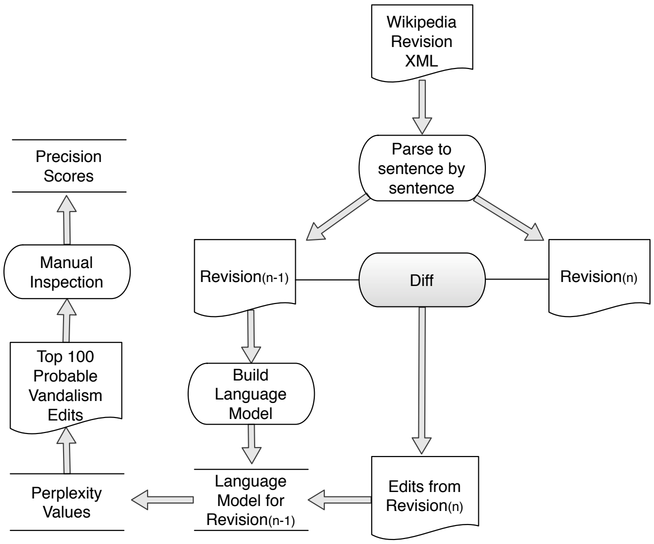

Figure 2: Flowchart of experiments.

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 of distinguishing between vandalism instance and its follow up revert, we evaluate each diff result against three language models: the model built from the previous revision, the previous 5th revision, and the previous 10th revision. Consequently, an instance of vandalism would need to have three evaluation results of high perplexity; and a revert would only have one high perplexity result which is using the model of the previous revision. The terminology we used to refer each result from the evaluation is organized in Table 4.

4.3 Statistical Language Models

Statistical language modeling(SLM) [15] computes the distribution of natural language and assigns a probability to the occurrence of a string S or a sequence of m words. SLM is commonly applied to many natural language processing tasks [14] such as speech recognition [6], machine translation [3], text summarization [2] and information retrieval [5]. CMU SLM toolkit [4] allows construction and testing of bigram and trigram language models. The *evallm* tool evaluates the language model dynamically, providing data of 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 three variations of language models: trigram modeling preserving Wikimarkups, bigram modeling preserving Wikimarkups, and bigram modeling eliminating Wikimarkups from the revision text. Thus each revision had a trigram model with Wikimarkups, a bigram model with Wikimarkups, and a bigram model without Wikimarkups. For each diff result, we used *evallm* tool to test it against the previous language model, the previous 5th language model, and the previous 10th language model. Therefore, each diff result has 9 results from *evallm*.

As vandalism usually involves unforeseen vocabulary in the edit to attract attention, our experiments built indicators to select instances with high perplexity value, small number of n-gram hits, large number of OOV, and high OOV percentage value. Table 4 summarize the terminology we used in the indicators and Table 5 describes each indicator function. Indicators I1 to I6 used various combinations of perplexity results in descending order to rank revisions. I7 and I8 identified revisions containing zero n-grams from previous language models and ranked revisions by the sum of three perplexity results (perplex_1, perplex_5, and perplex_10) in descending order. I9 and I10 ranked revisions by the sum of the percentage of out-of-vocabulary words in descending order, identifying revisions with high percentage of words that were not previously seen in the revision history.

Table 3: Diff Process

Revision	Text	Trigram with markup	Bigram with markup	Bigram without markup
n	He was born in a two-room [[log cabin]] on the Sinking Spring Farm.	He was born was born in born in a in a two-room two-room [[log [[log cabin]] on cabin]] on the on the Sinking Sinking Spring Farm.	He was was born born in in a a two-room two-room [[log [[log cabin]] cabin]] on on the the Sinking Sinking Spring Spring Farm.	He was was born born in in a a two-room two-room log log cabin cabin on on the the Sinking Sinking Spring Spring Farm.
n+1	He was born in a one-room log cabin on the 348 acre Sinking Spring Farm.	He was born was born in born in a in a one-room one-room log log cabin on cabin on the on the 348 the 348 acre 348 acre Sinking acre Sinking Spring Sinking Spring Farm.	He was was born born in in a a one-room one-room log log cabin cabin on on the the 384 384 acre acre Sinking Sinking Spring Spring Farm.	He was was born born in in a a one-room one-room log log cabin cabin on on the the 384 384 acre acre Sinking Sinking Spring Spring Farm.
Diff *(n+1) - n*		**in a one-room** **one-room log** **log cabin on** **cabin on the** **on the 348** **the 348 acre** **348 acre Sinking** **acre Sinking Spring.**	**a one-room** **one-room log** **log cabin** **cabin on** **the 384** **384 acre** **acre Sinking**	**a one-room** **one-room log** **the 384** **384 acre** **acre Sinking**

Table 4: Definition of Terms

Term	Definition
perplex_1	Perplexity computed from language model built from previous revision.
perplex_5	Perplexity computed from language model built from previous *5th* revision.
perplex_10	Perplexity computed from language model built from previous *10th* revision.
oov_num	Number of words that don't exist on the vocabulary list
oov_per_1	Percentage of words that are out of vocabulary based on language model built from previous revision.
oov_per_5	Percentage of words out of vocabulary based on language model built from previous *5th* revision.
oov_per_10	Percentage of words out of vocabulary based on language model built from previous *10th* revision.
word_num	Number of words used to compute the perplexity.

Table 5: Model Description

Model	Description
Indicator 1 (I1)	Perlex_1 from the previous bigram language model with Wikimarkup.
Indicator 2 (I2)	Perlex_1 from the previous bigram language model *without* Wikimarkup.
Indicator 3 (I3)	Perlex_1 from the previous trigram language model with Wikimarkup.
Indicator 4 (I4)	Sum of perplex_1, perplex_5, and perplex_10 from bigram model with Wikimarkup
Indicator 5 (I5)	Sum of perplex_1, perplex_5, and perplex_10 from bigram model *without* Wikimarkup
Indicator 6 (I6)	Sum of perplex_1 from bigram with markup and perplex_1 from bigram *without* Wikimarkup
Indicator 7 (I7)	Sum of perplex_1, perplex_5, and perplex_10 from bigram model with Wikimarkup, condition on zero word_num and the oov_num > 0
Indicator 8 (I8)	Sum of perplex_1, perplex_5, and perplex_10 from bigram model *without* Wikimarkup conditioned on zero word_num and the oov_num > 0
Indicator 9 (I9)	Sum of oov_per_1, oov_per_5, and oov_per_10 from bigram with Wikimarkup conditioned on the sum of word_num_1, word_num_5, and word_num_10 > 0
Indicator 10 (I10)	Sum of oov_per_1, oov_per_5, and oov_per_10 from bigram *without* Wikimarkup conditioned on the sum of word_num_1, word_num_5, and word_num_10 > 0

4.4 Results and Analysis

Our overall strategy is to test each revision in a page's history, computing the various language model-based indicator functions as shown in Table 5. For each indicator function we then rank the revisions by score returned and then analyze the top 100 revisions. We judge each revision as to whether it is an instance of vandalism; this is done if we do not already know the status of the revision (from the comments section). In this way we systematically add to the pool of judgments. After this process of judging the top 100 ranked revisions we compute precision at top 100.

Tables 6 to 8 show the precision of each indicator at the top 100 identified revisions. We manually inspected the top 100 revisions from all 10 indicators to annotate instances of vandalism. Column Prec@100 shows the precision score at the top 100 results and columns I1-I10 show the overlap between vandalism instances found by the different models. The entries are the ratio of the size of the intersection of the two results and the size of their union. For example, in Table 6, models I1 and I2 found seven common instances, so the reported number is 7 / 39 = 0.18. It is noted that trigram modeling had very limited performance. Generally, using a single perplexity value or the combination of several achieved around 30% precision. The best performance occurred in indicator I9, which achieved 74% precision for the article "Abraham Lincoln". Indicators I7, I8, I9, and I10 have minimal overlap with other indicators. It implies that these indicators have discovered different types of vandalism.

We examined the distribution of instances of vandalism for each type. We manually inspected all known instances of vandalism and categorized them based on the types outlined in Section 3. Table 9 summarizes the results for the three articles: Apple Inc., Microsoft, and Abraham Lincoln. In Table 9, italic percentage data under the SLM column refers to the percentage of instances discovered by the SLM approach using all ten models. For example, the SLM approach discovered 95 instances of Graffiti vandalism, which composed 56% (95/169) of discovered instances by the approach and 67% (95/142) among the discovered instances of Graffiti. Some instances of vandalism are associated with more than one type of vandalism. Suppose an instance of vandalism involves a blanking of the original article and then adding graffiti text, it would have annotations of both "blanking" and "graffiti." This explains why the sum of percentages is larger than one hundred.

Generally, "Graffiti" is the prevailing type of vandalism across the three articles. Instances of the types of "Graffiti," "Angry Expression," and "Misinformation" together made up at least 70% of vandalism. It is noted that, as an article in the "History" category, "Abraham Lincoln" had an exceptional number of instances in the type of "Misinformation." It implies that vandals are inclined to insert false information in historical articles. The SLM approach is effective in detecting instances from "Graffiti" and "Angry Expression." It also had reasonable performance in detecting "Irregular Formatting" and "Link spam." However, the approach is less effective in detecting instances from "Subtle Revision."

5. CONCLUSIONS

This paper explores the use of SLM to detect instances of vandalism. We categorize vandalism into ten major types, using a basic taxonomy of Wikipedia actions. Ten indica-

Table 6: Results – Apple

	Prec@100	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
I1	19	1.0									
I2	27	0.18	1.0								
I3	3	0.16	0.07	1.0							
I4	21	0.67	0.14	0.14	1.0						
I5	29	0.2	0.56	0.07	0.19	1.0					
I6	31	0.52	0.35	0.1	0.44	0.4	1.0				
I7	50	0.0	0.07	0.0	0.01	0.07	0.05	1.0			
I8	62	0.0	0.0	0.0	0.01	0.0	0.0	0.47	1.0		
I9	49	0.15	0.07	0.04	0.13	0.05	0.16	0.03	0.03	1.0	
I10	58	0.13	0.15	0.03	0.13	0.16	0.19	0.09	0.03	0.6	1.0

Table 7: Results – Microsoft

	Prec@100	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
I1	43	1.0									
I2	33	0.27	1.0								
I3	30	0.66	0.34	1.0							
I4	32	0.67	0.27	0.72	1.0						
I5	27	0.23	0.67	0.3	0.28	1.0					
I6	47	0.64	0.51	0.6	0.58	0.42	1.0				
I7	54	0.0	0.01	0.0	0.0	0.01	0.0	1.0			
I8	60	0.02	0.0	0.0	0.0	0.0	0.0	0.75	1.0		
I9	62	0.28	0.13	0.26	0.24	0.11	0.24	0.01	0.03	1.0	
I10	40	0.11	0.07	0.13	0.09	0.06	0.12	0.04	0.01	0.38	1.0

Table 8: Results – Abraham Lincoln

	Prec@100	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
I1	27	1.0									
I2	24	0.09	1.0								
I3	16	0.48	0.08	1.0							
I4	19	0.53	0.05	0.52	1.0						
I5	28	0.06	0.58	0.07	0.07	1.0					
I6	34	0.24	0.35	0.22	0.2	0.41	1.0				
I8	60	0.0	0.06	0.0	0.01	0.06	0.02	1.0			
I8	49	0.03	0.0	0.03	0.05	0.0	0.0	0.47	1.0		
I9	74	0.1	0.04	0.07	0.11	0.04	0.06	0.06	0.09	1.0	
I10	42	0.01	0.16	0.04	0.03	0.09	0.12	0.04	0.01	0.2	1.0

Table 9: Distributions of Vandalism Types

	Apple Inc		**Microsoft**		**Abraham Lincoln**	
	Total	SLM	Total	SLM	Total	SLM
Blanking	27 (8%)	2 (1%)*(7%)*	41 (10%)	2 (1%)*(5%)*	20 (4%)	5 (3%)*(25%)*
Image Attack	12 (4%)	4 (2%)*(33%)*	23 (5%)	9 (5%)*(39%)*	10 (2%)	5 (3%)*(50%)*
Link Spam	11 (3%)	5 (3%)*(45%)*	21 (5%)	7 (4%)*(33%)*	11 (2%)	8 (4%)*(73%)*
Graffiti	142 (42%)	95 (56%)*(67%)*	147 (34%)	88 (50%)*(60%)*	199 (42%)	109 (56%)*(55%)*
Angry Expression	52 (15%)	36 (21%)*(69%)*	69 (16%)	52 (30%)*(75%)*	34 (7%)	24 (12%)*(71%)*
Misinformation	50 (15%)	20 (12%)*(40%)*	87 (20%)	12 (7%)*(14%)*	192 (40%)	41 (21%)*(21%)*
Subtle Revision	38 (11%)	1 (1%)*(3%)*	51 (12%)	2 (1%)*(4%)*	5 (1%)	0 (0%)*(0%)*
Unproductive Cmts	35 (10%)	13 (8%)*(37%)*	38 (9%)	7 (4%)*(18%)*	12 (3%)	3 (2%)*(25%)*
Large-scale Editing	8 (2%)	1 (1%)*(13%)*	44 (10%)	19 (11%)*(43%)*	6 (1%)	2 (1%)*(33%)*
Irregular Formatting	8 (2%)	5 (3%)*(63%)*	18 (4%)	11 (6%)*(61%)*	17 (4%)	7 (4%)*(41%)*
TOTAL	**340**	**169**	**429**	**175**	**478**	**195**

tors were generated from the SLM *evallm* process to identify probable instances of vandalism. We manually inspected and annotated results from SLM indicators to present the distribution of instances among vandalism types.

The SLM approach is shown to be effective in detecting a large portion of vandalism, addressing the technical challenges of detecting vandalism described in Section 3. Our best results are generally obtained using indicators I7 through I10, which use conditioned combinations of perplexity values. The best precision scores for each of the 3 pages vary between 62% and 74%. It is also shown that articles from different categories ("History" as apposed to "Computing and Internet" in our experiments) appear to have different distributions of types of vandalism.

We have demonstrated the potential of using SLM to detect instances of vandalism. In future research we will apply multiple evidence combination functions such as CombSum or CombMNZ to combine the indicator functions to improve retrieval effectiveness. We will also design more indicators and apply them to a larger dataset of articles from different categories. We will also refine the language model by using different sets of tuning and smoothing techniques. Finally, will use SLM indicators as a new set of features for vandalism classification tasks.

6. 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 for their valuable comments.

7. 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] A. L. Berger and V. O. Mittal. OCELOT: A system for summarizing web pages. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 144–151, Athens, Greece, 2000. ACM.

[3] F. Casacuberta and E. Vidal. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(2):205–225, June 2004.

[4] 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.

[5] W. Croft. Language models for information retrieval. In *19th International Conference on Data Engineering*, pages 3–7, 2003.

[6] A. Ganapathiraju. *Support vector machines for speech recognition.* PhD thesis, Mississippi State University, 2002.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] R. Rosenfeld. The CMU statistical language modeling toolkit and its use in the 1994 ARPA CSR evaluation. In *Proceedings of the Spoken Language Systems Technology Workshop*, pages 47–50, 1995.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.