"Got You!": Automatic Vandalism Detection in Wikipedia with Web-based Shallow Syntactic-Semantic Modeling

William Yang Wang and Kathleen R. McKeown
Department of Computer Science
Columbia University
yw2347@columbia.edu kathy@cs.columbia.edu

Abstract

Discriminating vandalism edits from non-vandalism edits in Wikipedia is a challenging task, as ill-intentioned edits can include a variety of content and be expressed in many different forms and styles. Previous studies are limited to rule-based methods and learning based on lexical features, lacking in linguistic analysis. In this paper, we propose a novel Web-based shallow syntactic-semantic modeling method, which utilizes Web search results as resource and trains topic-specific n-tag and syntactic n-gram language models to detect vandalism. By combining basic task-specific and lexical features, we have achieved high F-measures using logistic boosting and logistic model trees classifiers, surpassing the results reported by major Wikipedia vandalism detection systems.

1 Introduction

Online open collaboration systems are becoming a major means of information sharing on the Web. With millions of articles from millions of resources edited by millions of people, Wikipedia is a pioneer in the fast growing, online knowledge collaboration era. Anyone who has Internet access can visit, edit and delete Wikipedia articles without authentication.

A primary threat to this convenience, however, is vandalism, which has become one of Wikipedia's biggest concerns (Geiger, 2010). To date, automatic countermeasures mainly involve rule-based approaches and these are not very effective. Therefore, Wikipedia volunteers have to spend a large amount of time identifying vandalized articles manually, rather than spending time contributing content to the articles. Hence, there is a need for more effective approaches to automatic vandalism detection.

In contrast to spam detection tasks, where a full spam message, which is typically 4K Bytes (Rigoutsos and Huynh, 2004), can be sampled and analyzed (Itakura and Clarke, 2009), Wikipedia vandals typically change only a small number of words or sentences in the targeted article. In our preliminary corpus (Potthast et al., 2007), we find the average size of 201 vandalized texts to be only 1K Byte. This leaves very few clues for vandalism modeling. The question we address in this paper is: given such limited information, how can we better understand and model Wikipedia vandalism?

Our proposed approach establishes a novel classification framework, aiming at capturing vandalism through an emphasis on shallow syntactic and semantic modeling. In contrast to previous work, we recognize the significance of natural language modeling techniques for Wikipedia vandalism detection and utilize Web search results to construct our shallow syntactic and semantic models. We first construct a baseline model that captures task-specific clues and lexical features that have been used in earlier work (Potthast et al., 2008; Smets et al., 2008) augmenting these with shallow syntactic and semantic features. Our main contributions are:

- Improvement over previous modeling methods with three novel lexical features
- Using Web search results as training data for syntactic and semantic modeling
- Building topic-specific n-tag syntax models and syntactic n-gram models for shallow syntactic and semantic modeling

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 1146–1154,
Beijing, August 2010

2 Related Work

So far, the primary method for automatic vandalism detection in Wikipedia relies on rule-based bots. In recent years, however, with the rise of statistical machine learning, researchers have begun to treat Wikipedia vandalism detection task as a classification task. To the best of our knowledge, we are among the first to consider the shallow syntactic and semantic modeling using Natural Language Processing (NLP) techniques, utilizing the Web as corpus to detect vandalism.

ClueBot (Carter, 2007) is one of the most active bots fighting vandalism in Wikipedia. It keeps track of the IP of blocked users and uses simple regular expressions to keep Wikipedia vandalism free. A distinct advantage of rule-based bots is that they have very high precision. However they suffer from fixed-size knowledge bases and use only rigid rules. Therefore, their average recall is not very high and they can be easily fooled by unseen vandalism patterns. According to Smets et al., (2008) and Potthast et al., (2008), rule-based bots have a perfect precision of 1 and a recall of around 0.3.

The Wikipedia vandalism detection research community began to concentrate on the machine learning approaches in the past two years. Smets et al. (2008) wrapped all the content in *diff* text into a bag of words, disregarding grammar and word order. They used Naïve Bayes as the classification algorithm. Compared to rule-based methods, they show an average precision of 0.59 but are able to reach a recall of 0.37. Though they are among the first to try machine learning approaches, the features in their study are the most straightforward set of features. Clearly, there is still room for improvement.

More recently, Itakura and Clarke (2009) have proposed a novel method using Dynamic Markov Compression (DMC). They model their approach after the successful use of DMC in Web and Mail Spam detection (Bratko et al., 2006). The reported average precision is 0.75 and average recall is 0.73.

To the best of our knowledge, Potthast et al., (2008) report the best result so far for Wikipedia vandalism detection. They craft a feature set that consists of interesting task-specific features. For example, they monitor the number of previously submitted edits from the same author or IP, which is a good feature to model author contribution. Their other contributions are the use of a logistic regression classifier, as well as the use of lexical features. They successfully demonstrate the use of lexical features like vulgarism frequency. Using all features, they reach an average precision of 0.83 and recall of 0.77.

In addition to previous work on vandalism detection, there is also earlier work using the web for modeling. Biadsy et al. (2008) extract patterns in Wikipedia to generate biographies automatically. In their experiment, they show that when using Wikipedia as the only resource for extracting named entities and corresponding collocational patterns, although the precision is typically high, recall can be very low. For that reason, they choose to use Google to retrieve training data from the Web. In our approach, instead of using Wikipedia edits and historical revisions, we also select the Web as a resource to train our shallow syntactic and semantic models.

3 Analysis of Types of Vandalism

In order to better understand the characteristics of vandalism cases in Wikipedia, we manually analyzed 201 vandalism edits in the training set of our preliminary corpus. In order to concentrate on textual vandalism detection, we did not take into account the cases where vandals hack the image, audio or other multimedia resources contained in the Wikipedia edit.

We found three main types of vandalism, which are shown in Table 1 along with corresponding examples. These examples contain both the title of the edit and a snippet of the *diff*-ed content of vandalism, which is the textual difference between the old revision and the new revision, derived through the standard *diff* algorithm (Heckel, 1978).

- **Lexically ill-formed**
  This is the most common type of vandalism in Wikipedia. Like other online vandalism acts, many vandalism cases in Wikipedia involve ill-intentioned or ill-formed words such as vulgarisms, invalid letter sequences, punctuation misuse and Web slang. An interesting observation is that vandals almost never add emoticons in Wikipedia. For the first example in

Vandalism Types	Examples
Lexically ill-formed	**Edit Title**: *IPod* shit!!!!!!!!!!!!!!!!!!!!
Syntactically ill-formed	**Edit Title**: *Rock music* DOWN WITH SOCIETY MADDISON STREET RIOT FOREVER.
	Edit Title: *Vietnam War* Crabinarah sucks dont buy it
Lexically + syntactically well-formed, semantically ill-intentioned	**Edit Title**: *Global Warming* Another popular theory involving global warming is the concept that global warming is not caused by greenhouse gases. The theory is that Carlos Boozer is the one preventing the infrared heat from escaping the atmosphere. Therefore, the Golden State Warriors will win next season.
	Edit Title: *Harry Potter* Harry Potter is a teenage boy who likes to smoke crack with his buds. They also run an illegal smuggling business to their headmaster dumbledore. He is dumb!

Table 1: Vandalism Types and Examples

Table 1, vulgarism and punctuation misuse are observed.

- **Syntactically ill-formed**

  Most vandalism cases that are lexically ill-intentioned tend to be syntactically ill-formed as well. It is not easy to capture these cases by solely relying on lexical knowledge or rule-based dictionaries and it is also very expensive to update dictionaries and rules manually. Therefore, we think that is crucial to incorporate more syntactic cues in the feature set in order to improve performance. Moreover, there are also some cases where an edit could be lexically well-intentioned, yet syntactically ill-formed. The first example of syntactic ill-formed in Table 1 is of this kind.

Feature Sets	Features
Task-specific	Number of Revisions; Revisions Size Ratio;
Lexical	Vulgarism; Web Slang; Punctuation Misuse; Comment Cue Words;
Syntactic	Normalized Topic-specific N-tag Log Likelihood and Perplexity
Semantic	Normalized Topic-specific Syntactic N-gram Log Likelihood and Perplexity

Table 2: Feature Sets and Corresponding Features of Our Vandalism Detection System

- **Lexically and syntactically well formed, but semantically ill-intentioned**

  This is the trickiest type of vandalism to identify. Vandals of this kind might have good knowledge of the rule-based vandalism detecting bots. Usually, this type of vandalism involves off-topic comments, inserted biased opinions, unconfirmed information and lobbying using very subjective comments. However, a common characteristic of all vandalism in this category is that it is free of both lexical and syntactic errors. Consider the first example of semantic vandalism in Table 1 with edit title "Global Warming": while the first sentence for that edit seems to be fairly normal (the author tries to claim another explanation of the global warming effect), the second sentence makes a sudden transition from the previous topic to mention a basketball star and makes a ridiculous conclusion in the last sentence.

In this work, we realize the importance of incorporating NLP techniques to tackle all the above types of vandalism, and our focus is on the syntactically ill-formed and semantically ill-intentioned types that could not be detected by rule-based systems and straightforward lexical features.

4 Our System

We propose a shallow syntactic-semantic focused classification approach for vandalism detection (Table 2). In contrast to previous work, our approach concentrates on the aspect of using natural language techniques to model vandalism. Our shallow syntactic and semantic modeling approaches extend the traditional n-gram language modeling method with topic-specific n-tag (Collins et al., 2005) syntax models and topic-specific syntactic n-gram semantic models. Moreover, in the Wikipedia vandalism detection task, since we do not have a sufficient amount of training data to model the topic of each edit, we propose the idea of using the Web as corpus by retrieving search engine results to learn our topic-specific n-tag syntax and syntactic n-gram semantic models. The difference between our syntactic and semantic modeling is that n-tag syntax models only model the order of sentence constituents, disregarding the corresponding words. Conversely, for our syntactic n-gram models, we do keep track of words together with their POS tags and model both the word and syntactic compositions as a sequence. The detail of our shallow syntactic-semantic modeling method will be described in subsection 4.4.

We use our shallow syntactic-semantic model to augment our base model, which builds on early work. For example, when building one of our task-specific features, we extract the name of the author of this revision to query Wikipedia about the historical behavior of this author. This kind of task-specific global feature tends to be very informative and thus forms the basis of our system. For lexical level features, we count vulgarism frequencies and also introduce three new lexical features: Web slang, punctuation misuse and comment cue words, all of which will be described in detail in 4.2 and 4.3.

4.1 Problem Representation

The vandalism detection task can be formulated as the following problem. Let's assume we have a vandalism corpus *C*, which contains a set of Wikipedia edits *S*. A Wikipedia edit is denoted as $\boldsymbol{e}_i$. In our case, we have $S = \{\boldsymbol{e}_1, \boldsymbol{e}_2 \ldots, \boldsymbol{e}_n\}$. Each edit $\boldsymbol{e}$ has two consecutive revisions (an old revision R_{old} and a new revision R_{new}) that are unique in the entire data set. We write that $\boldsymbol{e} = \{R_{old}, R_{new}\}$. With the use of the standard *diff* algorithm, we can produce a text R_{diff}, showing the difference between these two revisions, so that $\boldsymbol{e} = \{R_{old}, R_{new}, R_{diff}\}$. Our task is: given S, to extract features from edit $\boldsymbol{e} \in S$ and train a logistic boosting classifier. On receiving an edit $\boldsymbol{e}$ from the test set, the classifier needs to decide whether this $\boldsymbol{e}$ is a vandalism edit or a non-vandalism edit. $\boldsymbol{e} \rightarrow \{1,0\}$.

4.2 Basic Task-specific and Lexical Features

Task-specific features are domain-dependent and are therefore unique in this Wikipedia vandalism detection task. In this work, we pick two task-specific features and one lexical feature that proved effective in previous studies.

- **Number of Revisions**
  This is a very simple but effective feature that is used by many studies (Wilkinson and Huberman, 2007; Adler et al., 2008; Stein and Hess, 2007). By extracting the author name for the new revision R_{new}, we can easily query Wikipedia and count how many revisions the author has modified in the history.

- **Revision Size Ratio**
  Revision size ratio measures the size of the new revision versus the size of the old revision in an edit. This measure is an indication of how much information is gained or lost in the new revision R_{new}, compared to the old revision R_{old}, and can be expressed as:

$$\text{RevRatio}(\boldsymbol{e}) = \frac{\sum_{w \in R\ new} \text{Count (w)}}{\sum_{w \in R\ old} \text{Count (w)}}$$

  where W represents any word token of a revision.

- **Vulgarism Frequency**
  Revision size ratio measures the size of the new revision versus the Vulgarism frequency was first introduced by Potthast et al. (2008). However, note that not all vulgarism words should be considered as vandalism and sometime even the Wikipedia edit's title and content themselves contain vulgarism words.

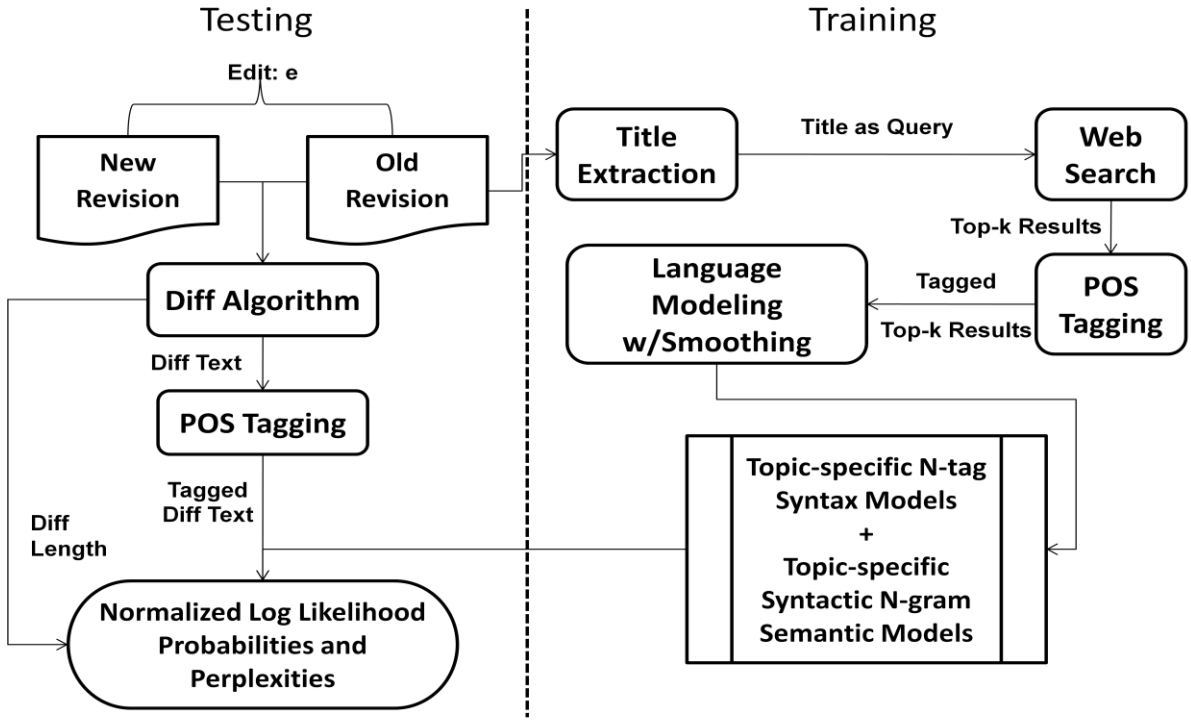

Figure 1. Topic-specific N-tag Syntax Models and Syntactical N-gram for Syntactical and Semantic Modeling

For each *diff* text in an edit ***e,*** we count the total number of appearances of vulgarism words *v* where *v* is in our vulgarism dictionary[1].

$$\text{VulFreq}(\boldsymbol{e}) = \sum_{v \in \text{Rdiff}} \text{Count}(v)$$

4.3 Novel Lexical Features

In addition to previous lexical features, we propose three novel lexical features in this paper: Web slang frequency, punctuation misuse, and comment cue words frequency.

- **Web Slang and Punctuation Misuse**

  Since Wikipedia is an open Web application, vandalism also contains a fair amount of Web slang, such as, "haha", "LOL" and "OMG". We use the same method as above to calculate Web slang frequency, using a Web slang dictionary[2]. In vandalism edits, many vandalism edits also contain punctuation misuse, for example, "!!!" and "???". However, we have not observed a significant amount of emoticons in the vandalism edits. Based on this, we only keep track of Web slang frequency and the occurrence of punctuation misuse.

- **Comment Cue Words**

  Upon committing each new revision in Wikipedia, the author is required to enter some comments describing the change. Well-intentioned Wikipedia contributors consistently use these comments to explain the motivation for their changes. For example, common non-vandalism edits may contain cue words and phrases like "edit revised, page changed, item cleaned up, link repaired or delinked". In contrast, vandals almost never take their time to add these kinds of comments. We can measure this phenomenon by counting the frequency of comment cue words.

[1] http://www.noswearing.com/dictionary
[2] http://www.noslang.com/dictionary/full

(a)
Rock/NNP and/CC roll/NN -LRB-/-LRB- also/RB spelled/VBD Rock/NNP 'n'/CC Roll/NNP
(b)
NNP CC NN -LRB- RB VBD NNP CC NNP
(c)
Rock/NNP !/. !/. !/. and/CC roll/VB you/PRP !/. !/. !/.
(d)
NNP . . . CC VB PRP . . .

Figure 2. Topic-specific N-tag and Syntactic N-gram modeling for the edit "Rock and Roll" in Wikipedia (a) The Web-derived POS tagged sequence (b) The Web-derived POS tag-only sequence (c) A POS tagged vandalism *diff* text R_{diff} (d) A POS tag-only vandalism R_{diff}

4.4 Topic-specific N-tag Syntax Models and Syntactic N-grams for Shallow Syntactic and Semantic Modeling

In Figure 1, we present the overview of our approach, which uses Web-trained topic-specific training for both: (1) n-tag syntax models for shallow syntactic modeling and (2) syntactic n-gram models for shallow semantic modeling.

For each Wikipedia edit, we consider its title as an approximate semantic representation, using it as a query to build topic-specific models. In addition, we also use the title information to model the syntax of this topic.

Given R_{diff}, we produce the syntactic version of the *diff*-ed text using a probabilistic POS tagger (Toutanova and Manning, 2000; Toutanova et al., 2003). The edit title is extracted from the corpus (either R_{new} or R_{old}) and is used to query multiple Web search engines in order to collect the n-tag and n-gram training data from the top-*k* results. Before we start training language models, we tag the top-*k* results using the POS tagger. Note that when modeling n-tag syntax models, it is necessary to remove all the words. With the POS-only sequences, we train topic-specific n-tag models to describe the syntax of normal text on the same topic associated with this edit. With the original tagged sequences, we train syntactic n-gram models to represent the semantics of the normal text of this edit.

After completing the training stage, we send the test segment (i.e. the *diff*-ed text sequence) to both the learned n-tag syntax models and the learned syntactic n-gram models. For the n-tag syntax model, we submit the POS tag-only version of the segment. For the syntactic n-gram model, we submit a version of the segment where each original word is associated with its POS-tag. In both cases we compute the log-likelihood and the perplexity of the segment.

Finally, we normalize the log likelihood and perplexity scores by dividing them by the length of R_{diff}, as this length varies substantially from one edit to another. [3] We expect an edit that has low log likelihood probability and perplexity to be vandalism, and it is very likely to be unrelated to the syntax and semantic of the normal text of this Wikipedia edit. In the end, the normalized log probability and perplexity scores will be incorporated into our back-end classifier with all task-specific and lexical features.

Web as Corpus: In this work, we leverage Web search results to train the syntax and semantic models. This is based on the assumption that the Web itself is a large corpus and Web search results can be a good training set to approximate the semantics and syntax of the query.

Topic-specific Modeling: We introduce a topic-specific modeling method that treats every edit in Wikipedia as a unique topic. We think that the title of each Wikipedia edit is an approximation of the topic of the edit, so we extract the title of each edit and use it as keywords to retrieve training data for our shallow syntactic and semantic modeling.

Topic-specific N-tag and Syntactic N-gram: In our novel approach, we tag all the top-*k* query results and *diff* text with a probabilistic POS tagger in both the training and test set of the vandalism corpus. Figure 2(a) is an example of a POS-tagged sequence in a top-*k* query result.

For shallow syntactic modeling, we use an n-tag modeling method (Collins et al., 2005). Given a tagged sequence, we remove all the words and only keep track of its POS tags: $tag_{i-2}\ tag_{i-1}$

[3] Although we have experimented with using the length of R_{diff} as a potential feature, it does not appear to be a good indicator of vandalism.

tag_i. This is similar to n-gram language modeling, but instead, we model the syntax using POS tags, rather than its words. In this example, we can use the system in Figure 2 (b) to train an n-tag syntactic model and use the one in Figure 2 (d) to test. As we see, for this test segment, it belongs to the vandalism class and has very different syntax from the n-tag model. Therefore, the normalized log likelihood outcome from the n-tag model is very low.

In order to model semantics, we use an improved version of the n-gram language modeling method. Instead of only counting $word_{i-2}$ $word_{i-1}$ $word_i$, we model composite tag/word feature, e.g. $tag_{i-2}word_{i-2}$ $tag_{i-1}word_{i-1}$ tag_iword_i. This syntactic n-gram modeling method has been successfully applied to the task of automatic speech recognition (Collins et al., 2005). In the example in Figure 2, the vandalism *diff* text will probably score low, because although it shares an overlap bigram "*and roll*" with the phrase "*rock and roll*" in training text, once we apply the shallow syntactic n-gram modeling method, the POS tag bigram "*and/CC roll/VB*" in *diff* text will be distinguished from the "*and/CC roll/NN*" or "*and/CC roll/NNP*" in the training data.

5 Experiments

To evaluate the effectiveness of our approach, we first run experiments on a preliminary corpus that is also used by previous studies and compare the results. Then, we conduct a second experiment on a larger corpus and analyze in detail the features of our system.

5.1 Experiment Setup

In our experiments, we use a Wikipedia vandalism detection corpus (Potthast et al., 2007) as a preliminary corpus. The preliminary corpus contains 940 human-assessed edits from which 301 edits are classified as vandalism. We split the corpus and keep a held-out 100 edits for each class in testing and use the rest for training. In the second experiment, we adopt a larger corpus (Potthast et al., 2010) that contains 15,000 edits with 944 marked as vandalism. The split is 300 edits for each class in held-out testing and the rest used for training. In the description of the second corpus, each edit has been reviewed by at least 3 and up to 15 annotators. If more than 2/3 of the annotators agree on a given edit, then the edit is tagged as one of our target classes. Only 11 cases are reported where annotators fail to form a majority inter-labeler agreement and in those cases, the class is decided by corpus authors arbitrarily.

In our implementation, the Yahoo![4] search engine and Bing[5] search engine are the source for collecting top-*k* results for topic-specific n-gram training data, because Google has a daily query limit. We retrieve top-100 results from Yahoo!, and combine them with the top-50 results from Bing.

For POS tagging, we use the Stanford POS Tagger (Toutanova and Manning, 2000; Toutanova et al., 2003) with its attached wsj3t0-18-bidirectional model trained from the Wall Street Journal corpus. For both shallow syntactic and semantic modeling, we train topic-specific trigram language models on each edit using the SRILM toolkit (Stolcke, 2002).

In this classification task, we used two logistic classification methods that haven't been used before in vandalism detection. Logistic model trees (Landwehr et al., 2005) combine tree induction with linear modeling. The idea is to use the logistic regression to select attributes and build logistic regression at the leaves by incrementally refining those constructed at higher levels in the tree. The second method we used, logistic boosting (Friedman et al., 2000), improves logistic regression with boosting. It works by applying the classification algorithm to reweighted versions of the data and then taking a weighted majority vote of the sequence of classifiers thus produced.

5.2 Preliminary Experiment

In the preliminary experiment, we tried logistic boosting classifiers and logistic model trees as classifiers with 10-fold cross validation. The rule-based method, ClueBot, is our baseline.

We also implemented another baseline system, using the bag of words (BoW) and Naive Bayes method (Smets et al., 2008) and the same toolkit (McCallum, 1996) that Smets et al. used. Then, we compare our result with Potthast et al. (2008), who used the same corpus as us.

[4] http://www.yahoo.com
[5] http://www.bing.com

Systems	Recall	Precision	F1
ClueBot	0.27	**1**	0.43
BoW + Naïve Bayes	0.75	0.74	0.75
Potthast et. al., 2008	0.77	0.83	0.80
Task-specific +Lexical (LMT)	0.87	0.87	0.87
Task-specific +Lexical (LB)	0.92	0.91	0.91
Our System (LMT)	0.89	0.89	0.89
Our System (LB)	**0.95**	0.95	**0.95**

Table 3: Preliminary Experiment Results; The acronyms: BoW: Bag of Words, LMT: Logistic Model Trees, LB: Logistic Boosting, Task-specific + Lexical: features in section 4.1 and 4.2

As we can see in Table 3, the ClueBot has a F-score (F1) of 0.43. The BoW + Naïve Bayes approach improved the result and reached an F1 of 0.75. Compared to these results, the system of Potthast et al. (2008) is still better and has a F1 of 0.80.

For the results of our system, LMT gives us a 0.89 F1 and LogitBoost (LB) gives a 0.95 F1. A significant F1 improvement of 15% was achieved in comparison to the previous study (Potthast et al., 2008). Another finding is that we find our shallow syntactic-semantic modeling method improves 2-4% over our task-specific and lexical features.

5.3 Results and Analysis

In the second experiment, a notable difference from the preliminary evaluation is that we have an unbalanced data problem. So, we use random down-sampling method to resample the majority class into balanced classes in the training stage. Then, we also use the two classifiers with 10-fold cross validation.

The F1 result reported by our BoW + Naïve Bayes baseline is 0.68. Next, we test our task-specific and lexical features that specified in section 4.1 and 4.2. The best result is a F1 of 0.82, using logistic boosting. Finally, with our topic-specific shallow syntactic and semantic modeling features, we have a precision of 0.86, a recall of 0.85 and F1 of 0.85.

Features	Recall	Precision	F1
BoW + Naïve Bayes	0.68	0.68	0.68
Task-specific (LMT)	0.81	0.80	0.80
Task-specific +Lexical(LMT)	0.81	0.81	0.81
Our System (LMT)	0.84	0.83	0.83
Task-specific (LB)	0.81	0.80	0.80
Task-specific + Lexical (LB)	0.83	0.82	0.82
Our System (LB)	**0.86**	**0.85**	**0.85**

Table 4: Second Experiment Results

Though we are surprised to see the overall F1 for the second experiment are not as high as the first one, we do see that the topic-specific shallow syntactic and semantic modeling methods play an important role in improving the result.

Looking back at the related work we mentioned in section 2, though we use newer data sets, our overall results still seem to surpass major vandalism detection systems.

6 Conclusion and Future Works

We have described a practical classification framework for detecting Wikipedia vandalism using NLP techniques and shown that it outperforms rule-based methods and other major machine learning approaches that are previously applied in the task.

In future work, we would like to investigate deeper syntactic and semantic cues to vandalism. We hope to improve our models using shallow parsing and full parse trees. We may also try lexical chaining to model the internal semantic links within each edit.

Acknowledgements

The authors are grateful to Julia Hirschberg, Yves Petinot, Fadi Biadsy, Mukund Jha, Weiyun Ma, and the anonymous reviewers for useful feedback. We thank Potthast et al. for the Wikipedia vandalism detection corpora.

References

Adler, B. Thomas, Luca de Alfaro, Ian Pye and Vishwanath Raman. 2008. Measuring Author Contributions to the Wikipedia. In *Proc. of the ACM 2008 International Symposium on Wikis*.

Biadsy, Fadi, Julia Hirschberg, and Elena Filatova. 2008. An Unsupervised Approach to Biography Production using Wikipedia. In *Proc. of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 807–815.

Bratko, Andrej, Gordon V. Cormack, Bogdan Filipic,Thomas R. Lynam and Blaz Zupan. 2006. Spam Filtering Using Statistical Data Compression Models. *Journal of Machine Learning Research*, pages 7:2673-2698.

Collins, Michael, Brian Roark and Murat Saraclar. 2005. Discriminative Syntactic Language Modeling for Speech Recognition. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics*. pages 507–514.

Friedman, Jerome, Trevor Hastie and Robert Tibshirani. 2000. Additive Logistic Regression: a Statistical View of Boosting. *Annals of Statistics* 28(2), pages 337-407.

Geiger, R. Stuart. 2010. The Work of Sustaining Order in Wikipedia: The Banning of a Vandal. In *Proc. of the 2010 ACM Conference on Computer Supported Cooperative Work*, pages 117-126.

Heckel, Paul. 1978. A Technique for Isolating Differences Between Files. *Communications of the ACM*, pages 264–268

Itakura, Kelly Y. and Charles L. A. Clarke. 2009. Using Dynamic Markov Compression to Detect Vandalism in the Wikipedia. In *Proc. of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 822-823.

Landwehr, Niels, Mark Hall and Eibe Frank. 2005. Logistic Model Trees. *Machine Learning*, 59(1-2), pages 161–205.

McCallum, Andrew. 1996. Bow: a Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering.

Potthast, Martin, Benno Stein, and Robert Gerling. 2008. Automatic Vandalism Detection in Wikipedia. In *Proc. of the 30th European Conference on Information Retrieval, Lecture Notes in Computer Science*, pages 663-668.

Potthast, Martin and Robert Gerling. 2007. Wikipedia Vandalism Corpus WEBIS-VC07-11. *Web Technology & Information Systems Group, Bauhaus University Weimar*.

Potthast, Martin, Benno Stein and Teresa Holfeld. 2010. PAN Wikipedia Vandalism Training Corpus PAN-WVC-10. *Web Technology & Information Systems Group, Bauhaus University Weimar*.

Rigoutsos, Isidore and Tien Huynh. 2004. Chung-Kwei: a pattern-discovery-based system for the automatic identification of unsolicited e-mail messages (SPAM). In *Proc. of the First Conference on E-mail and Anti-Spam.*

Smets, Koen, Bart Goethals and Brigitte Verdonk. 2008. Automatic Vandalism Detection in Wikipedia: Towards a Machine Learning Approach In *Proc. of AAAI '08, Workshop on Wikipedia and Artificial Intelligence*, pages 43-48.

Stein, Klaus and Claudia Hess. 2007. Does It Matter Who Contributes: a Study on Featured Articles in the German Wikipedia. In *Proc. of the ACM 18th Conference on Hypertext and Hypermedia*, pages 171–174.

Stolcke, Andreas. 2002. SRILM – an extensible language modeling toolkit. In *Proc. of the International Conference on Spoken Language Processing*, volume 2, pages 901–904.

Toutanova, Kristina and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63-70.

Toutanova, Kristina, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of Human Language Technology Conference and the North American Chapter of the Association of Computational Linguistics Series*, pages 252-259.

Wilkinson,Dennis and Bernardo Huberman. 2007. Cooperation and Quality in Wikipedia. In *Proc. of the ACM 2007 International Symposium on Wikis*, pages 157–164.