Wikipedia Vandalism Detection

Feature Review and New Proposals

Santiago M. Mola Velasco
<sanmove@posgrado.upv.es>

4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse

Outline

- The proposed system is based on previous work in (Potthast, Stein and Gerling, 2008).
- A number of very simple features are extracted from each edit.
- The usual suspects on supervised learning are employed.

A beautiful place...

...and some words by Cucciolo

Features I

- **Anonymous** Whether the editor is anonymous or not.
- **Comment length** Length in characters of the edit summary.
- **Upper to lower ratio** *(new)* Uppercase to lowercase ratio of inserted text.
- **Uppercase ratio** Uppercase to all characters ratio of inserted text.
- **Digit ratio** *(new)* Digit to all characters ratio of inserted text.
- **Non-alphanumeric ratio** *(new)* Non-alphanumeric to all characters ratio of inserted text.
- **Character diversity** *(new)* Measure of different characters compared to length of inserted text. $\text{length}^{\frac{1}{\text{different chars}}}$

Features II

Character distribution$^{(mod)}$ Kullback-Leibler divergence of the character distribution of the inserted text and the expectation.

Compressibility$^{(mod)}$ Compression (using LZW) rate of inserted text.

Size increment$^{(new)}$ Absolute increment of size.

Size ratio Size of the new revision relative to the old revision.

Average term frequency Average relative frequency of inserted words in the new revision.

Longest word Length of the longest word in inserted text.

Longest character sequence Longest consecutive sequence of the same character in inserted text.

Features III

Feature	Info gain ratio
Anonymous	0.06797
Character distribution	0.03714
Character diversity	0.0302
Upper to lower ratio	0.02874
Non-alphanumeric ratio	0.02699
Digit ratio	0.02352
Longest character seq.	0.0233
Average term frequency	0.02325
Uppercase ratio	0.0206
Longest word	0.02023
Size increment	0.01789
Compressibility	0.01577
Size ratio	0.01313
Comment length	0.00943

Features: Word list based I

- By now, we have already seen most of features, but there is something missing...
- ***"If you want to detect a vandal, you have to think like a vandal."***

Think like a vandal

Figure: Source: Banksy. More at http://banksy.co.uk.

Features: Word list based II

- Potthast *et al.* proposed two features based on lists of words: *frequency* and *impact*.
- Applied to vulgarisms and first and second personal pronouns.
- ClueBot also uses a list of 40 regular expressions detecting vulgar words.

Features: Word list based III

- We propose to define both frequency and impact for the following categories:

  Vulgarisms Vulgar words (e.g. fuck, suck, dick, pussy).

  Pronouns First and second person pronouns, including slang (e.g. you, ya, I'm, ima).

  Bad words Slang words and typos (e.g. dont, dosent, guise, wanna, gonna, dunno).

  Biased Biased words (e.g. everyone, cares, coolest, huge, ever).

  Sex Non-vulgar sex-related words (e.g. sex, penis, vagina).

  Good words Words or tokens uncommon in vandalism (e.g. infobox, category, {{).

Features: Word list based IV

Feature	Info gain ratio
Vulgarism frequency	0.4361
All frequency	0.33688
Bad word frequency	0.27337
Vulgarism impact	0.25837
Sex frequency	0.24745
Pronoun frequency	0.24006
Sex impact	0.21582
Biased frequency	0.21067
All impact	0.19395
Bad word impact	0.15263
Pronoun impact	0.07376
Biased impact	0.07195
Goodword impact	0.02144
Goodword frequency	0.01803

Features: Word list based V

- Dividing words in different categories allows learning algorithms to weight them.
- Using categories understandable by humans, as opposed to data-driven categories, allows us to extend them based on our knowledge.

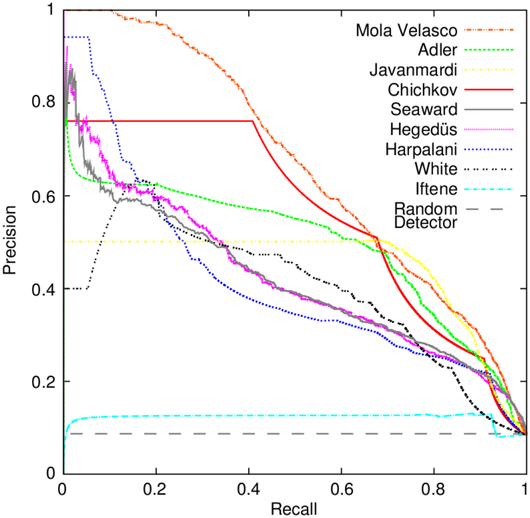
Classifiers I

- We used the Weka framework to build and evaluate diverse learning algorithms.
- We chose classifiers that either are very robust to noisy data (C4.5, Random Forests), do implicit feature selection (LogitBoost, Random Forests) or are resistant to class imbalance (SVM).
- In an attempt to mitigate class imbalance, experiments have been repeated with the corpus modified giving 10 times more weight to the vandalism class.

Classifiers II

Classifier	Precision	Recall	F-Measure	AUC
C4.5	0.739	0.529	0.617	0.928
LogitBoost	0.84	0.564	0.675	**0.966**
Random Forests	0.849	0.564	**0.678**	0.96
SVM	0.837	0.373	0.516	0.939
C4.5 + weight	0.424	0.775	0.548	0.947
LogitBoost + weight	0.43	0.852	0.571	0.**963**
Random Forests + w.	0.756	0.647	**0.697**	0.96
SVM + weight	0.321	0.88	0.47	0.949

PAN’10 Results



Some Remarks

- The proposed features, at detection time, don't need any data external to the edit itself.
- This makes the system really *fast* and *cheap*.
- But as popular culture says:
  "Fast, cheap and reliable, pick two."
- Further work should include the use of external sources (e.g. the Wikipedia database) to check semantic and pragmatic characteristics of the edit.

Conclusions

- Our approach can achieve precision near to 1 with recall around 0.2, so it could work autonomously.
- Or high recall and be used in a two-stage process, with human review.
- There are yet many features to explore. Specially regarding style, structure and context.

Thank you

Thank you.
Questions?

Figure extracted from
http://www.sinsign.com/.
Original source unknown (to me).