Using Dynamic Markov Compression to Detect Vandalism in the Wikipedia

Kelly Y. Itakura
David R. Cheriton School of Computer Science
University of Waterloo
200 University Ave. W.
Waterloo, ON, Canada
yitakura@cs.uwaterloo.ca

Charles L.A. Clarke
David R. Cheriton School of Computer Science
University of Waterloo
200 University Ave. W.
Waterloo, ON, Canada
claclark@uwateloo.ca

ABSTRACT

We apply the Dynamic Markov Compression model to detect spam edits in the Wikipedia. The method appears to outperform previous efforts based on compression models, providing performance comparable to methods based on manually constructed rules.

Categories and Subject Descriptors

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

General Terms

Security, Experimentation, Human Factors

Keywords

Information retrieval, spam filtering

1. INTRODUCTION

The Wikipedia [6] allows any user to contribute and edit articles. This freedom sometimes invites *vandalism*, defined by the Wikipedia as "a deliberate attempt to compromise the integrity" of its content. In this paper, we refer to a set of vandalism in a revision of a given page as a *spam edit*, contrasting with an attempt to improve the content, which we refer to as a *ham edit*.

The problem of spam detection in the Wikipedia differs from other spam filtering tasks, such as email and Web spam filtering. Email spam consists of entire messages; Web spam consists of entire pages. To some extent, email and Web spam has predictable content (offers for prescription drugs or easy money). Spam detection in the Wikipedia, on the other hand, requires us to spot parts of an article that are spam, that may consist of nothing more than a few words or characters. For example, an article may get vandalized by the addition of a nonsensical sequence of characters such as "lolololol". An identical edit may never appear again.

In this paper, we employ a simple algorithm, based on a compression model, to detect spam edits in the Wikipedia. Similar models are widely employed for email and web spam filtering. Our algorithm performs comparable to that of Potthast et al. [4], who used machine learning in combination with manually crafted rules to classify spam edits. Perhaps more surprisingly, our algorithm appears to outperform similar work, based on a different compression model, conducted by Smets et al. [5]. In that work, they classified entire revisions, which could contain many types of vandalism, instead of focusing on individual types of vandalism, as we do. We also approach training differently, having a balanced corpus. In contrast to the conclusions reached by Smets et al., we show that a simple compression-based algorithm works reasonably well, and could provide the foundation for an operational spam filter.

2. RELATED WORK

Dynamic Markov Compression (DMC) [3] is a compression algorithm that compresses a file by predicting the next bit based on previously seen bits. DMC has been used successfully for classifying email and Web spam [1]. For classification, separate compression models are built for ham and spam. The model that achieves the greatest compression ratio over new data indicates the class of that data.

Little formal research has been conducted on the identification of vandalism in the Wikiepdia. Potthast et al. [4] use logistic regression with 16 manually crafted features, to categorize an edit as spam or ham. Smets et al. [5] used the Prediction by Partial Match (PPM) compression model to classify an entire revision of a page in the Wikipedia. Using the Simple English Wikipedia as corpus, they summed probabilities of a revised page containing various types of spam. They also included user groups and revision comments as additional features, but removed Wikipedia tags from edits.

3. EXPERIMENTS

Instead of classifying entire revision pages, which could include multiple types of spam, our goal is to determine if we can identify two types of spam edits, insertion and change edits, separately. We used the standard English Wikipedia, used DMC as our predictor, and used edits from the revision history for training. This training set contains an equal numbers of spam and ham edits. For training, an edit that was later reverted by a user by removal or change is treated as spam, and an edit that is reverted to by addition or change is treated as ham. These edits were collected from the history pages of randomly selected Wikipedia ar-

Copyright is held by the author/owner(s).
SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
ACM 978-1-60558-483-6/09/07.

Experiment	Spam Precision	Spam Recall	Ham Precision	Ham Recall	Spam BEP	Ham BEP
insertion	86.56	69.65	74.61	89.19	79.30	78.26
change	74.07	53.33	63.54	81.33	65.28	65.28

Table 1: Precision, Recall, and Break Even Point (BEP) for Spam and Ham

ticles (using the Wikipedia's "Random article" feature). We retained Wikipedia tags in an edit, such as "[[link]]" for internal links.

Our task is to categorize two kinds of spam edits in the Wikipedia: *insertion* spam and *change* spam. The former is spam introduced by inserting inappropriate text into an article, which was later reverted by another user. The latter modifies parts of sentences or paragraphs of the previous revision, which were later reverted by another user.

Our training set for insertion edits comprises 5768 edits each for both ham and spam data, but in some experiments we varied the ratio of spam to ham training data to determine its effect on performance. The size of our insertion edit test set comprises 481 edits for both spam and ham. Our training set for change edits comprises 5768 edits each for both spam and ham. The test set comprises 600 edits each.

For all the experiments, we concatenated the spam and ham training sets into single files, and attached to the end of each file a test edit that could be either spam or ham. We computed the compression ratio using Cormack's implementation of DMC [2] and judged the test edit to be spam if the compression ratio with the spam training set was higher than the compression ratio with the ham training set. Moreover, we changed the threshold x such that the spam compression ratio must be higher than x times the ham compression ratio to be classified as spam. Adjusting x allowed us to compute a recall precision break-even point (BEP) for each collection.

4. RESULTS

Table 1 show the results of our experiments. Precision and recall values were calculated for both spam and ham. Spam precision is computed as the number of spam edits correctly identified as spam over all edits identified as spam. Spam recall is computed as the number of spam edits correctly identified as spam over all spam edits. Ham precision and recall values are computed similarly. The break even points were obtained by varying the threshold x, the spam to ham compression ratio, to categorize a test edit as spam.

5. DISCUSSION

The results indicate that identifying insertion spam is reasonably effective. On the other hand, change spam seems harder to identify. This may be because change edits tend to contain chunks of non-consecutive phrases that were changed and may need some surrounding context to improve effectiveness.

Our performance may be compared with the results reported in Smets et al. [5]. Overall, our results appear to outperform theirs, even those that use probabilistic sequence modeling. Their precision and recall values for classifying revision pages using only insertions models are 12.74% and 92.81%, and those using only change models are 11.77% and 83.62%. When they combined various predictors to judge if a revision page is a spam or not, their reported precision is 32.09% and their reported recall is 91.71%. However, we did not attempt to reproduce their results, and we may have misinterpreted some aspects of their experiments. Nonetheless, in contrast to their conclusions, our results suggest that compression models may provide a useful method for identifying spam edits.

Our differing performance might stem from a more balanced training corpus than that used by Smets et al. [5]. When we varied the ratio of spam to ham in the training set, the optimum performance was obtained by balanced ratios.

Our performance may also be compared with that of Potthast [4]. We slightly underperform their approach. However, we demonstrate that compression algorithms alone can compete reasonablly well against classifiers that use hand crafted features.

6. CONCLUSIONS

In this paper, we employed the DMC compression model to classify new edits in Wikipedia articles as spam or ham. In contrast to previous work, our results suggest that compression models might provide a useful method for classifying these edits. Future work might combine the results of compression models with other features and methods. It may also be possible to vary training methods, or increase the size of the training sets, to obtain additional improvements.

7. REFERENCES

[1] A. Bratko, B. Filipič, G. V. Cormack, T. R. Lynam, and B. Zupan. Spam filtering using statistical data compression models. *J. Mach. Learn. Res.*, 7:2673–2698, 2006.

[2] G. V. Cormack. Dynamic markov compression (dmc) version 0.0.0, 1993/1987. [Online; accessed 09-February-2009].

[3] G. V. Cormack and R. N. S. Horspool. Data compression using dynamic markov modelling. *Comput. J.*, 30(6):541–550, 1987.

[4] M. Potthast, B. Stein, and R. Gerling. Automatic vandalism detection in wikipedia. *Advances in Information Retrieval*, 4956:663–668, 2008.

[5] 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.

[6] Wikipedia. Wikipedia — Wikipedia, the free encyclopedia, 2009. [Online; accessed 25-January-2009].