

Wikipedia Vandalism Detection

by

Santiago M. Mola-Velasco

Supervised by

Paolo Rosso

A M.Sc. Thesis Presented in Partial Fulfillment  
of the Requirements for the Degree  
Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Departamento de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

September 2011

## Abstract

Wikipedia is an online encyclopedia that anyone can edit. The fact that there are almost no restrictions to contributing content is at the core of its success. However, it also attracts pranksters, lobbyists, spammers and other people who degrades Wikipedia's contents. One of the most frequent kind of damage is vandalism, which is defined as any bad faith attempt to damage Wikipedia's integrity.

For some years, the Wikipedia community has been fighting vandalism using automatic detection systems. In this work, we develop one of such systems, which won the 1st International Competition on Wikipedia Vandalism Detection. This system consists of a feature set exploiting textual content of Wikipedia articles. We performed a study of different supervised classification algorithms for this task, concluding that ensemble methods such as Random Forest and LogitBoost are clearly superior.

After that, we combine this system with two other leading approaches based on different kind of features: metadata analysis and reputation. This joint system obtains one of the best results reported in the literature. We also conclude that our approach is mostly language independent, so we can adapt it to languages other than English with minor changes.

## Resumen

Wikipedia es una enciclopedia en línea que cualquiera puede editar. El hecho que de apenas hay restricciones para contribuir contenido está en el corazón de su éxito. Sin embargo, esto también atrae a bromistas, cabilderos, *spammers* y otras personas que degradan los contenidos de Wikipedia. Uno de los tipos de daño más frecuente es el vandalismo, definido como cualquier intento, de mala fe, de dañar la integridad de Wikipedia.

Desde hace algunos años, la comunidad de Wikipedia ha estado luchando contra el vandalismo usando sistemas automáticos de detección. En este trabajo, desarrollamos uno de estos sistemas, que ganó la Primera Competición Internacional de Detección de Vandalismo en Wikipedia. Este sistema consiste en un conjunto de características que explotan el contenido textual de los artículos de Wikipedia. Realizamos un estudio de diferentes algoritmos de clasificación supervisada para esta tarea, concluyendo que los métodos de ensamble como *Random Forest* y *LogitBoost* son claramente superiores.

Después, combinamos este sistema con otras dos aproximaciones punteras basadas en distintos tipos de características: análisis de metadatos y reputación. Este sistema conjunto obtiene uno de los mejores resultados publicados en la literatura. También concluimos que nuestra aproximación es principalmente independiente del lenguaje, por lo que podemos adaptarlo a idiomas distintos al inglés con cambios menores.

# Contents

<b>Contents</b>	<b>iv</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 What is Wikipedia . . . . .	1
1.2 Wikis, MediaWiki and Wikipedia . . . . .	2
1.3 What is Vandalism and Why Does It Matter . . . . .	8
1.4 Organization of This Thesis . . . . .	9
<b>2 Wikipedia Vandalism</b>	<b>11</b>
2.1 Kinds of Vandalism . . . . .	11
2.2 Vandalism Statistics and Impact . . . . .	13
2.2.1 Vandalism Statistics . . . . .	13
2.2.2 Vandalism Impact: An Anecdote . . . . .	13
<b>3 Wikipedia Vandalism Detection</b>	<b>15</b>
3.1 Practical Tools Against Vandalism . . . . .	15
3.1.1 Anti-vandalism Patrolling . . . . .	15
3.1.2 Patrolling Assistance . . . . .	15
3.1.3 Automatic Systems, Bots and Edit Filters . . . . .	16
3.2 Problem Definition and Notation . . . . .	16
3.2.1 Immediate and Historic Detection . . . . .	17
3.3 Performance Measures . . . . .	17
3.4 Corpora . . . . .	19

3.4.1	Webis-WVC-07 . . . . .	19
3.4.2	Chin 2010 . . . . .	19
3.4.3	West 2010 . . . . .	19
3.4.4	PAN-WVC-10 . . . . .	20
3.4.5	PAN-WVC-11 . . . . .	21
3.4.6	ClueBot-NG dataset . . . . .	21
3.4.7	Wikipedia dumps . . . . .	22
3.4.8	Wikipedia User Contribution Dataset . . . . .	22
3.5	Related Work . . . . .	23
3.5.1	First Generation . . . . .	23
3.6	Second Generation . . . . .	24
3.6.1	Textual and Simple Metadata-based Features . . . . .	24
3.6.2	Compression Models . . . . .	25
3.6.3	Topic Modeling . . . . .	26
3.6.4	Article History Modeling . . . . .	26
3.6.5	Content-based Reputation . . . . .	26
3.6.6	Spatio-temporal Analysis of Metadata . . . . .	27
<b>4</b>	<b>Developing a Wikipedia Vandalism Detection System</b>	<b>31</b>
4.1	Participation at PAN 2010 . . . . .	31
4.1.1	Preprocessing . . . . .	31
4.1.2	Features . . . . .	32
4.1.3	Classification . . . . .	35
4.1.4	Evaluation . . . . .	36
4.2	Combining Natural Language, Metadata, and Reputation . . . . .	39
4.2.1	Features . . . . .	39
4.2.2	Evaluation . . . . .	44
4.2.3	Results and Discussion . . . . .	44
4.2.4	Conclusions . . . . .	57
<b>5</b>	<b>Conclusions</b>	<b>59</b>
5.1	Contributions . . . . .	59
5.2	Future Work . . . . .	59
<b>A</b>	<b>Publications</b>	<b>65</b>

# List of Tables

2.1	Summary of vandalism types by Wikipedia contributors (Wikipedia contributors 2010). Some categories outside the scope of this work have been left out of the table (malicious account creation, abuse of tags, avoidant vandalism, repeated upload of copyrighted material, gaming the system, talk page vandalism, user and user talk page vandalism, and vandabots). Table originally composed for (Mola-Velasco 2011). . . . .	12
3.1	Confusion matrix example. . . . .	18
3.2	Features used by Potthast, Stein, and Gerling (2008). Table extracted from (Potthast, Stein, and Gerling 2008, p. 3). . . . .	25
4.1	Summary of features used in Mola-Velasco 2010. . . . .	35
4.2	Performance of classifiers using (Mola-Velasco 2010). . . . .	37
4.3	Comprehensive listing of features used, organized by class. Note that features in the “!Z” (not zero-delay) class are those that are only appropriate for historical vandalism detection. In the SRC column, A stands for (Adler, Alfaro, and Pye 2010), M for (Mola-Velasco 2010) and W for (West, Kannan, and Lee 2010). Extrated from (Adler et al. 2011). . . . .	43
4.4	Performance of all feature combinations for immediate detection. . . .	46
4.5	Performance of all feature combinations for historic detection. . . . .	46

# List of Figures

1.1	Screenshot of the beginning of the Edsger W. Dijkstra article in Wikipedia.	4
1.2	Screenshot of the edit page of the Edsger W. Dijkstra article in Wikipedia.	5
1.3	Screenshot of the revision history of the Edsger W. Dijkstra article in Wikipedia. . . . .	6
1.4	Screenshot of a diff page corresponding to an edit on the Edsger W. Dijkstra article in Wikipedia. . . . .	7
1.5	Screenshot of the categories at the bottom of the Edsger W. Dijkstra article in Wikipedia. . . . .	7
1.6	Screenshot of a diff for a vandalism edit on the Edsger W. Dijkstra article in Wikipedia. . . . .	9
1.7	Screenshot of part of the Edsger W. Dijkstra article in Wikipedia, after vandalism. . . . .	9
2.1	<i>Blastoise attacks in Kaster, Belgium</i> , image depicting the incident in Kaster's Wikipedia article. . . . .	14
4.1	Precision-Recall curves for (Mola-Velasco 2010). . . . .	37
4.2	F-Measure curves for (Mola-Velasco 2010). . . . .	38
4.3	Precision-Recall curves for Logit Boost and Random Forest using all features, both for immediate and historic detection. . . . .	47
4.4	F-Measure curves for Logit Boost and Random Forest using all features, both for immediate and historic detection. . . . .	48
4.5	Precision-Recall curves for different systems in immediate detection, using Random Forest. . . . .	49
4.6	Precision-Recall curves for different systems in historic detection, using Random Forest. . . . .	50

4.7	F-Measure curves for different systems in immediate detection, using Random Forest. . . . .	51
4.8	F-Measure curves for different systems in historic detection, using Random Forest. . . . .	52
4.9	Precision-Recall curves for different feature classes in immediate detection, using Random Forest. . . . .	53
4.10	Precision-Recall curves for different feature classes in historic detection, using Random Forest. . . . .	54
4.11	F-Measure curves for different feature classes in immediate detection, using Random Forest. . . . .	55
4.12	F-Measure curves for different feature classes in historic detection, using Random Forest. . . . .	56



# Acknowledgements

I would like to thank Paolo Rosso for supervising my research; Sandra García Blasco for this one year of feedback and support on this research; Alberto Barrón Cedeño for repeatedly reviewing my papers and presentations.

Thanks to Andrew West, Thomas B. Adler and Luca de Alfaro for their time and hard work on our joint paper and Sampath Kannan, Insup Lee and Ian Pye for their previous work that made it possible.

Thanks to Martin Potthast for his help and feedback, as well as Teresa Holfeld, Benno Stein and Robert Gerling for their previous work on Wikipedia vandalism detection, which is the basis of this thesis.

Thanks to Sobha L. Devi, and her PhD students Pattabhi R. K. Rao and Vijay Sundar Ram R. for inviting and making my visit AU-KBC Research Centre a great experience. Thanks to Vasudeva Varma for his invitation and help for my visit to IIIT Hyderabad too.

Thanks to ClueBot-NG authors for their assistance in their IRC channel. As well as all those helpful IRC users hanging around in `#wikipedia`, `#wikipedia-es`, `#wikipedia-fr`, `#wikipedia-pl` and `#wikipedia-de` at `irc.freenode.net`.

Thanks to all free software developers who kindly produced all the software used in this research. Specially Mark Hall and other Weka developers and users that have directly or indirectly helped me to take the most out of this great data mining framework.

Thanks to Odri and my family, for bearing with me during deadlines



# Chapter 1

## Introduction

### 1.1 What is Wikipedia

Wikipedia's promise is nothing less than the liberation of human knowledge – both by incorporating all of it through the collaborative process, and by freely sharing it with everybody who has access to the internet. This is a radically popular idea.

— *The Economist*, 20 April 2006

Wikipedia is an online encyclopedia that is **free, collaborative, multilingual** and **global-scale**. *Free* because anyone is free to use, copy, redistribute and modify Wikipedia content, even with commercial purposes, as long as the result is also shared with the same license<sup>1</sup>. *Collaborative* because Wikipedia contents are created by the collaboration of thousands of individuals; Anyone can edit Wikipedia, even without being registered, and participate in the discussions about content and policies. *Multilingual* because there are editions of Wikipedia in 240 languages and growing. *Global-scale* because in its 10 years of life, Wikipedia has had an enormous growth. Today, it is the most popular source of encyclopedic knowledge and one of the most visited websites on the Internet, with 365 million estimated readers. Only the English edition contains more than 3 million articles, over 13 million registered users and 130 thousand active users<sup>2</sup>.

---

<sup>1</sup>Most of its content is licensed under the terms of Creative Commons Attribution/Share-Alike 3.0 and the GNU Free Documentation License. More information about Wikipedia license is available at <http://en.wikipedia.org/wiki/Wikipedia:Copyrights>.

<sup>2</sup>More statistics about Wikipedia available at <http://en.wikipedia.org/wiki/Wikipedia:Statistics> and <http://en.wikipedia.org/wiki/Special:>

The net result of Wikipedia contributors' work goes far beyond the Wikipedia project itself. There are ongoing efforts to create offline and printed editions for use in educational projects, specially in developing countries; most notably, the Wikipedia 1.0 project<sup>3</sup>. It is also used as a source of information in several projects, ranging from knowledge databases such as DBpedia<sup>4</sup> to definitions for dictionaries<sup>5</sup>. In short, the success of Wikipedia is also a key factor for the development of a wide range of academic, social and commercial projects beyond Wikipedia.

## 1.2 Wikis, MediaWiki and Wikipedia

A wiki (pronounced /'wiki/ wik-ee) is a website that allows the creation and editing of any number of interlinked web pages via a web browser using a simplified markup language or a WYSIWYG<sup>6</sup> text editor. Wikis are typically powered by wiki software and are often used collaboratively by multiple users.

Wiki — Wikipedia, The Free Encyclopedia (Wikipedia 2011)

In a wiki, collaborators can edit all the wiki pages. Wiki software keeps a record of all the changes performed by all collaborators along with metadata about these changes. This makes possible to consult, for each *page*, its *history* of *revisions*.

MediaWiki is one of the most popular wiki software available. It was originally created for Wikipedia, but nowadays it is used for a wide variety of other wikis. In this section, we introduce the basic elements about MediaWiki and Wikipedia needed to properly understand this work.

In Figure 1.1 we can see the Wikipedia article of Edsger W. Dijkstra. Most Wikipedia articles look similar to this. They have a title (marked with **T**), a structured text with sections (**S**), links to other Wikipedia articles (**L**), information boxes (**IB**) and other elements.

---

Statistics.

<sup>3</sup>See [http://meta.wikimedia.org/wiki/Wikipedia\\_1.0](http://meta.wikimedia.org/wiki/Wikipedia_1.0).

<sup>4</sup><http://dbpedia.org/>

<sup>5</sup>Such as those provided by Google. See <http://www.google.com/help/features.html>.

<sup>6</sup>WYSIWYG – *What You See Is What You Get*.

Clicking on the `Edit` button on the top will lead to the *Edit page*, as shown in Figure 1.2. The main element of this interface is a text area with the source of the article. This is what should be edited to create and modify articles. Source is written in a mark-up language known as Wiki markup. Through diverse special characters and tags, it is possible to control style (*e.g.* bold font, **B**), links to other articles (**L**), templates to format elements such as information boxes (**IB**), section headings (**S**), etc<sup>7</sup>. Other relevant elements of the edit interface are a checkbox to indicate if the edit is a minor change (**M**) and the edit summary text box (**ES**). This edit summary, also known as revision comment, is a one-sentence summary of what is being changed, so that other users can interpret the article change history.

Clicking on the `View history` button will lead to the revision history, shown in Figure 1.3. This history contains a record of all edits made to the article. For each edit, its metadata is shown. This metadata includes a timestamp, username of the editor or his IP if he is anonymous, the edit summary, the size in bytes of the article source after the edit and a flag if the edit is a minor change.

We can visualize the changes performed in an edit by using MediaWiki's diff tool, clicking on the `prev` link at the left of a revision. The output of the diff tool for one of the edits on this article is shown in Figure 1.4. Parts of the revision that were changed are presented side-by-side before and after the edit. Using a code of colors to indicate what has been inserted, changed or deleted<sup>8</sup>.

Most tasks in Wikipedia, including editing and human vandalism detection are performed using this interface.

Finally, another relevant aspect of MediaWiki to this research are categories.<sup>9</sup> Pages are organized in a hierarchy of categories, where each page can have any number of categories, or none. We can see an example of a category list for an article in Figure 1.5. There is also organization in namespaces<sup>10</sup> for different kinds of pages (*e.g.* articles, talk pages, user pages). In this work, we use only the main namespace, or NS0, which contains the articles themselves, excluding any

---

<sup>7</sup>An exhaustive guide to wiki markup is available at [http://en.wikipedia.org/wiki/Help:Wiki\\_markup](http://en.wikipedia.org/wiki/Help:Wiki_markup).

<sup>8</sup>More information about MediaWiki's diff tool is available at <http://en.wikipedia.org/wiki/Help:Diff>.

<sup>9</sup>More information about categories in MediaWiki is available at <http://en.wikipedia.org/wiki/Help:Category>.

<sup>10</sup>More information about namespaces used in the English edition of Wikipedia is available at <http://en.wikipedia.org/wiki/Wikipedia:Namespace>.

Article [Discussion](#) Read [Edit](#) [View history](#)

## Edsger W. Dijkstra<sup>T</sup>

From Wikipedia, the free encyclopedia

**Edsger Wybe Dijkstra**<sup>B</sup> (May 11, 1930 – August 6, 2002); Dutch pronunciation: [ˈɛtsxər ˈwibe ˈdɛɪkstra]  (ⓘ) (ⓘ)  (ⓘ) (ⓘ)  (ⓘ) (ⓘ)) was a Dutch computer scientist. He received the 1972 [Turing Award](#) for fundamental contributions to developing programming languages, and was the Schlumberger Centennial Chair of Computer Sciences at The University of Texas at Austin from 1984 until 2000.

Shortly before his death in 2002, he received the [ACM](#)<sup>L</sup> PODC Influential Paper Award in distributed computing for his work on [self-stabilization](#) of program computation. This annual award was renamed the [Dijkstra Prize](#) the following year, in his honor.


**Contents** [\[hide\]](#)

- 1 Life and work
- 2 EWDs and writing by hand
- 3 Awards and honors
- 4 See also
- 5 Footnotes
- 6 References
  - 6.1 Writings by E.W. Dijkstra
  - 6.2 Others about Dijkstra, eulogies
- 7 External links

**Life and work**<sup>S</sup> [\[edit\]](#)

Born in [Rotterdam](#), Dijkstra studied [theoretical physics](#) at [Leiden University](#), but quickly realized he was more interested in computer science. Originally employed by the [Mathematisch Centrum](#) in Amsterdam, he held a professorship at the [Eindhoven University of Technology](#), worked as a research fellow for [Burroughs Corporation](#) in the early 1970s, and later held the Schlumberger Centennial Chair in

**Edsger Wybe Dijkstra**



<b>Born</b>	May 11, 1930 <a href="#">Rotterdam, Netherlands</a>
<b>Died</b>	August 6, 2002 (aged 72) <a href="#">Nuenen, Netherlands</a>
<b>Fields</b>	<a href="#">Computer science</a>
<b>Institutions</b>	<a href="#">Mathematisch Centrum</a> <a href="#">Eindhoven University of Technology</a>
	<a href="#">The University of Texas at Austin</a>
<b>Doctoral advisor</b>	<a href="#">Adriaan van Wijngaarden</a>
<b>Doctoral students</b>	<a href="#">Nico Habermann</a> <a href="#">Martin Rem</a> <a href="#">David Naumann</a> <a href="#">Cornelis Hemerik</a> <a href="#">Jan Tijmen Udding</a> <a href="#">Johannes van de Snepscheut</a> <a href="#">Antonetta van Gasteren</a>

Figure 1.1: Screenshot of the beginning of the Edsger W. Dijkstra article in Wikipedia.

special type of page.

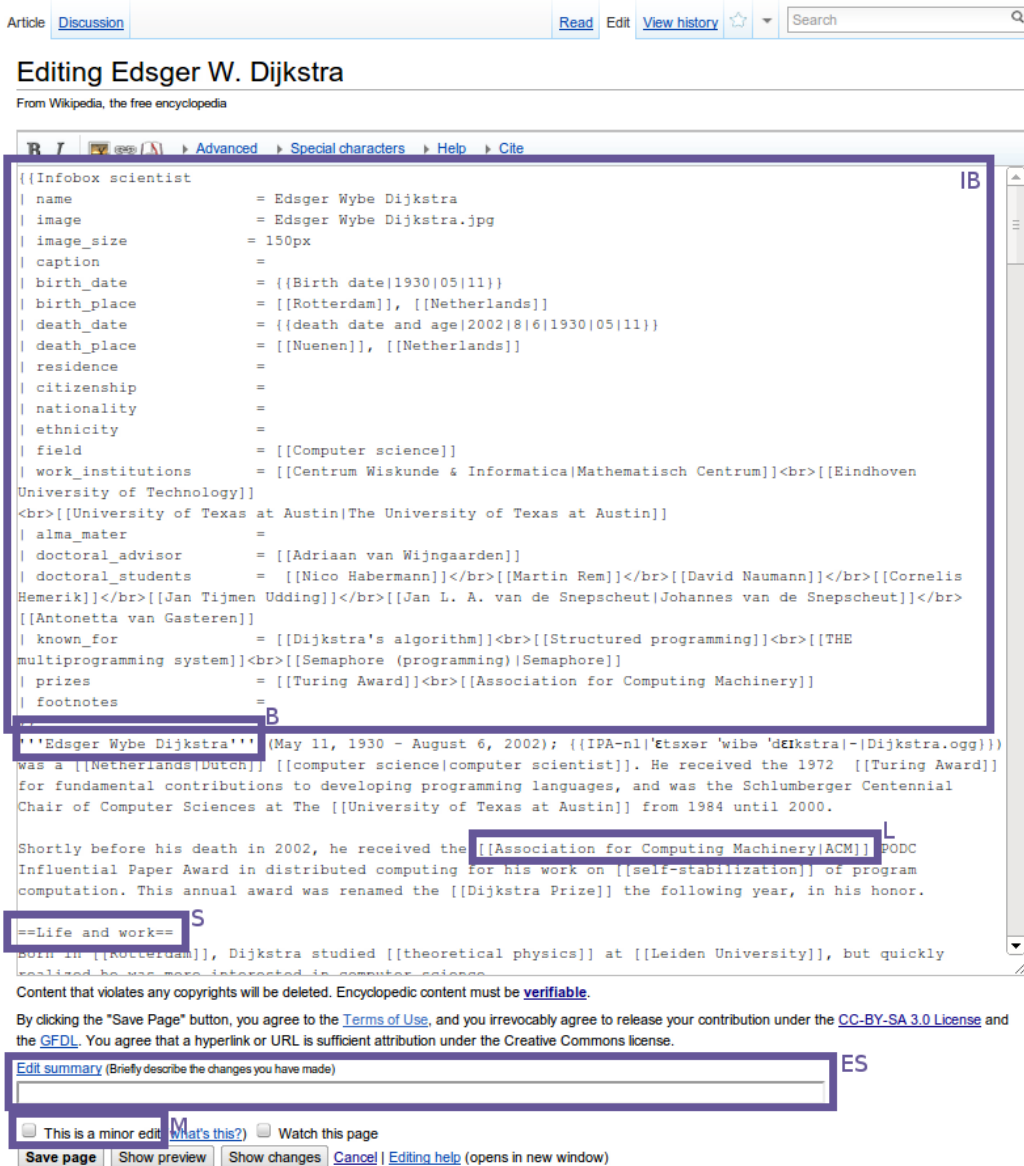


Figure 1.2: Screenshot of the edit page of the Edsger W. Dijkstra article in Wikipedia.

Article [Discussion](#) [Read](#) [Edit](#) [View history](#)

## Revision history of Edsger W. Dijkstra

From Wikipedia, the free encyclopedia  
[View logs for this page](#)

Browse history

From year (and earlier):  From month (and earlier):  Tag filter:  Deleted only

For any version listed below, click on its date to view it. For more help, see [Help:Page history](#) and [Help:Edit summary](#).  
 External tools: [Revision history statistics](#) · [Contributors](#) · [Revision history search](#) · [Number of watchers](#) · [Page view statistics](#)

(cur) = difference from current version, (prev) = difference from preceding version,  
 m = [minor edit](#), → = [section edit](#), ← = [automatic edit summary](#)  
 (latest | [earliest](#)) View (newer 50 | [older 50](#)) (20 | 50 | 100 | 250 | 500)

- [\(cur | prev\)](#)  [13:14, 9 July 2011](#) [Yobot](#) ([talk](#) | [contribs](#)) **m** (22,336 bytes) (*Updated infobox (BRFA 21) using [AWB](#) (7782)*) ([undo](#))
- [\(cur | prev\)](#)  [07:53, 9 July 2011](#) [Owertyus](#) ([talk](#) | [contribs](#)) (22,337 bytes) (*→Life and work: replace unsourced telescopes quote with sourced knife science quote (the telescope quote appears in fortune(6) since 4.3BSD, IIRC, but that's not an RS)*) ([undo](#))
- [\(cur | prev\)](#)  [08:23, 18 June 2011](#) [SchreyP](#) ([talk](#) | [contribs](#)) (21,670 bytes) (*Undid revision 434895779 by [85.185.67.235](#) ([talk](#)) revert not motivation deletion*) ([undo](#))
- [\(cur | prev\)](#)  [07:23, 18 June 2011](#) [85.185.67.235](#) ([talk](#)) (18,001 bytes) (*→EWDs and writing by hand*) ([undo](#)) (*Tag: section blanking*)
- [\(cur | prev\)](#)  [12:07, 9 June 2011](#) [Nbarth](#) ([talk](#) | [contribs](#)) (21,670 bytes) (*→Life and work: harmful clarification*) ([undo](#))
- [\(cur | prev\)](#)  [12:06, 9 June 2011](#) [Nbarth](#) ([talk](#) | [contribs](#)) (21,653 bytes) (*→Life and work: harmful*) ([undo](#))
- [\(cur | prev\)](#)  [12:57, 25 May 2011](#) [Rami R](#) ([talk](#) | [contribs](#)) **m** (21,508 bytes) (*Reverted edits by [Dessertsheep](#) ([talk](#)) to last version by [Figure19](#)*) ([undo](#))
- [\(cur | prev\)](#)  [11:59, 25 May 2011](#) [Dessertsheep](#) ([talk](#) | [contribs](#)) (21,727 bytes) (*→References*) ([undo](#))
- [\(cur | prev\)](#)  [11:58, 25 May 2011](#) [Dessertsheep](#) ([talk](#) | [contribs](#)) (21,635 bytes) (*→See also*) ([undo](#))
- [\(cur | prev\)](#)  [11:57, 25 May 2011](#) [Dessertsheep](#) ([talk](#) | [contribs](#)) (21,553 bytes) (*→Life and work*) ([undo](#))
- [\(cur | prev\)](#)  [05:13, 9 May 2011](#) [Figure19](#) ([talk](#) | [contribs](#)) (21,508 bytes) (*added [Category:Cancer deaths in the Netherlands](#)*)

Figure 1.3: Screenshot of the revision history of the Edsger W. Dijkstra article in Wikipedia.



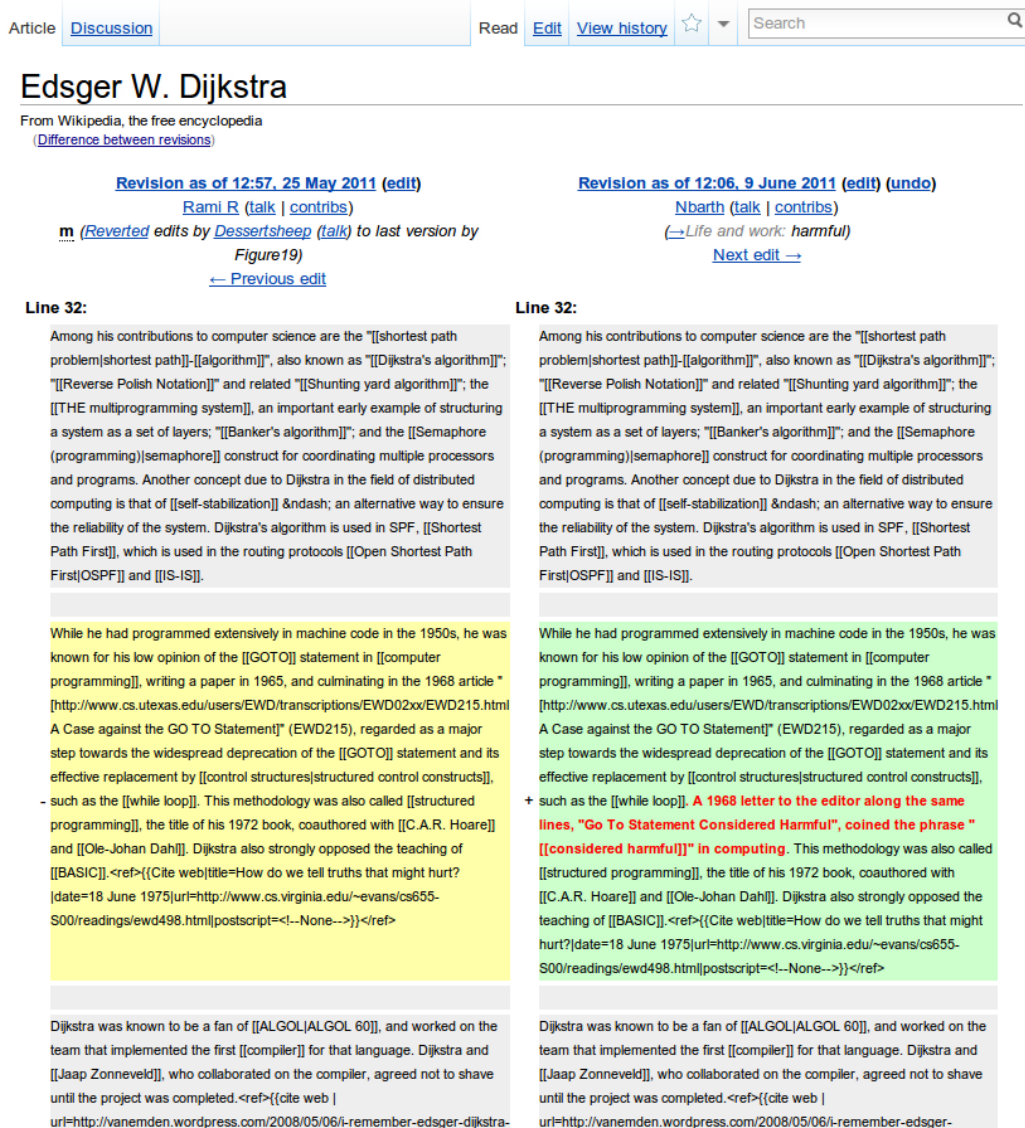


Figure 1.4: Screenshot of a diff page corresponding to an edit on the Edsger W. Dijkstra article in Wikipedia.

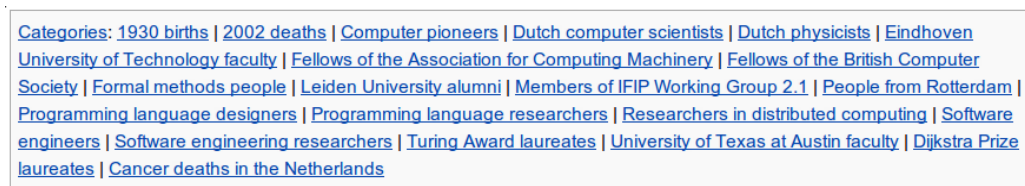


Figure 1.5: Screenshot of the categories at the bottom of the Edsger W. Dijkstra article in Wikipedia.

### 1.3 What is Vandalism and Why Does It Matter

The fact that anyone can edit Wikipedia at any time with very little practical restrictions is at the core of its success and, at the same time, it is one of its main sources of trouble. By guaranteeing any person freedom to edit its contents, Wikipedia has become a target for pranksters and, with its increasing popularity, for spammers, lobbyists and other people interested in self-promotion, manipulation and propaganda. This has a wide-ranging negative impact in Wikipedia itself and all applications that use Wikipedia as a knowledge source.

In this research, we focus in the phenomenon of *vandalism*, which is defined by Wikipedia itself as follows (Wikipedia contributors 2010):

*Vandalism is any addition, removal, or change of content made in a deliberate attempt to compromise the integrity of Wikipedia. [...] Common types of vandalism are the addition of obscenities or crude humor, page blanking, and the insertion of nonsense into articles.*

*Any good-faith effort to improve the encyclopedia, even if misguided or ill-considered, is not vandalism. Even harmful edits that are not explicitly made in bad faith are not vandalism.*

As an example of vandalism, we can take the article shown in Section 1.2. In its very recent history, as shown in Figure 1.3, it has suffered three acts of vandalism by a user called *Dessertsheep*. If we use the diff tool, we see that he inserted a vulgar sentence, as shown in Figure 1.6. After this edit, part of the article appeared as shown in Figure 1.7.

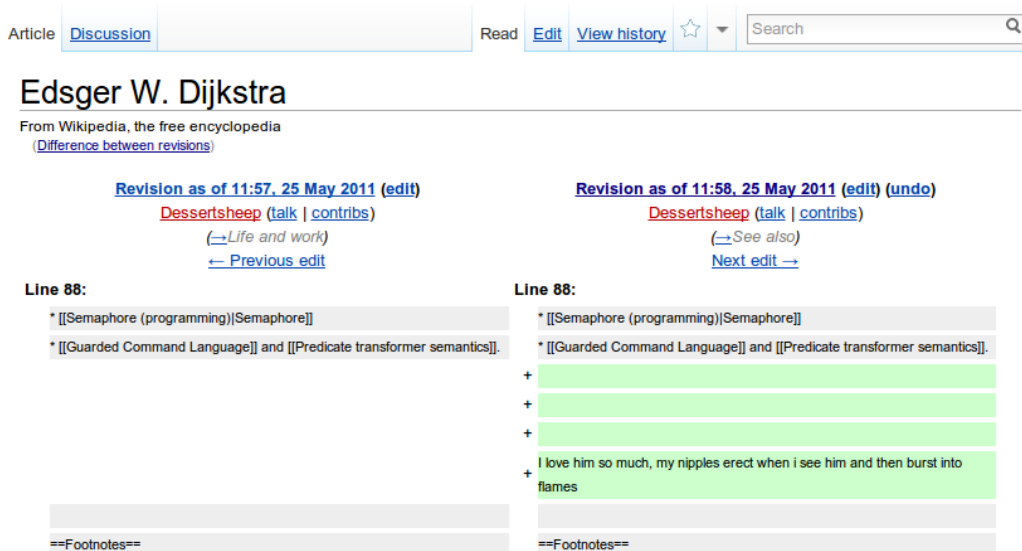


Figure 1.6: Screenshot of a diff for a vandalism edit on the Edsger W. Dijkstra article in Wikipedia.

## See also

- [Dijkstra's algorithm](#)
- [Smoothsort](#)
- [Dining philosophers problem](#)
- ["The Cruelty of Really Teaching Computer Science"](#)
- [Semaphore](#)
- [Guarded Command Language](#) and [Predicate transformer semantics](#).

I love him so much, my **nipples** erect when i see him and then burst into flames

Figure 1.7: Screenshot of part of the Edsger W. Dijkstra article in Wikipedia, after vandalism.

## 1.4 Organization of This Thesis

The rest of this thesis is structured as follows: Chapter 2 explains the phenomenon of vandalism. Chapter 3 defines the Wikipedia vandalism detection tasks and discusses available methods and datasets. Chapter 4 presents our research, consisting of Section 4.1 with the system we developed for the PAN 2010 competition and Section 4.2 with the joint effort made with other authors to combine and evalu-

ate some of the top vandalism detection systems. Chapter 5 concludes with our contributions and a discussion of future work.

# Chapter 2

## Wikipedia Vandalism

### 2.1 Kinds of Vandalism

Vandalism is a highly subjective and wide concept. There have been attempts to give a concise definition by creating taxonomies of vandalism. There are many kinds of vandalism (Chin et al. 2010; Priedhorsky et al. 2007; Viégas, Wattenberg, and Dave 2004), Wikipedia contributors identify 20 categories (Wikipedia contributors 2010), shown in Table 2.1. Still, the only reference is the ever evolving consensus in each Wikipedia community<sup>1</sup> on what is vandalism and what is not, so there is no gold standard.

Tightly attached to the concept of vandalism are good and bad faith, which are terms regularly used in the Wikipedia community. However, from a computational point of view, we are actually studying vandalism as *damage* to the encyclopedia, regardless of intentions and leaving judgmental issues to human experts.

---

<sup>1</sup>Wikipedia editions in different languages have independent communities.

Table 2.1: Summary of vandalism types by Wikipedia contributors (Wikipedia contributors 2010). Some categories outside the scope of this work have been left out of the table (malicious account creation, abuse of tags, avoidant vandalism, repeated upload of copyrighted material, gaming the system, talk page vandalism, user and user talk page vandalism, and vandabots). Table originally composed for (Mola-Velasco 2011).

<b>Type</b>	<b>Description</b>
Blanking	Removing all or significant parts of a page's content without any reason.
Edit summary vandalism	Making offensive edit summaries in an attempt to leave a mark that cannot be easily expunged from the record.
Hidden vandalism	Any form of vandalism not visible in the final article but visible during editing.
Image vandalism	Uploading shock images, inappropriately placing explicit images on pages, or simply using any image in a way that is disruptive.
Link vandalism	Adding or changing internal or external links on a page to disruptive, irrelevant, or inappropriate targets while disguising them with mislabeling.
Illegitimate page creation	Creating new pages with the sole intent of malicious behaviour.
Page lengthening	Adding very large amounts of content to a page so as to make the page's load time abnormally long.
Page-move vandalism	Changing the names of pages to disruptive, irrelevant or inappropriate names.
Silly vandalism	Adding profanity, graffiti or patent nonsense to pages.
Sneaky vandalism	Vandalism that is harder to spot, or that otherwise circumvents detection, including adding plausible misinformation and hiding vandalism through multiple edits.
Spam external linking	Adding links to irrelevant sites after having been warned.
Template vandalism	Modifying the wiki language or text of a template in a harmful or disruptive manner.

## 2.2 Vandalism Statistics and Impact

### 2.2.1 Vandalism Statistics

The Wikipedia community conducts its own quantitative and qualitative studies on vandalism. *Study 1* consisted of manually checking 100 random articles with a total of 668. Observed vandalism constituted a 4.6%. The observed time period comprised 2004, 2005 and 2006 and vandalism percentage appeared to be stable, oscillating between 3% and 6% of total edits. The most common vandalism type was obvious vandalism (83.87%) followed by deletion vandalism (9.68%) (Wikipedia contributors 2011).

Currently, the most accurate estimation of vandalism in the English edition of Wikipedia is around 7% of all edits (Potthast 2010). If we consider that there were 10 million edits between August 20 and October 10 2010, which makes almost 200 thousand edits per day on average<sup>2</sup>, we can assume the order of magnitude of vandalism edits per day is  $10^4$ .

According to Wikipedia's *Study 1*, 96.77% of all vandalism edits were performed by unregistered users. In 74.19% of cases, vandalism was reverted by a registered user.

Another important statistic is how much time vandalism remains in Wikipedia and how many people view it. Viégas, Wattenberg, and Dave (2004) estimated that mass deletions remain 7.7 days on average with a median time of 2.8 minutes, while mass deletions involving obscenities remain 1.8 days on average with a median time of 1.7 minutes. Priedhorsky et al. (2007) further studied the problem with results consistent with those by Viégas, Wattenberg, and Dave (2004), and estimated that the probability that a view of Wikipedia between 2003 and 2006 included damaged content was of 0.0037. This probably can be translated to 188 million views of vandalism during the studied period.

### 2.2.2 Vandalism Impact: An Anecdote

To further illustrate the impact of vandalism, we expose an anecdote<sup>3</sup>. On March 14th 2010, a prankster edited the Wikipedia article of a small belgian town:

---

<sup>2</sup>See more statistics at <http://en.wikipedia.org/wiki/User:Katalaveno/TBE> and <http://en.wikipedia.org/wiki/Wikipedia:Statistics>

<sup>3</sup>Thanks to Damiano Spina Valenti, who made us aware of this anecdote.



Figure 2.1: *Blastoise attacks in Kaster, Belgium*, image depicting the incident in Kaster’s Wikipedia article.

Kaster.<sup>4</sup> After this edit, Kaster’s article introduction was the following:

**Kaster** is a village in Belgium, part of the municipality of Anzegem. Recently, the town made headlines when a serial rapist dressed as a Blastoise Pokemon raped and killed 65 men.

Most of the time, this would have been corrected quickly and nobody would have noticed. However, the edit was not reverted until April 9th 2010<sup>5</sup>. During that month, Google’s spider fetched the article and indexed it its database and as a result, typing *define:kaster* in Google’s search engine would show up the above prank as the definition for Kaster.

What started as a small prank by two brothers<sup>6</sup> ended up being an embarrassing thing for Wikipedia and Google. Leaving aside the fun that this provided to many people<sup>7</sup>, Wikipedia needs to put measures in place to fight vandalism and prevent damages on the credibility of the project.

<sup>4</sup>The vandalism edit can be seen in Wikipedia’s history at <http://en.wikipedia.org/w/index.php?title=Kaster&diff=prev&oldid=349895615>.

<sup>5</sup>The revert edit is available at <http://en.wikipedia.org/w/index.php?title=Kaster&diff=next&oldid=349895615>.

<sup>6</sup>The authors of the prank later admitted it and discussed with Reddit readers. Discussion available at [http://www.reddit.com/r/IAmA/comments/dqvqk/iam\\_the\\_cocreator\\_of\\_the\\_kaster\\_blastoise/](http://www.reddit.com/r/IAmA/comments/dqvqk/iam_the_cocreator_of_the_kaster_blastoise/).

<sup>7</sup>Some people consider this as a funny Internet meme, as documented in Know Your Meme database. Available at <http://knowyourmeme.com/memes/blastoise-attacks-in-kaster-belgium>.



# Chapter 3

## Wikipedia Vandalism Detection

### 3.1 Practical Tools Against Vandalism

#### 3.1.1 Anti-vandalism Patrolling

The main force against vandalism is people who manually checks latest changes made to Wikipedia and review them to find vandalism and revert it. This activity is known as *patrolling*.<sup>1</sup>

The classic method of patrolling is opening a browser tab with the list of recent changes and skim through the list, then open in other tabs suspicious edits, check them and revert them if necessary.

#### 3.1.2 Patrolling Assistance

A wide variety of tools have been developed to assist patrollers in their work.<sup>2</sup> These range from tools aimed at browsing and editing Wikipedia in a faster and more convenient way, such as Twinkle<sup>3</sup> or Huggle<sup>4</sup>, to automatic detection systems that work with human supervision, such as STiki<sup>5</sup>.

---

<sup>1</sup>Patrolling might refer also to activities such as fixing typos, and not only vandalism. More information available at <http://en.wikipedia.org/wiki/Wikipedia:Patrols>.

<sup>2</sup>A list of tools is available at [http://en.wikipedia.org/wiki/Category:Wikipedia\\_counter-vandalism\\_tools](http://en.wikipedia.org/wiki/Category:Wikipedia_counter-vandalism_tools).

<sup>3</sup>Available at <http://en.wikipedia.org/wiki/Wikipedia:Twinkle>.

<sup>4</sup>Available at <http://en.wikipedia.org/wiki/Wikipedia:Huggle>.

<sup>5</sup>Available at <http://en.wikipedia.org/wiki/Wikipedia:STiki>.

### 3.1.3 Automatic Systems, Bots and Edit Filters

Automatic detection systems are designed to work with very limited human intervention or no intervention at all. In practice, there are two ways of implement them: as bots or edit filters.

On one hand, bots operate autonomously as agents external to Wikipedia, and as such, they detect and revert vandalism some time after it is performed. We will go into deeper detail about bots in Section 3.5.1.

On the other hand, edit filters<sup>6</sup> are a recent addition to the MediaWiki, deployed since 2009. They look for common patterns of vandalism at the edit time. If the edit matches one of these patterns, MediaWiki will reject it. The advantage of this approach is that when a vandalism edit is detected, it is rejected before it takes effect.

## 3.2 Problem Definition and Notation

We will define the Wikipedia vandalism detection problem and all associated elements in an attempt to unify previous notations and provide consistence for this work<sup>7</sup>.

A revision  $r$  is the state of an article in a given point of its history. We use  $r^-$  and  $r^+$  to denote a past or future revision with respect to  $r$ , respectively. We use subindices  $r_i$ ,  $r_{i-1}$  or  $r_{i+1}$  to denote specific past or future revisions.

An edit  $e$  is the transition between two consecutive revisions. The Wikipedia vandalism detection task consists in decide whether a given edit  $e$  is vandalism or not. From the point of view of machine learning, given the set of  $E$  of all edits, we use:

- A corpus  $E_c \subset E$  of labeled edits.
- An edit model  $\alpha : E \rightarrow \mathbf{E}$  that maps each edit  $e$  onto a feature set  $\mathbf{e}$  quantifying characteristics of  $e$  that are useful for discriminating between vandalism and non-vandalism edits.

<sup>6</sup>More information about edit filters is available at [http://en.wikipedia.org/wiki/Wikipedia:Edit\\_filter](http://en.wikipedia.org/wiki/Wikipedia:Edit_filter).

<sup>7</sup>Notation was adopted as introduced for the 1st International Competition on Wikipedia Vandalism Detection (Potthast, Stein, and Holfeld 2010, p. 1). Notation for revisions is based on that presented by Adler and Alfaro (2007, p. 4) and Adler, Alfaro, and Pye (2010, pp. 4-5).

- A classifier  $c : E \rightarrow [0, 1]$ . The result of this classifier is the confidence of a given edit  $e$  being vandalism.
- A threshold  $\tau$  is defined so that any  $c(\mathbf{e}) \geq \tau$  indicates vandalism, and  $c(\mathbf{e}) \leq \tau$  indicates otherwise.
- For any unseen edit  $e \in E \setminus E_c$ , we check whether it is vandalism or not by computing  $c(\alpha(e)) > \tau$ .

### 3.2.1 Immediate and Historic Detection

Vandalism detection includes two different tasks: *immediate*<sup>8</sup> and *historic* detection.

Immediate detection is the most extended: detecting vandalism right after it happens. The historic variant is detecting vandalism at any point in the past. The technical difference between them is that, in the case of historic detection, information about everything that happened after the vandalism act is available to the system. Immediate detection is the most applied and useful to maintain Wikipedia clean of vandalism. The interest in historic detection is that much higher performance can be achieved, making it useful for building corpora to train immediate detection systems and also creating clean snapshots of Wikipedia, by selecting revisions of each articles that are guaranteed to be vandalism-free.

## 3.3 Performance Measures

Wikipedia vandalism detection is a binary or one-class classification problem. We define those performance measures used in this thesis<sup>9</sup>.

The elemental measures are given by the confusion matrix, as shown in Table 3.1. These are true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). These are used to calculate the following performance measures: *Precision* (P) is the fraction of samples classified as positives that are actually positives, as defined in Equation 3.1; *Recall* (R) or True Positive Rate

<sup>8</sup>Also called *zero-delay* in (Adler, Alfaro, and Pye 2010).

<sup>9</sup>F-Measure curves were calculated with a custom script by the author. The rest of performance measures and graphs were calculated using AUCCalculator 0.2 by Davis and Goadrich (2006). Available at <http://mark.goadrich.com/programs/AUC/>.

(TPR) is the fraction of positive samples correctly classified, as defined in Equation 3.3; False Positive Rate (FPR) is the fraction of negative samples misclassified as positive, as defined in Equation ??.

F-Measure<sup>10</sup> is sometimes used as a measure to compare classifiers, defined as the harmonic mean of precision and recall, see Equation 3.4.

Table 3.1: Confusion matrix example.

		Actual classification	
		Positive	Negative
Predicted classification	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	False positive (FP)

$$Precision = \frac{TP}{TP + FP} \quad (3.1)$$

$$Recall = TPR = \frac{TP}{TP + FN} \quad (3.2)$$

$$FPR = \frac{FP}{FP + TN} \quad (3.3)$$

$$F\text{-Measure} = 2 \cdot \frac{P \cdot R}{P + R} \quad (3.4)$$

If we plot Precision against Recall, for every confidence threshold, we obtain the Precision-Recall curve. We use this curve to grasp the performance of different classifiers in a comprehensive and intuitive way. Given this plot, we calculate the Area Under Precision-Recall Curve (AUC-PR) which we use as the main evaluation criterion for vandalism systems.

If we plot the True Positive Rate against False Positive Rate, we obtain the Receiver Operating Characteristic (ROC) curve. We use area under this curve (AUC-ROC) as a secondary evaluation criterion.

AUC-ROC is often used for binary classification problems, but AUC-PR better accounts for the fact that vandalism is a rare phenomenon (Davis and Goadrich 2006), and offers a more discriminating look into the performance of the various feature combinations. We will present results using both measures in our experimentation.

<sup>10</sup>Also known as F<sub>1</sub>-Score and F-Score.

## 3.4 Corpora

For the best of our knowledge, there are six Wikipedia vandalism corpora. All our work used the PAN-WVC-10 corpus, although we will present all the six corpus for reference.

### 3.4.1 Webis-WVC-07

The Webis Wikipedia vandalism corpus<sup>11</sup>, or Webis-WVC-07, is the first public Wikipedia vandalism corpus reported in the literature. It consists of 940 edits annotated by humans, 301 of them annotated as vandalism. (Potthast and Gerling 2007; Potthast, Stein, and Gerling 2008)

### 3.4.2 Chin 2010

This corpus was built and used for (Chin et al. 2010). It was built based on the Wikipedia revision history up to February 24th, 2009 and it consists of the full history of two of the most vandalized pages<sup>12</sup>: Abraham Lincoln (8,816 revisions) and Microsoft (8,220 revisions).

Annotation was performed in an active learning fashion. A first classification model was built using the Webis-WVC-07 corpus, and that model was used to get a rank of the top 50 candidates to be vandalism. An annotator revised these candidates and annotated them. The annotated edits were added to the training corpus and the process was repeated iteratively.

This annotation method makes (Chin et al. 2010) an interesting approach to solve the problem of annotating a corpus big enough to be used for supervised classification.

### 3.4.3 West 2010

West, Kannan, and Lee 2010 use a unique approach to annotate their corpus<sup>13</sup>. In Wikipedia, some privileged users have the right to revert an edit using a single-click feature called *rollback*, used to undo blatantly unproductive edits. The au-

---

<sup>11</sup>Available at <https://www.uni-weimar.de/cms/medien/webis/research/corpora/webis-wvc-07.html>.

<sup>12</sup>More information about the most vandalized pages available at [http://en.wikipedia.org/wiki/Wikipedia:Most\\_vandalized\\_pages](http://en.wikipedia.org/wiki/Wikipedia:Most_vandalized_pages).

<sup>13</sup>Available at <http://www.cis.upenn.edu/~westand/>.

thors define an *offending edit* as one that was reverted using the rollback function. Although this is only a small portion of vandalism edits, this approach results in a very high confidence for positive annotations. The corpus contains 5,713,762 edits labeled as *blatantly unproductive* using the described automatic method; it also contains 5,291 vandalism edits that were manually annotated. This makes West 2010 the largest Wikipedia vandalism corpus reported until now.

### 3.4.4 PAN-WVC-10

The PAN Wikipedia Vandalism Corpus 2010<sup>14</sup>, or PAN-WVC-10, is the successor of Webis-WVC-07 (Potthast 2010). It consists of 32,439 edits, of which 2,394 are annotated as vandalism<sup>15</sup>. Amazon Mechanical Turk<sup>16</sup> was used to distribute the task amongst hundreds of human annotators. Each edit was annotated by 3 people. If they did not agree, the edit was annotated by 3 more people. This process was repeated until every edit had an annotation with more than 2/3 of inter-annotator agreement. After 8 iterations, there were 70 tie edits that were reviewed by the corpus authors.

Due to its use in the 1st International Competition on Wikipedia Vandalism Detection<sup>17</sup> (Potthast, Stein, and Holfeld 2010) it is one of the most widely used corpus in the scientific literature.

In this thesis, we use a modified version of PAN-WVC-10 where 157 edits were removed<sup>18</sup>. This was because of these edits were deleted from the Wikipedia history<sup>19</sup> at the time of writing (Adler et al. 2011). Statistics for our corpus version

<sup>14</sup>Available at <https://www.uni-weimar.de/cms/medien/webis/research/corpora/pan-wvc-10.html>.

<sup>15</sup>(Potthast 2010) reports 32,452 edits and 2,391 annotated as vandalism. However, we use the statistics of the corpus as fetched on July 19th 2011, with MD5 checksum `fed384796c5cdb066d2ab9d1c0ec7764`. The differences are due to, mainly, error correction.

<sup>16</sup><http://www.mturk.com/>

<sup>17</sup>This competition is part of the International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse. More information available at <http://pan.webis.de/>.

<sup>18</sup>For the shake of reproducibility: The MD5 checksum of the PAN-WVC-10 corpus at the time we used it was `b6729c1700da7b26f280966a24ad1110`. The `gold-annotations.csv` file of that version is available at <http://bitsnbrains.net/resources/wikipedia-vandalism/pan-wvc-10-b6729c1700da7b26f280966a24ad1110-gold-annotations.csv> and a list of the IDs of the 157 edits we removed is available at [http://bitsnbrains.net/resources/wikipedia-vandalism/adler11\\_missing\\_ids.dat](http://bitsnbrains.net/resources/wikipedia-vandalism/adler11_missing_ids.dat).

<sup>19</sup>Edits are usually removed from Wikipedia history when the article they pertain to is removed. These edits are preserved by Wikipedia, but they can only be seen and restored by administrators.

are: 32,282 total edits, with 2,395 vandalism edits.

### 3.4.5 PAN-WVC-11

The PAN Wikipedia Vandalism Corpus 2011<sup>20</sup>, or PAN-WVC-11, is a supplement to PAN-WVC-10. It is the first multilingual corpus, including sections for English, German and Spanish.

The English section consists of new 9985 annotated edits of the same time period as those compiled for PAN-WVC-10. 1144 of them are annotated as vandalism. The German section consists of 9990 edits, 589 of them annotated as vandalism. The Spanish section consists of 9974 edits, 1081 of them annotated as vandalism.

### 3.4.6 ClueBot-NG dataset

ClueBot-NG<sup>21</sup> dataset is an ever evolving one. Through its online review interface<sup>22</sup> a multitude of Wikipedia users annotate edits as *vandalism*, *constructive* or *skipped*. The final classification is decided as follows:

- A minimum of 2 annotators agreeing is required for the edit to be considered as annotated.
- If more than a half of annotators skipped the edit, it is annotated as *skipped*.
- If, at least, *constructive* annotations are thrice the *vandalism* annotations, the edit is annotated as *constructive*.
- If, at least, *vandalism* annotations are thrice the *constructive* annotations, the edit is annotated as *vandalism*.
- If none of the previous criteria is met, the edit is not considered as annotated and therefore it is not added to the final dataset.

We fetched this dataset on July 14th of 2011 and contained:

- 3284 edits.

---

<sup>20</sup>Available at <https://www.uni-weimar.de/cms/medien/webis/research/corpora/pan-wvc-11.html>.

<sup>21</sup>More information about ClueBot-NG is available at [http://en.wikipedia.org/wiki/User:ClueBot\\_NG](http://en.wikipedia.org/wiki/User:ClueBot_NG).

<sup>22</sup>Available at <http://cluebotreview.g.cluenet.org/>.

- 2874 edits annotated as *vandalism* or *constructive*, of which:
  - 2308 (80.31%) were *constructive*.
  - 566 (19.69%) were *vandalism*.
  
- 566 vandalism

The strong point of this corpus is that it is annotated by experts. Therefore, we can expect a high quality in annotation and compliance with Wikipedia standards. This is an advantage over PAN-WVC-10, whose annotations might be less reliable; and over West 2010, which has a high amount of false negatives. However, its size is one order of magnitude below PAN-WVC-10 and four below West 2010.

### 3.4.7 Wikipedia dumps

Wikimedia offers XML and SQL dumps of the entire database for all its projects<sup>23</sup>. These dumps include the full revision history of every article, along with other information. This is a resource commonly used to build Wikipedia vandalism corpora and detection systems.

### 3.4.8 Wikipedia User Contribution Dataset

Javanmardi, Lopes, and Baldi (2010) created a dataset<sup>24</sup> of content insertions and deletions per user. This dataset comprehends all Wikipedia insertions and deletions since its creation to January 30th, 2010. This is a valuable approach for user reputation methods.

---

<sup>23</sup>Dumps are available at <http://dumps.wikimedia.org/>. More information about what is available and its format is available at [http://meta.wikimedia.org/wiki/Data\\_dumps](http://meta.wikimedia.org/wiki/Data_dumps).

<sup>24</sup>This dataset and an online API to query it are available at <http://nile.ics.uci.edu/events-dataset-api/>.



## 3.5 Related Work

### 3.5.1 First Generation

One of the first antivandalism tools was Vandal Fighter<sup>25</sup>, released in 2005. Vandal Fighter is a tool assisting patrollers when spotting and reverting vandalism. Among other features, it includes a list of regular expressions that are used to find vandalism.

The first generation of automatic Wikipedia vandalism detection systems emerged from the Wikipedia community itself and have been actively developed and used since 2006. These systems are bots<sup>26</sup> that review latest changes made to Wikipedia, check them and revert them if vandalism is found. As such, all of them are systems in production.

An on-going effort to create a bot census by Posada and Wikipedia contributors (2011) accounts for most anti-vandalism bots, both first and second generation and for different languages<sup>27</sup>.

First generation bots use the following methods:

1. Regular expressions to detect offensive terms or patterns often used by vandals.
2. A set of heuristics combing regular expressions, amount of deleted, inserted or changed text, editor statistics and status and revision metadata.
3. A scoring system that takes into account regular expressions and heuristics to compute a score for each edit. An edit is considered vandalism when its score is greater than a manually-adjusted threshold.

Prominent first generation bots include: ClueBot in the English Wikipedia (Carter 2010), AVBOT in the Spanish Wikipedia (Posada 2010), Salebot in the

---

<sup>25</sup>Its historical page is available at <http://en.wikipedia.org/wiki/User:CryptoDerk/CDVF>.

<sup>26</sup>A *bot* is any system that works autonomously performing a task in an environment where also humans work. In Wikipedia, a bot is any software that edits Wikipedia in a unsupervised or semi-supervised fashion. More information about bots in Wikipedia available at <http://en.wikipedia.org/wiki/Wikipedia:Bots>.

<sup>27</sup>We used data from the census version of February 11th 2011, available at [http://en.wikipedia.org/w/index.php?title=User:Emijrp/Anti-vandalism\\_bot\\_census&oldid=413291924](http://en.wikipedia.org/w/index.php?title=User:Emijrp/Anti-vandalism_bot_census&oldid=413291924).

French and Portuguese Wikipedias and AntiVandalBot in the Simple English Wikipedia<sup>28</sup>.

## 3.6 Second Generation

The second generation of vandalism detection systems apply machine learning to go beyond the performance previously achieved by regular expressions and scoring systems.

### 3.6.1 Textual and Simple Metadata-based Features

Druck, Miklau, and McCallum (2008) advanced some of the basic content and metadata-based features applied to quality evaluation of Wikipedia edits. Potthast, Stein, and Gerling (2008) then published the first vandalism system that we can consider in the second generation. They created an edit representation with 16 features, detailed in Table 3.2 and evaluated the performance using a logistic regression with their Webis-WVC-07 corpus<sup>29</sup>.

Other features in this category have been explored in the literature. A more exhaustive compilation can be found in (Potthast, Stein, and Holfeld 2010).

---

<sup>28</sup>Previously also used in the English Wikipedia, but now superseded by ClueBot-NG.

<sup>29</sup>See Section 3.4.1.

Table 3.2: Features used by Potthast, Stein, and Gerling (2008). Table extracted from (Potthast, Stein, and Gerling 2008, p. 3).

<b>Feature</b>	<b>Description</b>
char distribution	deviation of the edit's character distribution from the expectation
char sequence	longest consecutive sequence of the same character in an edit
compressibility	compression rate of an edit's text
upper case ratio	ratio of upper case letters to all letters of an edit's text
term frequency	average relative frequency of an edit's words in the new revision
longest word	length of the longest word
pronoun frequency	number of pronouns relative to the number of an edit's words (only first-person and second-person pronouns are considered)
pronoun impact	percentage by which an edit's pronouns increase the number of pronouns in the new revision
vulgarism frequency	number of vulgar words relative to the number of an edit's words
vulgarism impact	percentage by which an edit's vulgar words increase the number of vulgar words in the new revision
size ratio	the size of the new version compared to the size of the old one
replacement similarity	similarity of deleted text to the text inserted in exchange
context relation	similarity of the new version to Wikipedia articles found for keywords extracted from the inserted text
anonymity	whether an edit was submitted anonymously, or not
comment length	the character length of the comment supplied with an edit
edits per user	number of previously submitted edits from the same editor or IP

### 3.6.2 Compression Models

Compression models are widely used in spam detection. Smets, Goethals, and Verdonk (2008) originally applied compression models to the vandalism detection task, using the Probabilistic Sequence Modeling method by Bratko et al. (2006). Itakura and Clarke (2009) further refined this approach by using Dynamic Markov Compression by treating edits where text is inserted or changed as two separate problems. Then, the compression ratio is tested against previous vandalism ed-

its and previous non-vandalism edits. An edit is considered vandalism if it has a higher compression ratio against vandalism edits than against non-vandalism edits.

### 3.6.3 Topic Modeling

Vandalism often presents vocabulary that is not common in an encyclopedic article of a given topic. This has been presented as a problem in the literature, since it is hard to spot vandalism when topic-specific knowledge is required. Wang and McKeown (2010) approaches this problem by retrieving web pages using the title of the article as a query in a search engine. These web pages are then used to build language models and compute the likelihood and perplexity against each edit. A vandalism edit is expected to have low likelihood and high perplexity for these topic-specific language models.

### 3.6.4 Article History Modeling

Chin et al. (2010) work on the assumption that most vandalism produces content that is alien to the article where it is performed. Based on that assumption, for each revision, they create bigram models of some past revisions. Then, they use these language models of past revisions to compute perplexity, out-of-vocabulary words and related metrics of the current revision, and use them as features for a supervised classification algorithm.

This might be seen as a simple model based on similar assumptions to those made in content-based reputation systems.

### 3.6.5 Content-based Reputation

Adler, Alfaro, and Pye (2010) introduce content-based reputation. On their previous work, they created WIKITRUST<sup>30</sup>, an online tool to measure reputation of Wikipedia's authors and contents (adler08; Adler and Alfaro 2007). The authors created a vandalism detection system built on top of WikiTrust. Also, they introduced the immediate and historic vandalism division. The most important features unique to this approach are:

---

<sup>30</sup>Available at <http://www.wikitrust.net/>.

**Author reputation** <sup>31</sup> It is 0 for anonymous or novice users. It improves with good edits.

**Minimum revision quality** (*Historic only*). WikiTrust computes quality for each revision with respect 6 future and 6 past revisions. The quality  $q$  with respect a past revision  $r^-$  and a future revision  $r^+$  is computed as  $q(r|r^-, r^+) = \frac{d(r^-, r^+) - d(r, r^+)}{d(r^-, r)}$ , where  $d(r, r')$  is a distance metric. This feature represents the minimum  $q(r|r^-, r^+)$  for a given revision. According to the authors, this was the most influential feature in their system.

**Average revision quality** (*Historic only*). It is the average quality with respect 6 future and 6 past revisions.

**Maximum dissent** (*Historic only*). It measures how close the average revision quality is to the minimum revision quality.

**Previous text histogram** . For each revision, WikiTrust computes the trust of each word based on how much has been revised by reputable authors, that is, when a reputable author introduces text, this text has high trust, when he deletes text, its trust decreases. The text of the previous revision is divided in 10 sections and a histogram of trust is calculated, producing 10 features.

**Current text histogram** . Analogous to the previous features, but for the text of the current revision.

**Histogram difference** . It measures the difference between the previous and current text histograms.

Similar reputation models have been developed by other authors both for vandalism detection and quality assessment **javanmardi11**; Javanmardi, Lopes, and Baldi (2010); Wöhner and Peters (2009).

### 3.6.6 Spatio-temporal Analysis of Metadata

As presented in previous sections, some metadata-based features are widely used. However, West, Kannan, and Lee (2010) have been the first to exploit metadata in depth and as the main component for vandalism detection. They do so through

<sup>31</sup>This feature, as used in this thesis is only for historic vandalism detection. However, it is possible to use it in immediate detection through the WikiTrust vandalism detection tool.

what they call spatio-temporal analysis of metadata<sup>32</sup>. Given the important role that plays this approach inside this thesis, we explain it in depth.

West, Kannan, and Lee (2010) use the West 2010 corpus described in Section 3.4.3. Its features are divided in two groups: (a) *simple features*, which are calculated, mainly, from the metadata of a single edit and (b) *aggregate features*, which are calculated from the history of *offending edits* (OE).

Simple features are:

**Time-of-day and day-of-week** . IP addresses are visible for anonymous users and they are used to geolocate the user. With this geographical data, the GMT offset is obtained and used to compute the time-of-day and day-of-week of the edit in the author's local time.

**Time since user registration** . Time of user registration is estimated by taking the timestamp of his first edit.

**Time since last article edit** . Difference between the time of the current edit and the previous one.

**Time since last user Offending Edit** . Difference between the time of the current edit and the last OE by the same user. This is undefined when the user has no OE.

**Revision comment length** . Length of the comment written by the user as summary of his edit.

**Registered user properties** . User properties indicating if he is anonymous or not, if he has special privileges or if it is a bot.

In order to calculate aggregate features, the following variables and functions are defined:

- $\alpha$  is an entity.
- $G$  is a spatial grouping function.
- $g = G(\alpha)$  is the group  $\alpha$  belongs to according its  $G$  function.

---

<sup>32</sup>We could see this as *metadata-based reputation* as opposed to *content-based reputation*.

- $oe\_hist(g)$  returns a list of timestamps  $t_{oe}$  for every OE corresponding to an element in  $g$ .
- $decay(t)$  is a time-decay function for weighting elements according to its age. It is defined as  $decay(t) = 2^{\frac{\Delta t}{h}}$  where  $\Delta t = t_{now} - t_{oe}$  and  $h$  is the half-life.
- $rep(g)$  is the reputation of a group  $g$ , calculated as  $rep(g) = \sum_{t_{oe} \in oe\_hist(g)} \frac{decay(t_{oe})}{size(g)}$ .  
A high value of reputation is an indicator of vandalism.

Aggregate functions are:

**Article reputation** . Reputation of the article being edited. This is an application of  $rep(\alpha)$  where  $\alpha$  is the article. Timestamp  $t_{now}$  is the one corresponding to the edit being analyzed.

**User reputation** . Identical to article reputation, being  $\alpha$  the user.

**Category reputation** . For each category that the article belongs to,  $rep(c)$  is calculated, being  $c$  the set of articles in the category. The maximum  $rep(c)$  value is used for this feature.

**Country reputation** . Using geolocation,  $rep$  is applied with  $g$  the users of the same country. The  $size$  normalizer is the number of prior edits in the same country.

The authors have materialized this approach in the anti-vandalism assistance tool STiki.

Other interesting metadata features have been studied by **chichkov10**; **white10**





# Chapter 4

## Developing a Wikipedia Vandalism Detection System

### 4.1 Participation at PAN 2010

Our first approach to Wikipedia vandalism detection was part of our participation at PAN 2010, originally published in (Mola-Velasco 2010)<sup>1</sup>. We based our work upon (Potthast, Stein, and Gerling 2008) by refining and extending their feature set, as well as conducting a more exhaustive evaluation of different classification models. In this section, we describe in detail the developed system<sup>2</sup>. This system was the basis upon the rest of our contributions have been built.

#### 4.1.1 Preprocessing

Some features require tokenization as a preprocessing step. Given an edit, we tokenize the text of previous and current revisions with the following rules: (1) Any character sequence delimited by spacing is a token and (2) the following character sequences are independent tokens even if they are not delimited by spaces: ., /, :, ;, ", «, », ', |, ?, !, =, (, ), \*, [ [ , ] ], [ . ], { { , } }, { and }.

---

<sup>1</sup>Significant portions of the text of this section were copied from this previous article, although it has been reviewed exhaustively. Experimentation has been repeated, since previous results were calculated with 10-fold cross-validation using just the PAN-WVC-10 training set. A more extensive evaluation of classifiers has been conducted for this thesis.

<sup>2</sup>The resulting system is open source. Code is available on demand by requesting it to the author at [santiago.mola@bitsnbrains.net](mailto:santiago.mola@bitsnbrains.net). However, its development and maintainance has been discontinued in favor of more viable frameworks such as ClueBot-NG.

### 4.1.2 Features

In this section, we describe our feature set. A summary is presented in Table 4.1. Features marked with \* were already defined in (Potthast, Stein, and Gerling 2008) and those marked with † are modifications of features also defined in that work.

All our features are calculated using metadata and the text of single edits. They can be divided in three groups: Metadata, Text, and Language. Metadata-based features are the following:

**Anonymous\*** Whether the editor is anonymous or not.

Vandals are likely to be anonymous. This feature is used in a way or another in most antivandalism working bots such as ClueBot and AVBOT. In the PAN-WVC-10 training set (Potthast 2010) anonymous edits represent 29% of the regular edits and 87% of vandalism edits.

**Comment length\*** Length in characters of the edit summary.

Long comments might indicate regular editing and short or blank ones might suggest vandalism. However, this feature is quite weak, since leaving an empty comment in regular editing is a common practice.

**Size increment** Absolute increment of size, *i.e.*,  $|new| - |old|$ .

The value of this feature is already well-established since first-generation systems. For example, ClueBot uses various thresholds of size increment for its heuristics, *e.g.* a big size decrement is considered an indicator of blanking.

**Size ratio\*** Size of the new revision relative to the old revision, *i.e.*,  $\frac{1+|new|}{1+|old|}$ .

Complements size increment.

Text-based features are the following:

**Upper to lower ratio†** Uppercase to lowercase letters ratio, *i.e.*,  $\frac{1+|upper|}{1+|lower|}$ .

Vandals often do not follow capitalization rules, writing everything in lowercase or in uppercase.

**Upper to all ratio†** Uppercase letters to all letters to ratio, *i.e.*,  $\frac{1+|upper|}{1+|lower|+|upper|}$ .

**Digit ratio** Digit to all characters ratio, i.e.,  $\frac{1+|digit|}{1+|all|}$ .

This feature helps to spot minor edits that only change numbers. This might help to find some cases of subtle vandalism where the vandal changes arbitrarily a date or a number to introduce misinformation.

**Non-alphanumeric ratio** Non-alphanumeric to all characters ratio, i.e.,  $\frac{1+|nonalphanumeric|}{1+|all|}$ .

An excess of non-alphanumeric characters in short texts might indicate use of emoticons, excessive use of exclamation marks or gibberish.

**Character diversity** Measure of different characters compared to the length of inserted text, given by the expression  $length^{\frac{1}{different\ chars}}$ .

This feature helps to spot random keyboard hits and other non-sense.

**Character distribution**<sup>†</sup> Kullback-Leibler divergence of the character distribution of the inserted text with respect the expectation. Useful to detect non-sense.

**Compressibility**<sup>†</sup> Compression rate of inserted text using the LZW algorithm<sup>3</sup>.

Useful to detect non-sense, repetitions of the same character or words, etc.

**Good tokens** Number of tokens rarely used by vandals, mainly wiki-syntax elements (e.g. `__TOC__`, `<ref>`).

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

In long and well-established articles too many words that do not appear in the rest of the article indicates that the edit might be including non-sense or non-related content.

**Longest word**\* Length of the longest inserted word. Its value is 0 if there are no inserted words.

Useful to detect non-sense.

**Longest character sequence**\* Longest sequence of the same character in the inserted text.

---

<sup>3</sup>LZW was chosen after evaluating the behaviour of LZW, gzip and bzip2, although, an exhaustive comparison is still pending. We used the TIFF LZW algorithm (Adobe Developers Association 1992) as implemented in python-lzw 0.01 by Joe Bowers, available at <http://www.joe-bowers.com/static/lzw/>.

Long sequences of the same character are frequent in vandalism (*e.g. aagggggh-hhhhhh!!!!, soooooo huge*).

Our language-dependent features are based in counters of words in certain categories. Following (Potthast, Stein, and Gerling 2008), for each word category, two features are calculated: frequency and impact. Frequency is the frequency of these words relative to the total words inserted during the edit. Impact is the percentage by which the edit increases the amount of these words. Our word categories are:

**Vulgarisms**<sup>†</sup> Vulgar and offensive words (*e.g. fuck, suck, stupid*).

**Pronouns**<sup>†</sup> First and second person pronouns, including slang spellings (*e.g. I, you, ya*).

**Bias** Colloquial words with high bias (*e.g. coolest, huge*).

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

**Bad** Hodgepodge category for colloquial contractions and some typos associated with bad (*e.g. wanna, gotcha*) and some typos associated with bad writing skills (*e.g. dosent*).

**All** A meta-category containing words from all the previous ones.

Table 4.1: Summary of features used in Mola-Velasco 2010.

Feature	Description
<i>Metadata</i>	
<b>Anonymous*</b>	Whether the editor is anonymous or not.
<b>Comment length*</b>	Length in characters of the edit summary.
<b>Size increment</b>	Absolute increment of size.
<b>Size ratio*</b>	Size of the new revision relative to the old revision.
<i>Text</i>	
<b>Upper to lower ratio<sup>†</sup></b>	Uppercase to lowercase letters ratio.
<b>Upper to all ratio<sup>†</sup></b>	Uppercase letters to all letters to ratio.
<b>Digit ratio</b>	Digit to all characters ratio.
<b>Non-alphanumeric ratio</b>	Non-alphanumeric to all characters ratio.
<b>Character diversity</b>	$length \frac{1}{different\ chars}$ .
<b>Character distribution<sup>†</sup></b>	KLd between the character distribution of the inserted text and the expectation.
<b>Compressibility<sup>†</sup></b>	Compression rate of inserted text using LZW.
<b>Good tokens</b>	Number of tokens rarely used by vandals, mainly wiki-syntax elements.
<b>Average frequency*</b>	<b>term</b> Average relative frequency of inserted words in the new revision.
<b>Longest word*</b>	Length of the longest inserted word.
<b>Longest character sequence*</b>	Longest sequence of the same character in the inserted text.
<i>Language</i>	
<b>Vulgarisms<sup>†</sup></b>	Vulgar and offensive words.
<b>Pronouns<sup>†</sup></b>	First and second person pronouns, including slang spellings.
<b>Bias</b>	Colloquial words with high bias.
<b>Sex</b>	Non-vulgar sex-related words.
<b>Bad</b>	Hodgepodge category for colloquial contractions and some typos associated with bad and typos associated with bad writing skills.
<b>All</b>	A meta-category containing words from all the previous ones.

### 4.1.3 Classification

Classification has been conducted using the Weka framework (Hall et al. 2009). After preliminary evaluations, we have tried to choose classifiers which fulfill all or most of the following conditions: (a) require little or no preprocessing of data,

(*b*) require little parameter adjustment, (*c*) do implicit feature selection, (*d*) are resistant to noise and outliers and (*e*) are resistant to severe class imbalance.

Our baseline classifier is C4.5 decision tree (Quinlan 1993) which is a well-established algorithm and, to some extent, fulfills our criteria. LogitBoost (Friedman, Hastie, and Tibshirani 2000) and Random Forest (Breiman 2001) are attractive because of their implicit feature selection, generalization properties and a low number of primary parameters that need tuning. We have also included Bagging with C4.5 (Breiman 1996) and Support Vector Machines with linear and radial kernels (Joachims 1999; Vapnik 1998).

#### 4.1.4 Evaluation

For this thesis, we repeated and extended experiments in (Mola-Velasco 2010) using 10-fold stratified cross-validation<sup>4</sup>. Parameters for every classifier were the defaults of Weka 3.6<sup>5</sup> except SVM (Radial) which used  $L^+ = 0.5$  after tuning; LogitBoost, whose results are presented for 10 and 200 iterations; and Random Forest, whose results are presented for 500 and 1000 iterations. We also present the results for theoretic classifiers giving random results, and classifying every edit as positive.

In Table 4.2 we present the performance for each classifier. In Figures 4.1 and 4.2 we show Precision-Recall curves and F-Measure curves, respectively. Bagging C4.5 presents a bad performance and tuning of parameters did not help to improve it significantly. Support Vector Machines also presented a very low performance, however, we think that it is still worth to conduct an exhaustive parameter tuning for it. LogitBoost and Random Forest were clearly superior to the rest of classifiers. LogitBoost slightly outperforms Random Forest. This is surprising since LogitBoost assumes variable independence between features and it is not exploiting the relations between features that we expect by intuition. This suggests that there is still room for improvement in the classification model choice.

---

<sup>4</sup>10-fold stratified cross-validation is generally accepted as validation method with low variance and pessimistic bias (Kohavi 1995).

<sup>5</sup>API documentation available at <http://weka.sourceforge.net/doc.stable/>.

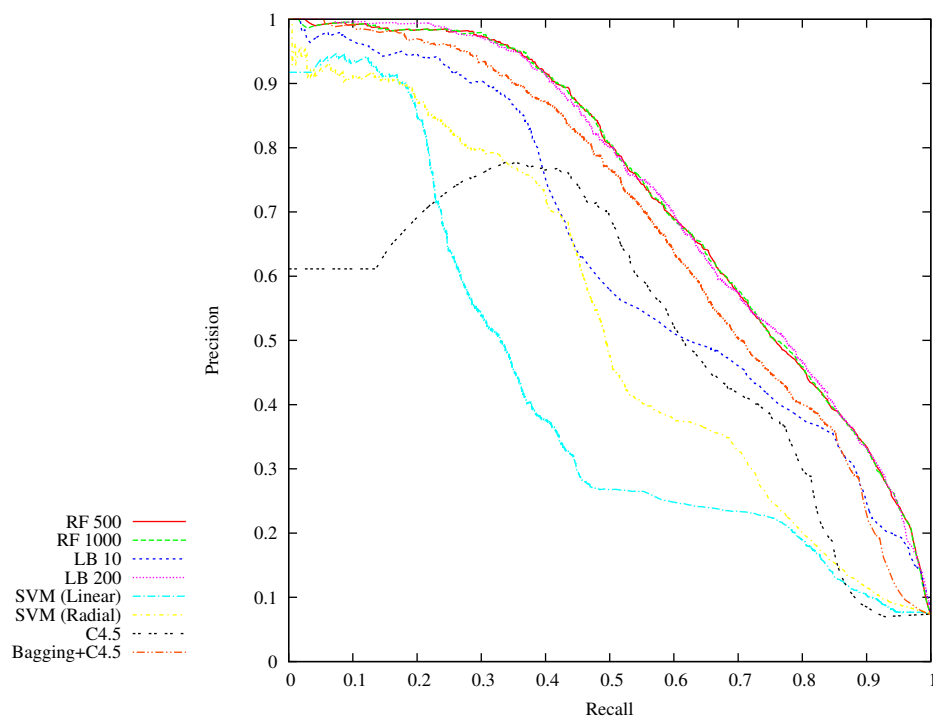


Figure 4.1: Precision-Recall curves for (Mola-Velasco 2010).

Table 4.2: Performance of classifiers using (Mola-Velasco 2010).

Model	AUC-PR	AUC-ROC
Bagging+C4.5	0.68541	0.91332
C4.5	0.51779	0.84123
LogitBoost (10 iter.)	0.63229	0.92824
LogitBoost (200 iter.)	<b>0.73058</b>	<b>0.94759</b>
Random Forest (500 iter.)	<b>0.72982</b>	<b>0.94681</b>
Random Forest (1000 iter.)	<b>0.73018</b>	<b>0.94666</b>
SVM (Linear)	0.42201	0.80813
SVM (Radial)	0.55645	0.87066

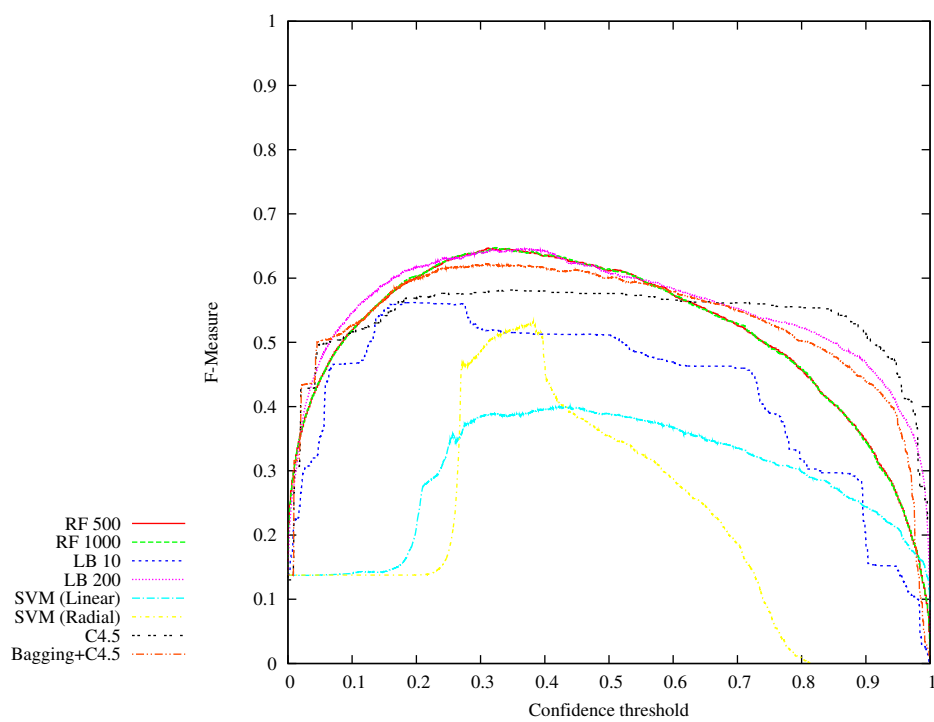


Figure 4.2: F-Measure curves for (Mola-Velasco 2010).



## 4.2 Combining Natural Language, Metadata, and Reputation

In (Adler et al. 2011) we evaluated the combination of features from different approaches<sup>6,7</sup>:

- Textual and Language features, mainly based on (Mola-Velasco 2010). Explained in-depth in Section 4.1.
- Metadata features, mainly based on (West, Kannan, and Lee 2010). Explained in Section 3.6.6.
- Reputation features, mainly based on (Adler, Alfaro, and Pye 2010). Explained in Section 3.6.5.

We also divide features into three classes according to the complexity required to compute them, and according to the difficulty of generalizing them across multiple languages. These classes are: Metadata, Text, Reputation, and Language, abbreviated as **M**, **T**, **R**, and **L**, respectively.

### 4.2.1 Features

#### Metadata

*Metadata* (M) refers to properties of a revision that are immediately available, such as the identity of the editor, or the timestamp of the edit. This is an important class of features because it has minimal computational complexity. Beyond the properties of each revision found directly in the database (*e.g.* whether the editor is anonymous, used by nearly every previous work), there are some examples that we feel expose the unexpected similarities in vandal behavior:

- **Time since article last edited** (West, Kannan, and Lee 2010). Highly-edited articles are frequent targets of vandalism. Similarly, quick fluctuations in content may be indicative of edit wars or other controversy.

---

<sup>6</sup>Significant parts of this sections are copied verbatim from (Adler et al. 2011), which is Copyright © 2011 Springer-Verlag GmbH Berlin Heidelberg. According to this, any copyright assignment or license of this thesis is not applied to this section.

<sup>7</sup>We would like to thank again Thomas B. Adler, Luca de Alfaro, Ian Pye, Andrew West, Sampath Kannan and Insup Lee for their work.

- **Local time-of-day** and **day-of-week** (West, Kannan, and Lee 2010). Using IP geolocation, it is possible to determine the *local* time when an edit was made. Evidence shows vandalism is most prominent during weekday “school/office hours.”
- **Revision comment length** (Adler, Alfaro, and Pye 2010; Mola-Velasco 2010; West, Kannan, and Lee 2010). Vandals decline to follow community convention by leaving either very short revision comments or very long ones.

## Text

We label as *Text* (T) those language-independent<sup>8</sup> features derived from analysis of the edit content. Very long articles may require a significant amount of processing. As the content of the edit is the true guide to its usefulness, there are several ideas for how to measure that property:

- **Uppercase ratio** and **digit ratio** (Mola-Velasco 2010; West, Kannan, and Lee 2010). Vandals sometimes will add text consisting primarily of capital letters to attract attention; others will change only numerical content. These ratios (and similar ones (Mola-Velasco 2010)) create features which capture behaviors observed in vandals.
- **Average** and **minimum edit quality** (Adler, Alfaro, and Pye 2010) (Historic only). Comparing the content of an edit against a future version of the article provides a way to measure the Wikipedia community’s approval of the edit (Adler and Alfaro 2007; Druck, Miklau, and McCallum 2008). To address the issue of edit warring, the comparison is done against several future revisions. This feature uses edit distance (rather than the blunt detection of reverts) to produce an implicit quality judgement by later edits; see (Adler and Alfaro 2007).

---

<sup>8</sup>Not all of these features are strictly language-independent. Some of them assume that uppercase and lowercase are defined in the writing system, making them applicable to most Indo-European languages using Latin, Greek or Cyrillic alphabets. However, they should serve as source of inspiration for language-specific features in languages with different writing systems. For example, in Japanese, the features Kanji, Hiragana, Katakana and Latin characters ratio could be added.

## Language

Similar to text features, *Language* (L) features must inspect edit content. A distinction is made because these features require expert knowledge about the (natural) language. Thus, these features require effort to be re-implemented for each different language. Some of the features included in our analysis are:

- **Pronoun frequency** and **pronoun impact** (Mola-Velasco 2010). The use of first and second-person pronouns, including slang spellings, is indicative of a biased style of writing discouraged on Wikipedia (non-neutral point-of-view). *Frequency* considers the ratio of first and second-person pronouns relative to the size of the edit. *Impact* is the percentage increase in first and second-person pronouns that the edit contributes to the overall article.
- **Biased** and **bad words** (Mola-Velasco 2010). Certain words indicate a bias by the author (*e.g.* superlatives: “coolest”, “huge”), which is captured by a list of regular expressions. Similarly, a list of bad words captures edits which appear inappropriate for an encyclopedia (*e.g.* “wanna”, “gotcha”) and typos (*e.g.* “seperate”). Both these lists have corresponding frequency and impact features that indicate how much they dominate the edit and increase the presence of biased or bad words in the overall article.

## Reputation

We consider a feature in the *Reputation* (R) category if it necessitates extensive historical processing of Wikipedia to produce a feature value. The high cost of this computational complexity is sometimes mitigated by the ability to build on earlier computations, using incremental calculations.

- **User reputation** (Adler, Alfaro, and Pye 2010) (Historic only<sup>9</sup>) User reputation as computed by WikiTrust (Adler and Alfaro 2007). The intuition is that users who have a history of good contributions, and therefore high reputation, are unlikely to commit vandalism.
- **Country reputation** (West, Kannan, and Lee 2010). For anonymous/IP edits, it is useful to consider the geographic region from which an edit origi-

---

<sup>9</sup>In a live system, user reputation is available at the time a user makes an edit, and therefore, user reputation is suitable for immediate vandalism detection. However, since WikiTrust only stores the current reputation of users, *ex post facto* analysis was not possible for this study.

nates. This feature represents the likelihood that an editor from a particular country is a vandal, by aggregating behavior histories from that same region. Location is determined by geo-locating the IP address of the editor.

- **Previous and current text trust histogram** (Adler, Alfaro, and Pye 2010). When high-reputation users revise an article and leave text intact, that text accrues reputation, called “trust” (Adler, Alfaro, and Pye 2010). Features are (1) the histogram of word trust in the edit, and (2) the difference between the histogram before and after the edit.

### Summary

Table 4.3 summarizes all features used, their classes and whether they are used in immediate detection or only in historic detection.

Table 4.3: Comprehensive listing of features used, organized by class. Note that features in the “!Z” (not zero-delay) class are those that are only appropriate for historical vandalism detection. In the SRC column, A stands for (Adler, Alfaro, and Pye 2010), M for (Mola-Velasco 2010) and W for (West, Kannan, and Lee 2010). Extrated from (Adler et al. 2011).

FEATURE	CLS	SRC	DESCRIPTION
IS_REGISTERED	M	A/M/W	Whether editor is anonymous/registered (boolean)
COMMENT_LENGTH	M	A/M/W	Length (in chars) of revision comment left
SIZE_CHANGE	M	A/M/W	Size difference between prev. and current versions
TIME_SINCE_PAGE	M	A/W	Time since article (of edit) last modified
TIME_OF_DAY	M	A/W	Time when edit made (UTC, or local w/geolocation)
DAY_OF_WEEK	M	W	Local day-of-week when edit made, per geolocation
TIME_SINCE_REG	M	W	Time since editor’s first Wikipedia edit
TIME_SINCE_VAND	M	W	Time since editor last caught vandalizing
SIZE_RATIO	M	M	Size of new article version relative to new one
PREV_SAME_AUTH	M	A	Is author of current edit same as previous? (boolean)
REP_EDITOR	R	W	Reputation for editor via behavior history
REP_COUNTRY	R	W	Reputation for geographical region (editor groups)
REP_ARTICLE	R	W	Reputation for article (on which edit was made)
REP_CATEGORY	R	W	Reputation for topical category (article groups)
WT_HIST	R	A	Histogram of text trust distribution after edit
WT_PREV_HIST_N	R	A	Histogram of text trust distribution before edit
WT_DELT_HIST_N	R	A	Change in text trust histogram due to edit
DIGIT_RATIO	T	M	Ratio of numerical chars. to all chars.
ALPHANUM_RATIO	T	M	Ratio of alpha-numeric chars. to all chars.
UPPER_RATIO	T	M	Ratio of upper-case chars. to all chars.
UPPER_RATIO_OLD	T	M	Ratio of upper-case chars. to lower-case chars.
LONG_CHAR_SEQ	T	M	Length of longest consecutive sequence of single char.
LONG_WORD	T	M	Length of longest token
NEW_TERM_FREQ	T	M	Average relative frequency of inserted words
COMPRESS_LZW	T	M	Compression rate of inserted text, per LZW
CHAR_DIST	T	M	Kullback-Leibler divergence of char. distribution
PREV_LENGTH	T	M	Length of the previous version of the article
VULGARITY	L	M	Freq./impact of vulgar and offensive words
PRONOUNS	L	M	Freq./impact of first and second person pronouns
BIASED_WORDS	L	M	Freq./impact of colloquial words w/high bias
SEXUAL_WORDS	L	M	Freq./impact of non-vulgar sex-related words
MISC_BAD_WORDS	L	M	Freq./impact of miscellaneous typos/colloquialisms
ALL_BAD_WORDS	L	M	Freq./impact of previous five factors in combination
GOOD_WORDS	L	M	Freq./impact of “good words”; wiki-syntax elements
COMM_REVERT	L	A	Is rev. comment indicative of a revert? (boolean)
NEXT_ANON	!Z/M	A	Is the editor of the <i>next</i> edit registered? (boolean)
NEXT_SAME_AUTH	!Z/M	A	Is the editor of <i>next</i> edit same as current? (boolean)
NEXT_EDIT_TIME	!Z/M	A	Time between current edit and <i>next</i> on same page
JUDGES_NUM	!Z/M	A	Number of later edits useful for implicit feedback
NEXT_COMM_LGTH	!Z/M	A	Length of revision comment for <i>next</i> revision
NEXT_COMM_RV	!Z/L	A	Is <i>next</i> edit comment indicative of a revert? (boolean)
QUALITY_AVG	!Z/T	A	Average of implicit feedback from judges
QUALITY_MIN	!Z/T	A	Worst feedback from any judge
DISSENT_MAX	!Z/T	A	How close QUALITY_AVG is to QUALITY_MIN
REVERT_MAX	!Z/T	A	Max reverts possible given QUALITY_AVG
WT_REPUTATION	!Z/R	A	Editor rep. per WikiTrust (permitting future data)
JUDGES_WGHT	!Z/R	A	Measure of relevance of implicit feedback

## 4.2.2 Evaluation

We evaluate the performance of (Mola-Velasco 2010), (West, Kannan, and Lee 2010) and (Adler, Alfaro, and Pye 2010) as well as different feature combinations. As in our previous experiments, we evaluate each model using 10-fold cross-validation on our version of the PAN-WVC-10 corpus. Each set of features has been used to train a LogitBoost<sup>10</sup> and a Random Forest<sup>11</sup> model, since those clearly outperformed other models<sup>12</sup>. Evaluation measures are AUC-PR and AUC-ROC.

## 4.2.3 Results and Discussion

In this section, we present results and discussion of our experiments using different combinations of features. Tables 4.4 and 4.5 summarizes the performance of these subsets for the immediate and historic detection tasks, respectively.

Figures 4.3 and 4.4 show Performance-Recall and F-Measures curves for LogitBoost and Random Forest classifiers using all features. Both have similar performance. Random Forest outperforms LogitBoost on small feature sets, while LogitBoost obtained the best result with the combination of all features. Both results are likely to be inadequate parameter tuning with respect the amount of features and indicates that we must conduct a systematic and more extensive parameter tuning on both algorithms. LogitBoost assumes features to be independent<sup>13</sup>, which is far from being the case in this task, so the performance compared to Random Forest makes us think that parameters of the later are far from being properly adjusted.

In Figures 4.5 and 4.6 we show precision-recall curves for each system, using Random Forest and distinguishing between immediate and historic vandalism cases, respectively. Figures 4.7 and 4.8 show F-Measure curves. Only Adler, Alfaro, and Pye (2010) consider features explicitly for the historic cases. We find a significant increase in performance when transitioning from immediate to historical detection scenarios.

Analysis of our feature taxonomy, per Figures 4.9, 4.10, 4.11 and 4.12, leads to some additional observations in a comparison between immediate and historic

---

<sup>10</sup>Using 200 iterations.

<sup>11</sup>Using 500 trees.

<sup>12</sup>See Section 4.1.4.

<sup>13</sup>As long as we use Decision Stumps as our weak learner.

vandalism tasks:

- The most obvious is the improvement in the performance of the Language (L) set, due entirely to the **next comment revert** feature. The feature evaluates whether the revision comment for the next edit contains the word “revert” or “rv,” which is used to indicate that the prior edit was vandalism (Adler, Alfaro, and Pye 2010). This is no surprise since this feature has been widely used to automatically annotate vandalism datasets.
- Both Metadata (M) and Text (T) show impressive gains in going from the *immediate* task to the *historic* task. For Metadata, our investigation points to `NEXT_EDIT_TIME` as being the primary contributor. This is likely to be due to two facts: (a) obvious vandalism is often reverted very quickly by antivandalism bots and human patrollers and (b) the most popular and edited pages are more likely to be vandalized. For Text, the set of features added in the *historic* task all relate to the implicit feedback given by later editors, showing a correlation between negative feedback and vandalism.
- A surprise in comparing the feature sets is that the predictive power of  $[M+T]$  and  $[M+T+R]$  are nearly identical in the historic setting. That is, once one knows the future community reaction to a particular edit, there is much less need to care about the past performance of the editor. We surmise that bad actors quickly discard their accounts or are anonymous, so reputation would be useful in the *immediate* detection case, but is less useful in *historic* detection.

One of the primary motivations for this work was to establish the significance of Language (L) features as compared to other features, because language features are more difficult to generate and maintain for each language edition of Wikipedia. In the case of immediate vandalism detection, we see the interesting scenario of the AUC-PR for  $[M+T+L]$  being nearly identical to that of  $[M+T+R]$ . That is, the predictive power of Language (L) and Reputation (R) features is nearly the same when there are already Metadata (M) and Text (T) features present. However, the improvement when all features are taken together suggests that Language (L) and Reputation (R) features capture different behavior patterns which only occasionally overlap.

Table 4.4: Performance of all feature combinations for immediate detection.

Features	LB		RF	
	PR	ROC	PR	ROC
Adler <i>et al.</i>	0.66010	0.94924	0.64120	0.94614
Mola-Velasco	0.72983	0.94681	0.73058	0.94759
West <i>et al.</i> <sup>14</sup>	0.51718	0.91536	0.51443	0.91264
Language	0.42030	0.74667	0.44726	0.76669
Metadata	0.44534	0.90165	0.47501	0.90429
Reputation	0.65321	0.94121	0.66138	0.94691
Text	0.52161	0.88372	0.50044	0.87375
M+T	0.68864	0.95013	0.68147	0.94319
M+T+L	0.76231	0.95842	0.76373	0.95652
M+T+R	0.79206	0.96931	0.79607	0.96838
All	0.84009	0.97494	0.83980	0.97462

Table 4.5: Performance of all feature combinations for historic detection.

Features	LB		RF	
	PR	ROC	PR	ROC
Adler <i>et al.</i>	0.74115	0.95870	0.72006	0.95770
Mola-Velasco	0.72983	0.94681	0.73058	0.94759
West <i>et al.</i> <sup>15</sup>	0.51718	0.91536	0.51443	0.91264
Language	0.58198	0.85736	0.61437	0.87186
Metadata	0.66236	0.93767	0.65028	0.93932
Reputation	0.68202	0.95171	0.67978	0.95405
Text	0.71602	0.95013	0.72425	0.95214
M+T	0.81245	0.97218	0.81741	0.97065
M+T+L	0.85050	0.97719	0.85220	0.97540
M+T+R	0.81608	0.97217	0.82580	0.97274
All	0.85344	0.97645	0.86055	0.97705



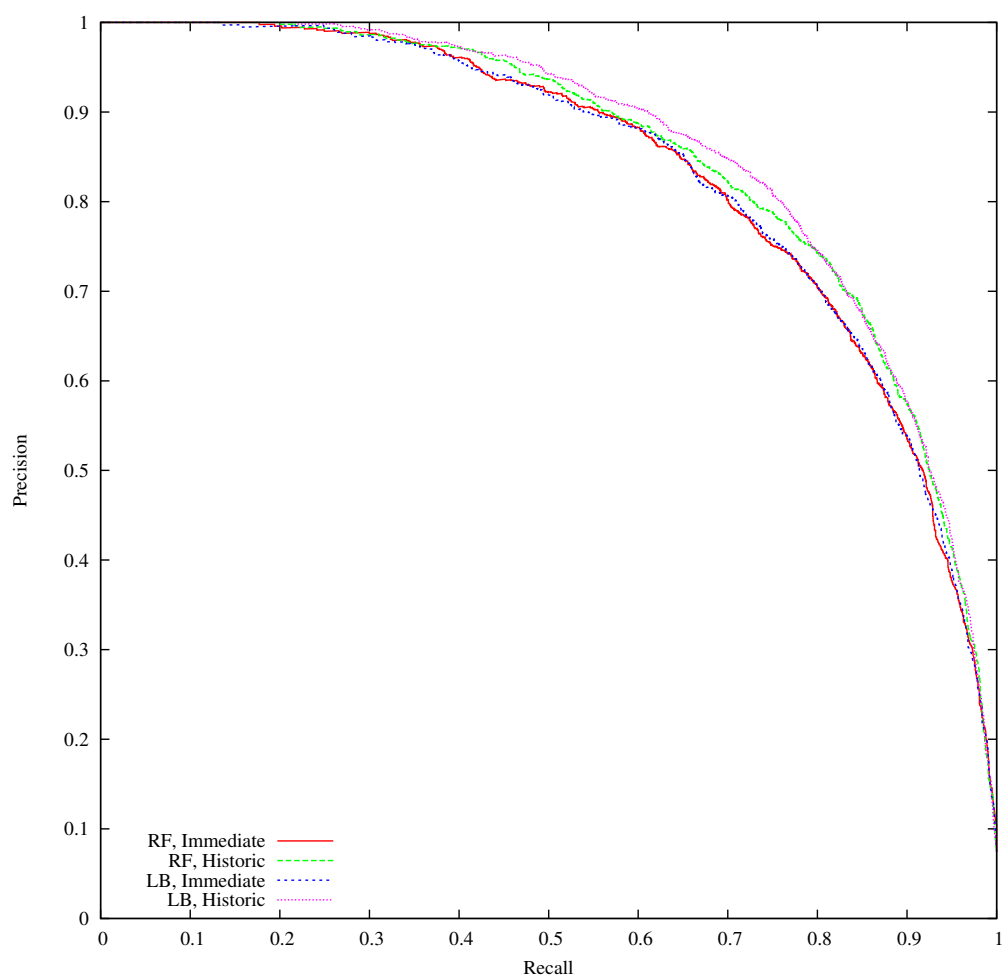


Figure 4.3: Precision-Recall curves for Logit Boost and Random Forest using all features, both for immediate and historic detection.

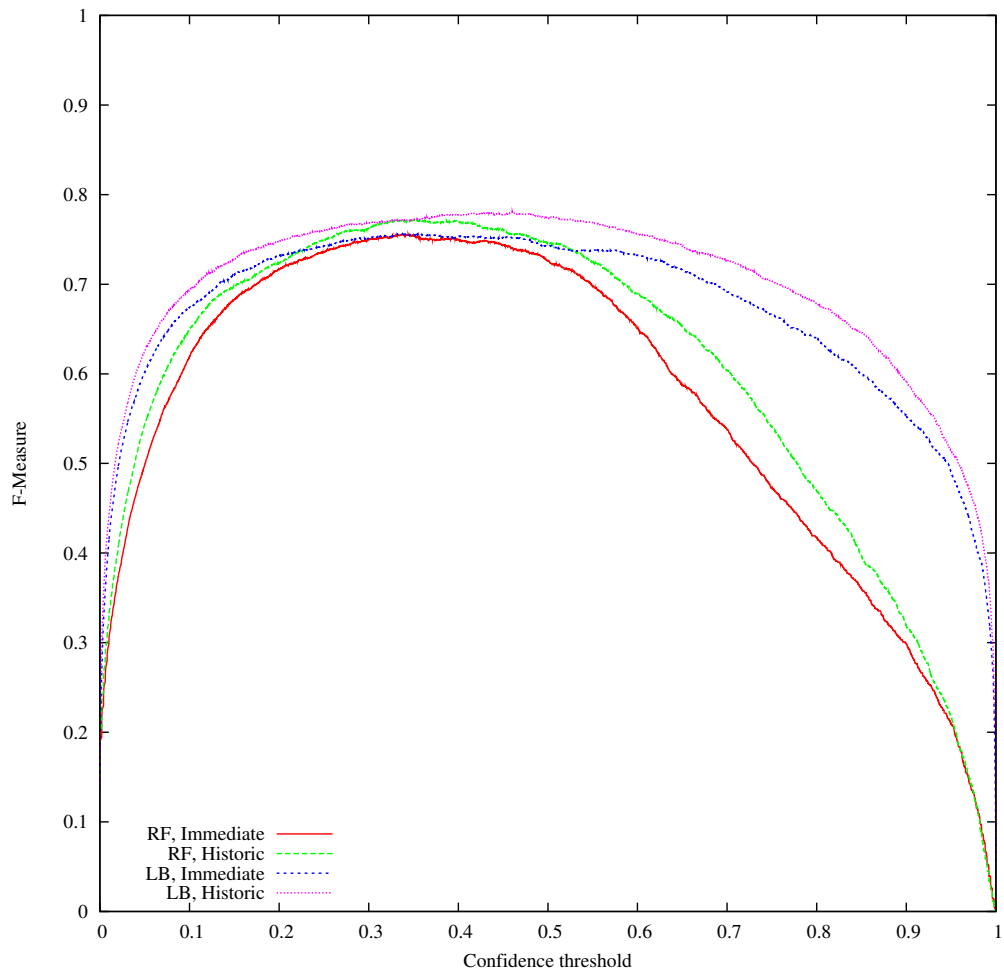


Figure 4.4: F-Measure curves for Logit Boost and Random Forest using all features, both for immediate and historic detection.

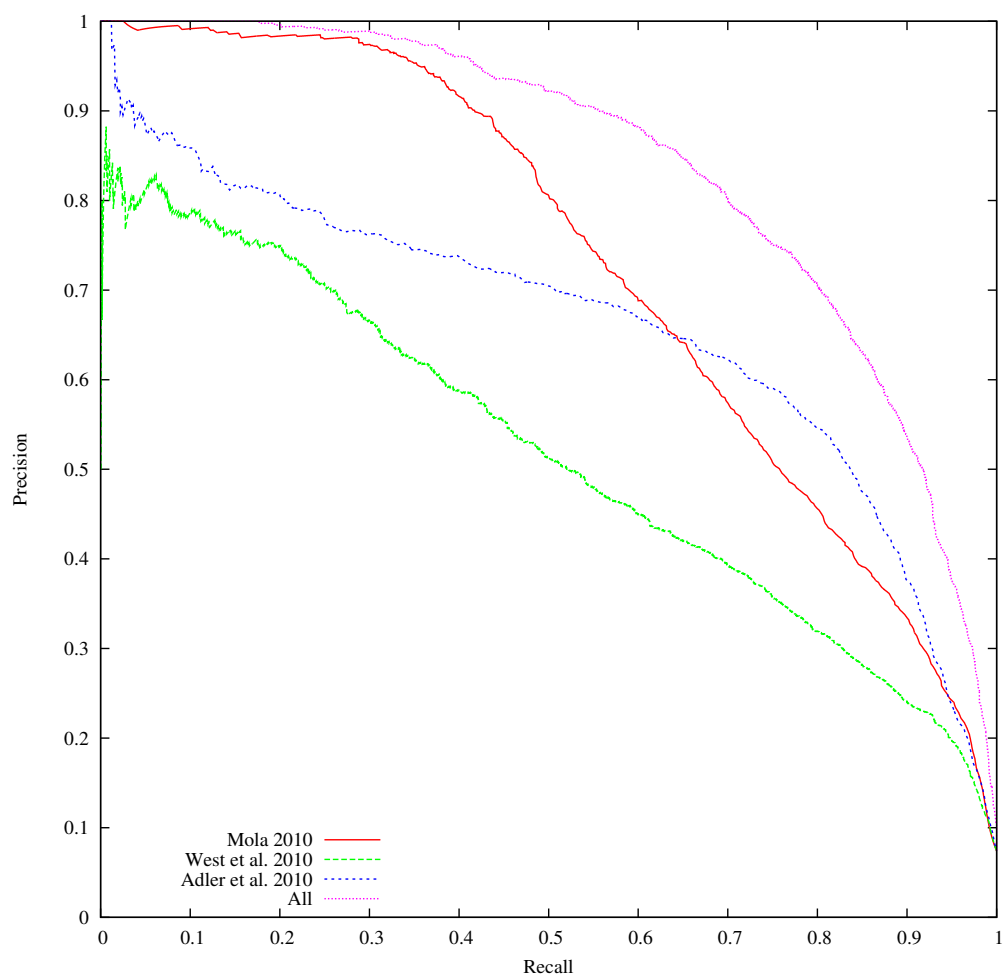


Figure 4.5: Precision-Recall curves for different systems in immediate detection, using Random Forest.

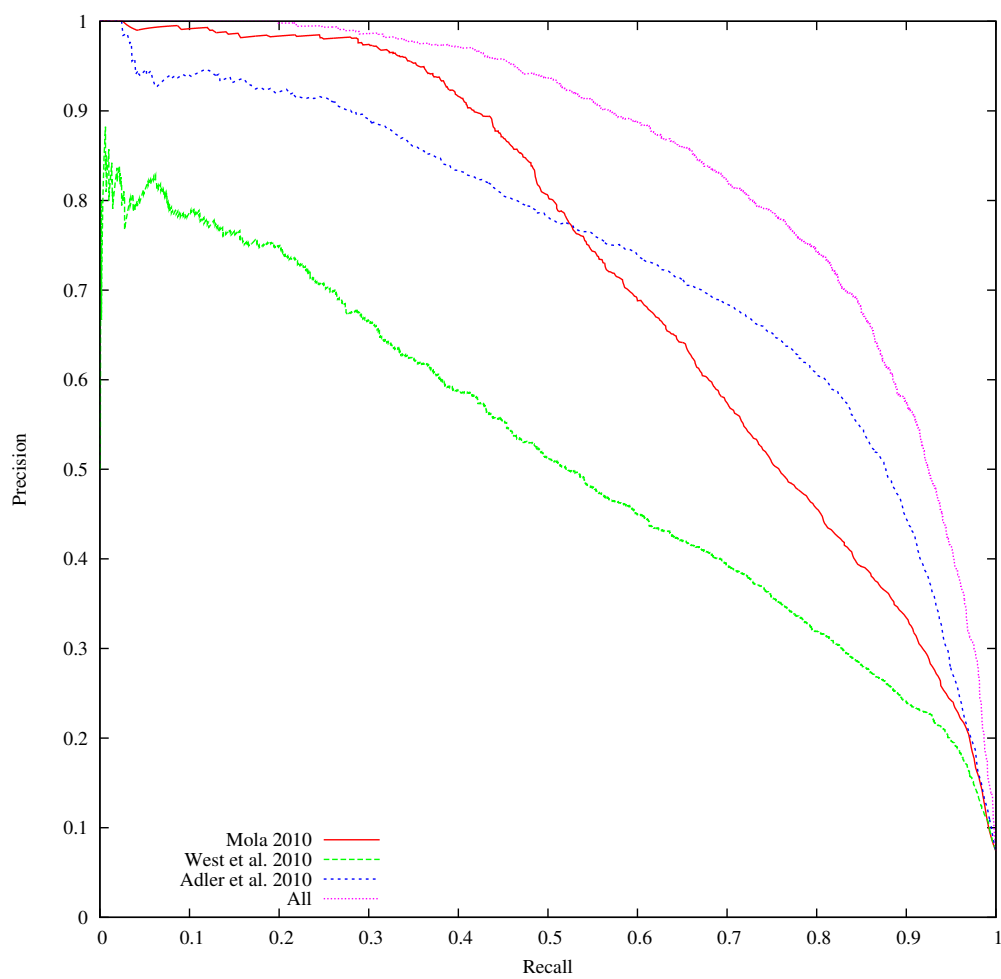


Figure 4.6: Precision-Recall curves for different systems in historic detection, using Random Forest.

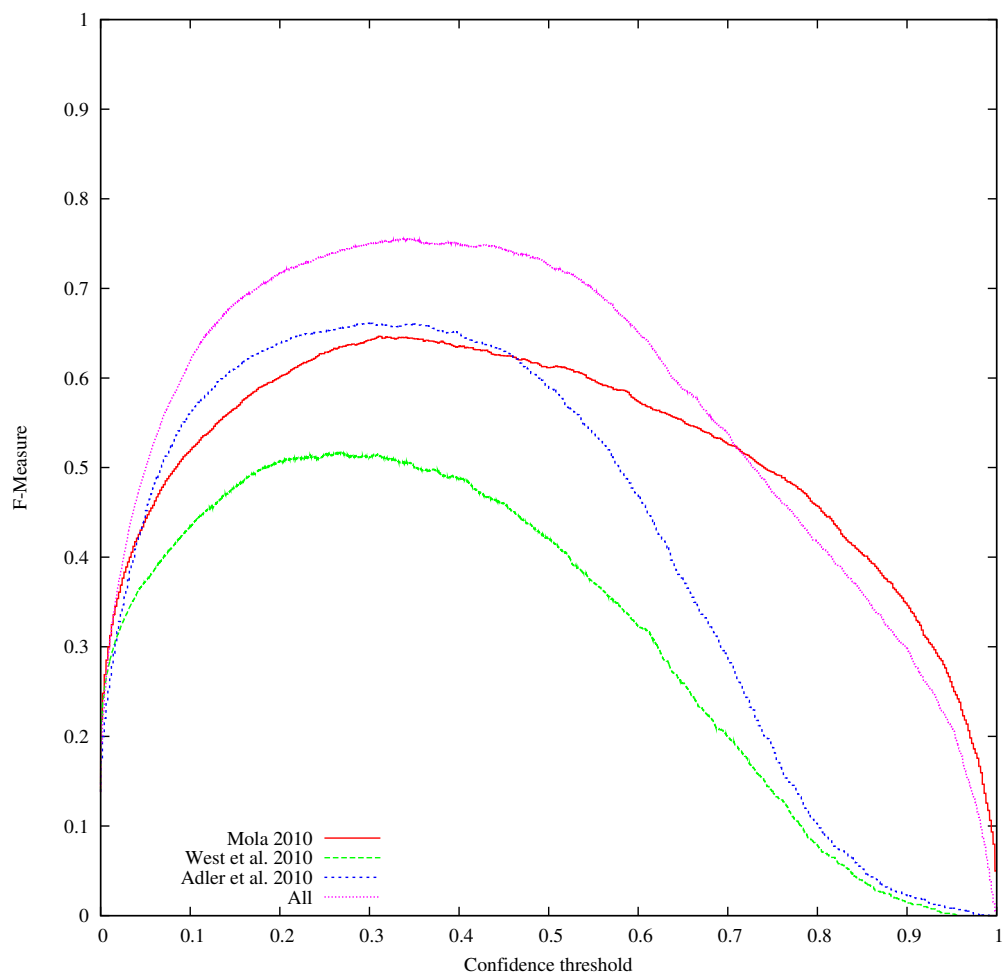


Figure 4.7: F-Measure curves for different systems in immediate detection, using Random Forest.

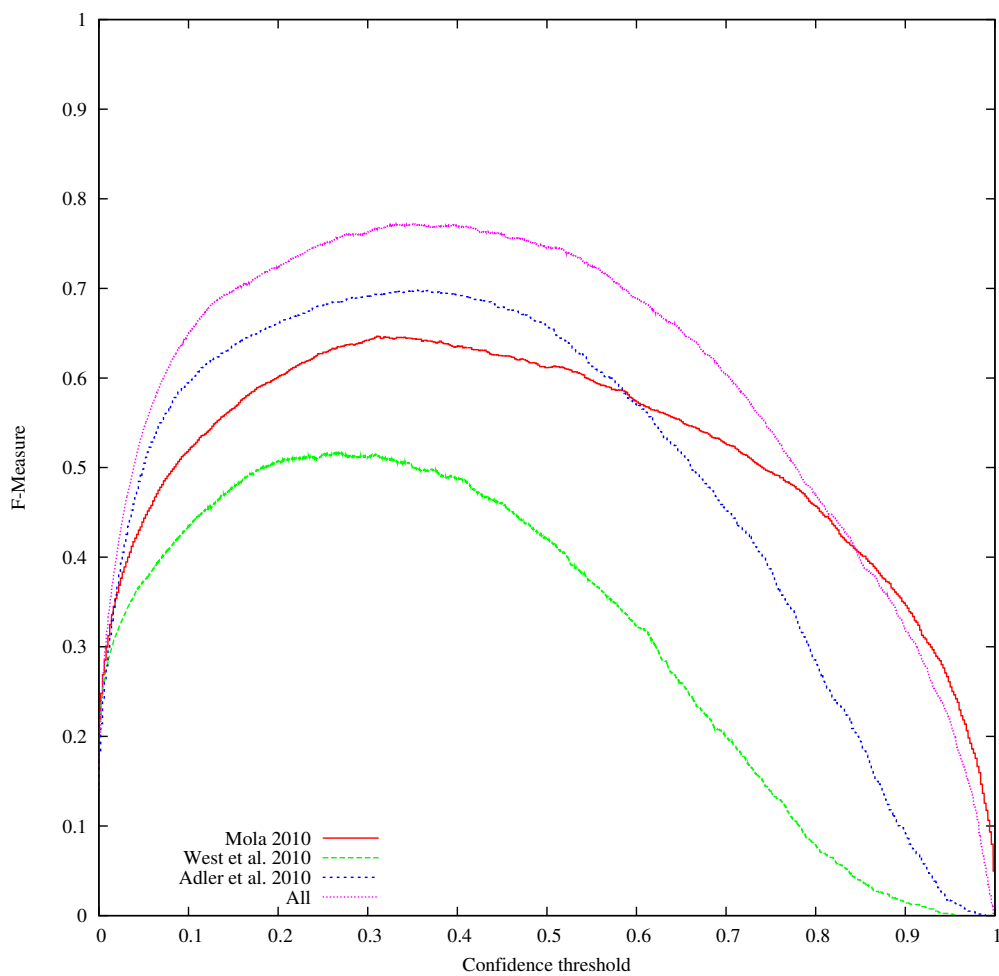


Figure 4.8: F-Measure curves for different systems in historic detection, using Random Forest.

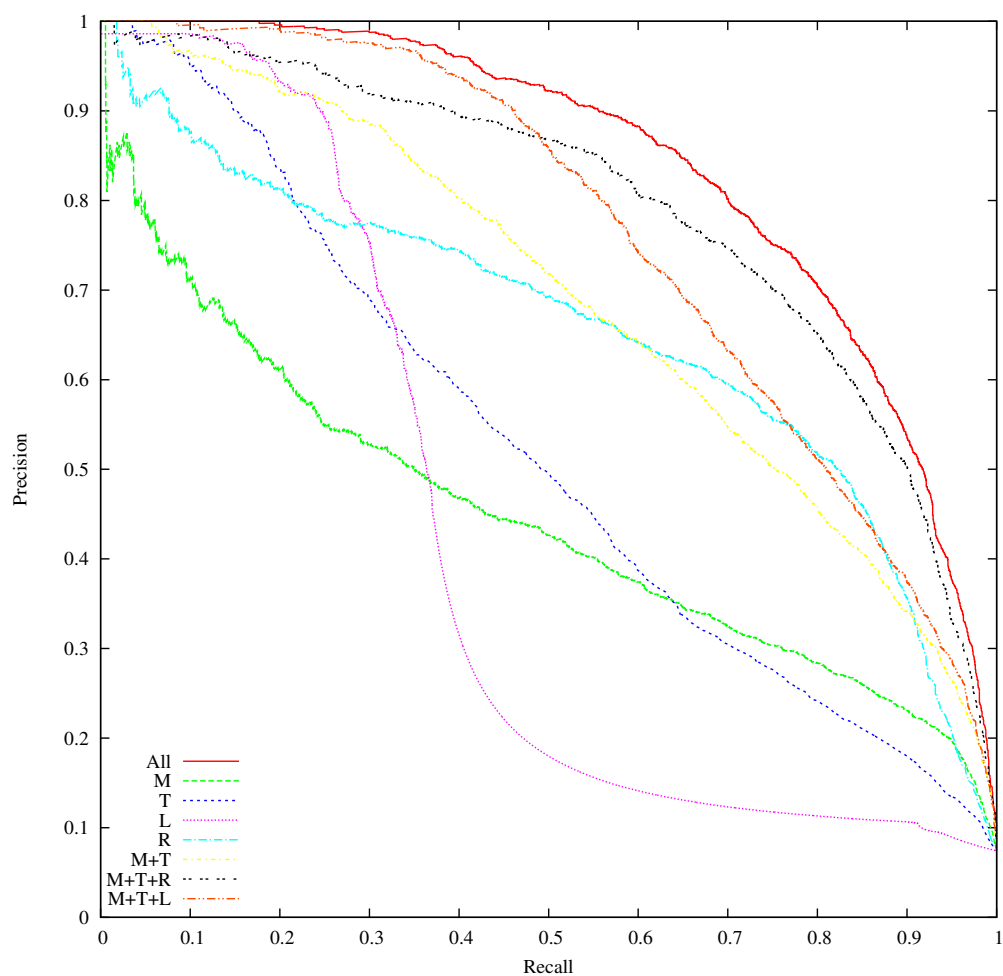


Figure 4.9: Precision-Recall curves for different feature classes in immediate detection, using Random Forest.

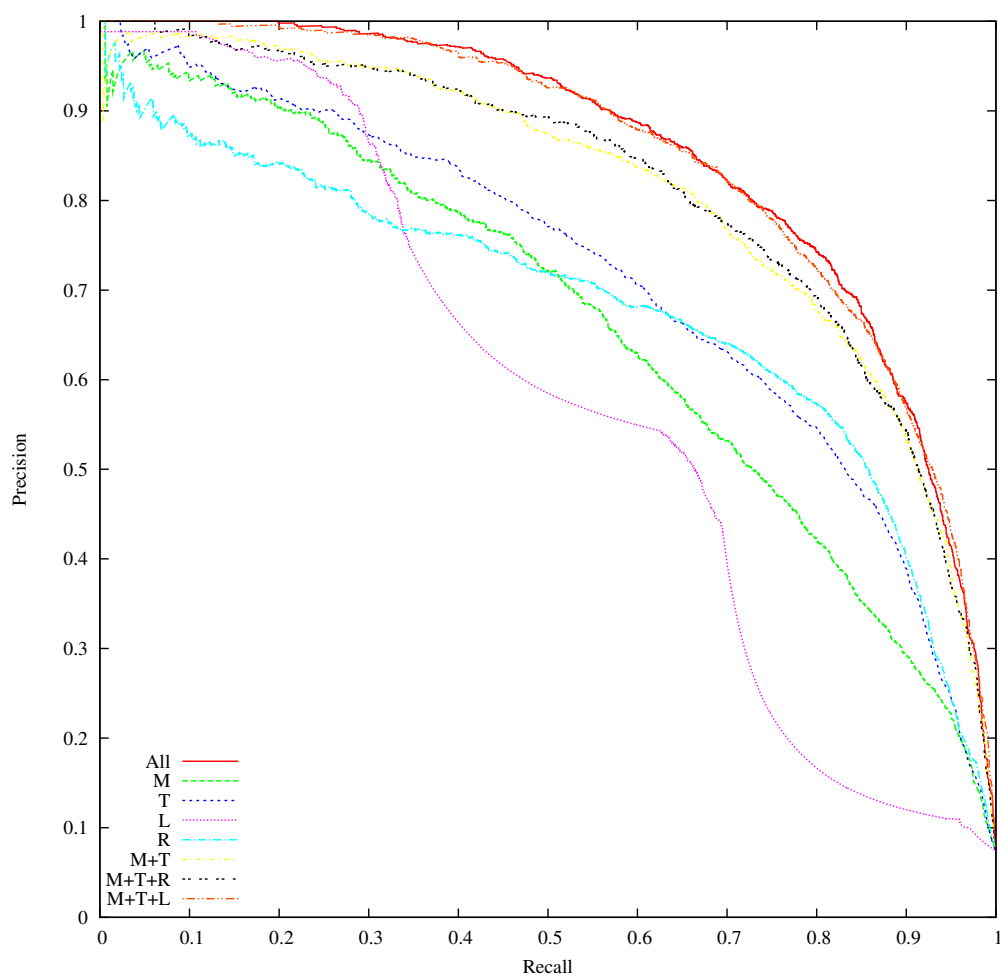


Figure 4.10: Precision-Recall curves for different feature classes in historic detection, using Random Forest.



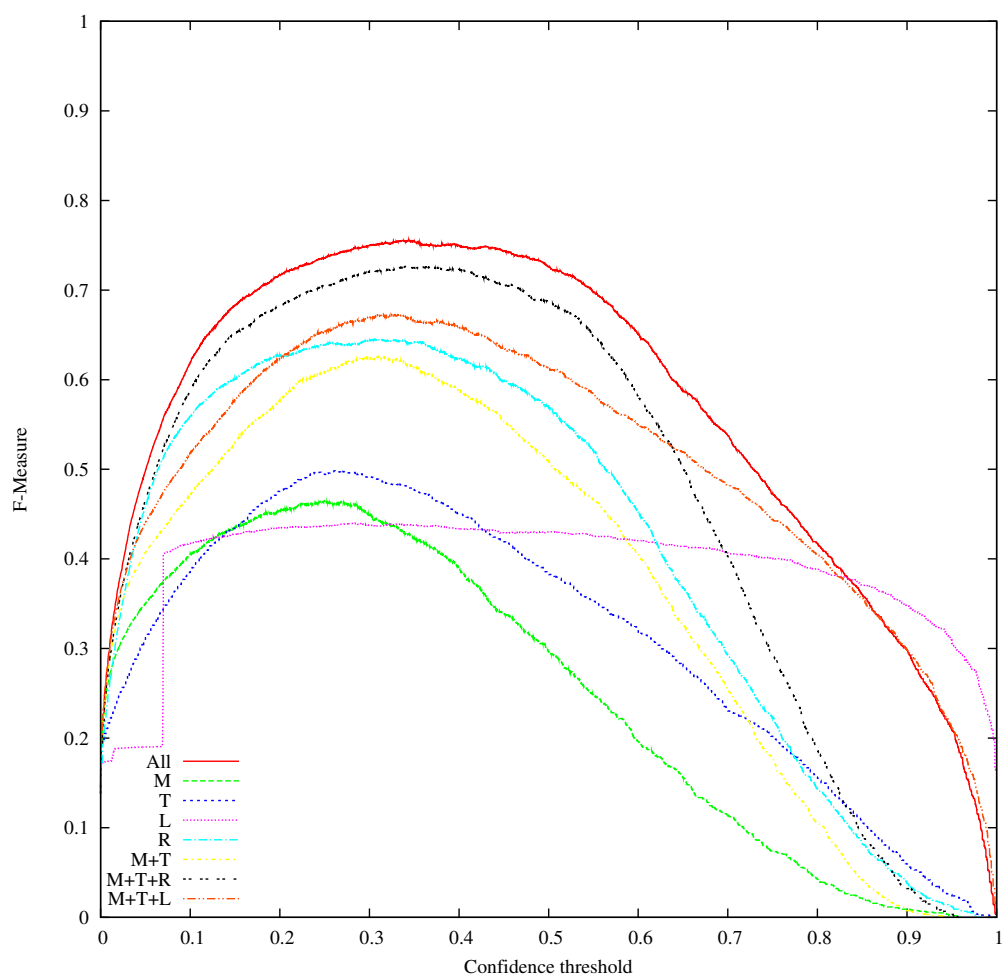


Figure 4.11: F-Measure curves for different feature classes in immediate detection, using Random Forest.

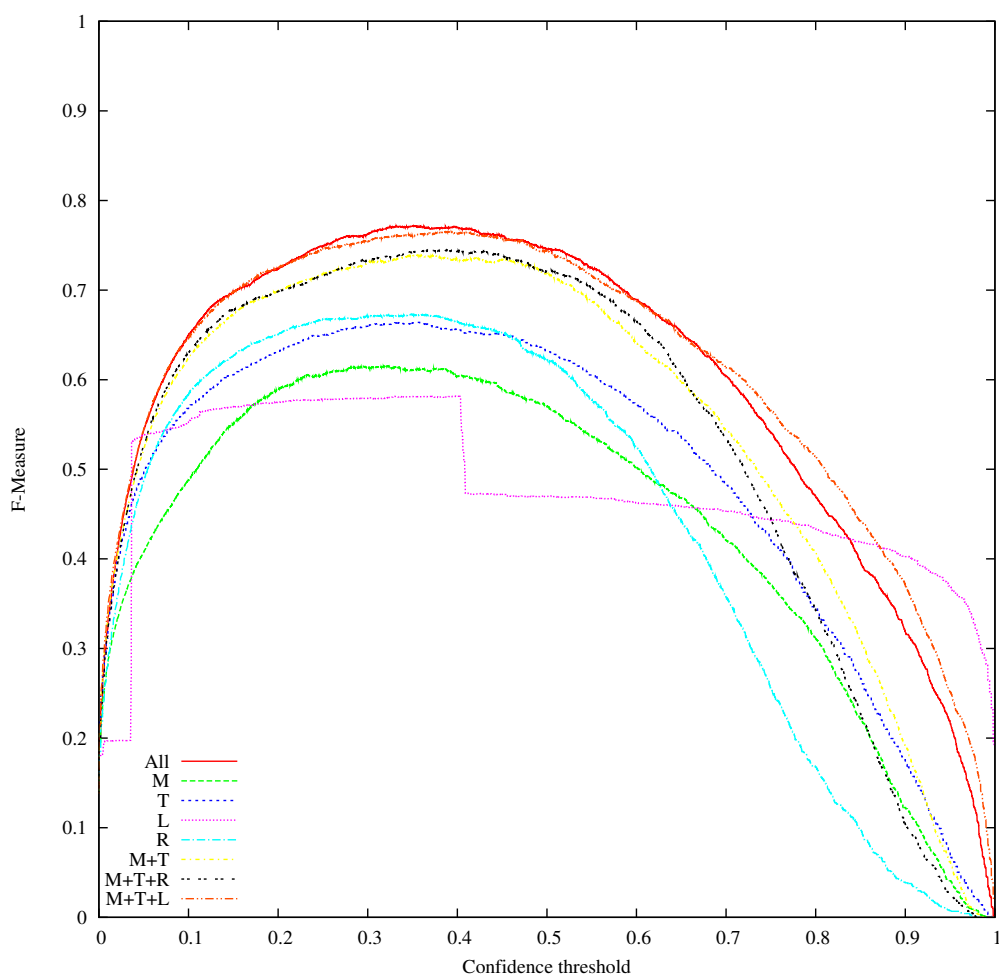


Figure 4.12: F-Measure curves for different feature classes in historic detection, using Random Forest.

#### 4.2.4 Conclusions

We proposed that solving the vandalism detection problem requires a more thorough exploration of the available feature space. We combined the features of three previous works, each representing a unique dimension in feature selection. Each feature was categorized as either metadata, text, reputation, or language, according to the nature of how they are computed and roughly corresponding to their computational complexity.

Our results outperform the winning system of the PAN 2010 competition (Mola-Velasco 2010), showing that the feature combination explored in this work considerably improves the state of the art (73% vs. 84% AUC-PR). Finally, a classifier combining all our feature sets could be suitable for the autonomous reversion of *some* bad edits – in a 99% precision setting, 32% recall was achieved.

We discovered that language features only provide an additional 4% of performance over the combined efforts of mostly language-independent features. This suggests that, given a proper corpus, our classifier might be applied to most Wikipedia editions.



# Chapter 5

## Conclusions

### 5.1 Contributions

The first and main contribution of this research has been a simple feature set for Wikipedia vandalism detection that won the 1st International Competition on Wikipedia Vandalism Detection, as published in (Mola-Velasco 2010; Potthast, Stein, and Holfeld 2010). The combination of this approach with the reputation system by Adler, Alfaro, and Pye (2010) and the metadata analysis by West, Kannan, and Lee (2010) achieved the one of the best results reported in the literature so far. A good performance was achieved even using mostly language-independent features. That means that our system might be adapted to other languages without major changes.

Further refinement of the classification models has been done for this thesis, previously unpublished, pushing performance to a new top mark.

### 5.2 Future Work

Future work should be focused in four areas:

1. Work on better corpora by:
  - a) Incorporating expert annotators as in ClueBot-NG dataset on a large-scale corpus such as PAN-WVC-11.
  - b) Explore active learning such as that proposed by Chin et al. (2010).

- c) Adding positive instances without human intervention using the method proposed by West, Kannan, and Lee (2010).
2. Refine current feature set. For example, vulgarisms are calculated ignoring caseness, but it would be desirable to consider caseness and style. For example, *dick* is very likely to be slang for penis and *DICK* is a strong indicator of vandalism, but *Dick* is more likely to be the diminutive for Richard or a surname.
3. Analyze and compare the relevance of all vandalism-indicating features proposed in the literature. Currently, there is no extensive study carrying out such a survey.
4. Integrate research developments into a production system such as ClueBot-NG. Working on a production system imposes technical restrictions on the features that can be used, since it must run in real-time.
5. Work on multi-lingual corpora, language independent features, and effective adaptation of production and research systems to new languages.

## References

- Adler, B. Thomas and Luca de Alfaro (2007). “A Content-Driven Reputation System for the Wikipedia”. In: *WWW '07: Proceedings of the 16th international conference on World Wide Web*. Banff, Alberta, Canada: ACM Press, pp. 261–270.
- Adler, B. Thomas, Luca de Alfaro, and Ian Pye (2010). “Detecting Wikipedia Vandalism using WikiTrust - Lab Report for PAN at CLEF 2010”. In: *CLEF (Notebook Papers/LABs/Workshops)*. Ed. by Martin Braschler, Donna Harman, and Emanuele Pianta.
- Adler, B. Thomas, Luca de Alfaro, Santiago M. Mola-Velasco, Paolo Rosso, and Andrew G. West (2011). “Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features”. In: *CICLing (2)*. Ed. by Alexander F. Gelbukh. Vol. 6609. Lecture Notes in Computer Science. Tokyo, Japan: Springer, pp. 277–288.
- Adobe Developers Association (June 1992). *TIFF Revision 6.0: Final*. Adobe Systems Incorporated.
- Bratko, Andrej, Gordon V. Cormack, Bogdan Filipic, Thomas R. Lynam, and Blaz Zupan (2006). “Spam Filtering Using Statistical Data Compression Models”. In: *Journal of Machine Learning Research* 6, pp. 2673–2698.
- Breiman, Leo (1996). “Bagging Predictors”. In: *Machine Learning* 24.2, pp. 123–140.
- (2001). “Random Forests”. In: *Machine Learning* 45.1, pp. 5–32.
- Carter, Jacobi (2010). “ClueBot and Vandalism on Wikipedia”. In:
- Chin, Si-Chi, W. Nick Street, Padmini Srinivasan, and David Eichmann (Apr. 2010). “Detecting Wikipedia Vandalism with Active Learning and Statistical Language Models”. In: *Proceedings of the Fourth Workshop on Information Credibility on the Web*. WICOW '10. Raleigh, North Carolina, USA: ACM, pp. 3–10.

- Davis, Jesse and Mark Goadrich (2006). “The Relationship Between Precision-Recall and ROC Curves”. In: *ICML '06: Proceedings of the 23rd international conference on Machine learning*. Pittsburgh, Pennsylvania: ACM, pp. 233–240.
- Druck, Gregory, Gerome Miklau, and Andrew McCallum (2008). “Learning to Predict the Quality of Contributions to Wikipedia”. In: *WikiAI'08: Proceedings of the Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy*. AAAI Press, pp. 7–12.
- Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2000). “Additive Logistic Regression: a Statistical View of Boosting”. In: *Annals of Statistics* 28.
- Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten (2009). “The WEKA Data Mining Software: an Update”. In: *SIGKDD Explorations* 11.1, pp. 10–18.
- Itakura, Kelly Y. and Charles L. A. Clarke (2009). “Using Dynamic Markov Compression to Detect Vandalism in the Wikipedia”. In: *SIGIR*. Ed. by James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel. ACM, pp. 822–823.
- Javanmardi, Sara, Cristina Videira Lopes, and Pierre Baldi (2010). “Modeling User Reputation in Wikis”. In: *Statistical Analysis and Data Mining* 3.2, pp. 126–139.
- Joachims, Thorsten (1999). “Making Large-Scale SVM Learning Practical”. In: *Advances in Kernel Methods - Support Vector Learning*. Ed. by Bernhard Schölkopf, Christopher J.C. Burges, and A. Smola. Cambridge, MA, USA: MIT Press.
- Kohavi, Ron (1995). “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. In: *IJCAI*, pp. 1137–1145.
- Mola-Velasco, Santiago M. (2010). “Wikipedia Vandalism Detection Through Machine Learning: Feature Review and New Proposals - Lab Report for PAN at CLEF 2010”. In: *Notebook Papers of CLEF 2010 LABs and Workshops*. Ed. by Martin Braschler, Donna Harman, and Emanuele Pianta. Padua, Italy.
- (2011). “Wikipedia Vandalism Detection”. In: *WWW (Companion Volume)*. Ed. by Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar. ACM, pp. 391–396.
- Posada, Emilio José Rodríguez and Wikipedia contributors (2011). *Anti-vandalism bot census*.



- Posada, Emilio José Rodríguez (2010). “AVBOT: detección y corrección de vandalismos en Wikipedia”. In: *NovATIca* 203, pp. 51–53.
- Potthast, Martin (July 2010). “Crowdsourcing a Wikipedia Vandalism Corpus”. In: *33rd Annual International ACM SIGIR Conference*. Ed. by Hsin-Hsi Chen, Efthimis N. Efthimiadis, Jaques Savoy, Fabio Crestani, and Stéphane Marchand-Maillet. ACM, pp. 789–790.
- Potthast, Martin and Robert Gerling (2007). *Wikipedia Vandalism Corpus Webis-WVC-07*. <http://www.uni-weimar.de/medien/webis/research/corpora>.
- Potthast, Martin, Benno Stein, and Robert Gerling (2008). “Automatic Vandalism Detection in Wikipedia”. In: *ECIR*. Ed. by Craig Macdonald, Iadh Ounis, Vasilis Plachouras, Ian Ruthven, and Ryen W. White. Vol. 4956. Lecture Notes in Computer Science. Springer, pp. 663–668.
- Potthast, Martin, Benno Stein, and Teresa Holfeld (2010). “Overview of the 1st International Competition on Wikipedia Vandalism Detection”. In: *CLEF (Notebook Papers/LABs/Workshops)*. Ed. by Martin Braschler, Donna Harman, and Emanuele Pianta.
- Priedhorsky, Reid, Jilin Chen, Shyong Tony, Katherine Panciera, Loren Terveen, and John Riedl (2007). “Creating, Destroying, and Restoring Value in Wikipedia”. In: *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*. Sanibel Island, Florida, USA: ACM, pp. 259–268.
- Quinlan, J. Ross (1993). *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Smets, Koen, Bart Goethals, and Brigitte Verdonk (2008). “Automatic Vandalism Detection in Wikipedia: Towards a Machine Learning Approach”. In: *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy (WikiAI08)*. AAAI Press, pp. 43–48.
- Vapnik, Vladimir N. (1998). *Statistical Learning Theory*. Wiley-Interscience.
- Viégas, Fernanda B., Martin Wattenberg, and Jushal Dave (2004). “Studying Cooperation and Conflict between Authors with History Flow Visualizations”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: ACM, pp. 575–582.
- Wang, William Yang and Kathleen McKeown (2010). ““Got You!”: Automatic Vandalism Detection in Wikipedia with Web-based Shallow Syntactic-Semantic

- Modeling”. In: *COLING*. Ed. by Chu-Ren Huang and Dan Jurafsky. Tsinghua University Press, pp. 1146–1154.
- West, Andrew G., Sampath Kannan, and Insup Lee (2010). “Detecting Wikipedia Vandalism via Spatio-Temporal Analysis of Revision Metadata”. In: *EUROSEC’10: Proceedings of the Third European Workshop on System Security*. Paris, France: ACM, pp. 22–28.
- Wikipedia (2011). *Wiki — Wikipedia, The Free Encyclopedia*. [Online; accessed 1-July-2011].
- Wikipedia contributors (2010). *Wikipedia: Vandalism – Wikipedia, The Free Encyclopedia*. [accessed 23-Oct-2010].
- (2011). *Vandalism Study 1 — Wikipedia, The Free Encyclopedia*. [Online; accessed 26-July-2011].
- Wöhner, Thomas and Ralf Peters (2009). “Assessing the quality of Wikipedia Articles with Lifecycle based Metrics”. In: *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*. WikiSym ’09. Orlando, Florida: ACM, 16:1–16:10.

# Appendix A

## Publications

During the research described in this thesis, we produced the following publications:

- Adler, B. Thomas, Luca de Alfaro, Santiago M. Mola-Velasco, Paolo Rosso, and Andrew G. West (2011). “Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features”. In: *CICLing (2)*. Ed. by Alexander F. Gelbukh. Vol. 6609. Lecture Notes in Computer Science. Tokyo, Japan: Springer, pp. 277–288.
- Mola-Velasco, Santiago M. (2010). “Wikipedia Vandalism Detection Through Machine Learning: Feature Review and New Proposals - Lab Report for PAN at CLEF 2010”. In: *Notebook Papers of CLEF 2010 LABs and Workshops*. Ed. by Martin Braschler, Donna Harman, and Emanuele Pianta. Padua, Italy.
- (2011). “Wikipedia Vandalism Detection”. In: *WWW (Companion Volume)*. Ed. by Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar. ACM, pp. 391–396.
- Mola-Velasco, Santiago M. and Paolo Rosso (2010). “Detección automática de vandalismo en Wikipedia”. In: *III Jornada de Innovación Docente*. ESTINF, Polytechnic University of Valencia. Valencia, Spain.