Martin Braschler and Donna Harman (Eds.):
Notebook Papers of CLEF 2010 LABs and Workshops, 22-23 September,
Padua, Italy. ISBN 978-88-904810-0-0. 2010.

Overview of the 1st International Competition on Wikipedia Vandalism Detection

Martin Potthast, Benno Stein, and Teresa Holfeld

Web Technology & Information Systems
Bauhaus-Universiät Weimar, Germany

pan@webis.de http://pan.webis.de

Abstract This paper overviews 9 vandalism detectors that have been developed and evaluated within PAN'10. We start with a survey of 55 different kinds of features employed in the detectors. Then, the detectors' performances are evaluated in detail based on precision, recall, and the receiver operating characteristic. Finally, we set up a meta detector that combines all detectors into one, which turns out to outperform even the best performing detector.

1 Introduction

Wikipedia allows everyone to edit its articles, and most of Wikipedia's editors do so for the best. Some, however, don't, and undoing their *vandalism* requires the time and effort of many. In recent years, a couple of tools have been developed to assist with detecting vandalism, but little is known about their detection performance, while research on vandalism detection is still in its infancy. To foster both research and development, we have organized the 1st competition on vandalism detection, held in conjunction with the 2010 CLEF conference. In this paper we overview the detection approaches of the 9 participating groups and evaluate their performance.

1.1 Vandalism Detection

We define an edit e as the transition from one Wikipedia article revision to another, where E is the set of all edits on Wikipedia. The task of a vandalism detector is to decide whether a given edit e has been done in bad faith or not. To address this task by means of machine learning three things are needed: a corpus $E_c \subset E$ of pre-classified edits, an edit model $\alpha : E \rightarrow \mathbf{E}$, and a classifier $c : \mathbf{E} \rightarrow \{0, 1\}$. The edit model maps an edit e onto a vector $\mathbf{e}$ of numerical values, called features, where each feature quantifies a certain characteristic of e that indicates vandalism. The classifier maps these feature vectors onto $\{0, 1\}$, where 0 denotes regular edits and 1 vandalism edits. Some classifiers map onto $[0, 1]$ instead, where values between 0 and 1 denote the classifier's confidence. To obtain a discrete, binary decision from such classifiers, a threshold $\tau \in [0, 1]$ is applied to map confidence values onto $\{0, 1\}$. In any case, the mapping of c is trained with a learning algorithm that uses the edits in E_c as examples. If c captures the concept of vandalism, based on α and E_c, then a previously unseen edit $e \in E \setminus E_c$ can be checked for vandalism by computing $c(\alpha(e)) > \tau$.

1.2 Evaluating Vandalism Detectors

To evaluate a vandalism detector, a corpus of pre-classified edits along with detection performance measures are required. The corpus is split into a training set and a test set. The former is used to train a vandalism detector, while the latter is used to measure its detection performance. For this purpose we have compiled the PAN Wikipedia vandalism corpus 2010, PAN-WVC-10 [10]. As detection performance measures we employ precision and recall as well as the receiver operating characteristic, ROC.

Vandalism Corpus. Until now, two Wikipedia vandalism corpora were available [11, 13], however, both have shortcomings which render them insufficient for evaluations: they disregard the true distribution of vandalism among all edits, and they have not been double-checked by different annotators. Hence, we have compiled a new, large-scale corpus whose edits were sampled from a week's worth of Wikipedia edit logs. The corpus comprises 32 452 edits on 28 468 different articles. It was annotated by 753 annotators recruited from Amazon's Mechanical Turk, who cast more than 190 000 votes so that each edit has been reviewed by at least three of them. The annotator agreement was analyzed in order to determine whether an edit is regular or vandalism, and 2 391 edits were found to be vandalism.

Detection Performance Measures. A starting point for the quantification of any classifier's performance is its confusion matrix, which contrasts how often its predictions on a test set match the actual classification:

Classifier	**Actual**	
Prediction	P	N
P	TP	FP
N	FN	TN

In the case of vandalism detectors, vandalism is denoted as P and regular edits as N: TP is the number of edits that are correctly identified as vandalism (true positives), and FP is the number of edits that are untruly identified as vandalism (false positives). Likewise, FN and TN count false negatives and true negatives. Important performance measures are computed from this matrix, such as the TP rate, the FP rate, or recall and precision:

$$\text{recall} = TP\text{-rate} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP} \qquad FP\text{-rate} = \frac{FP}{FP + TN}$$

Plotting precision versus recall spans the precision-recall space, and plotting the TP rate versus the FP rate spans the ROC space. The former is used widely in information retrieval as performance visualization, while the latter is used preferably in machine learning. Despite the fact that recall and TP rate are the same, both spaces visualize different performance aspects and they possess unique properties. In Figure 1 the two

Edit	Actual	Detector A
1	regular	0.93
2	vandalism	0.89
3	vandalism	0.86
4	vandalism	0.83
5	regular	0.79
6	regular	0.76
7	vandalism	0.73
8	vandalism	0.69
9	vandalism	0.66
10	vandalism	0.63

Edit	Actual	Detector A
11	vandalism	0.59
12	regular	0.56
13	vandalism	0.53
14	regular	0.49
15	vandalism	0.46
16	regular	0.43
17	regular	0.39
18	regular	0.36
19	regular	0.33
20	regular	0.29

(a)

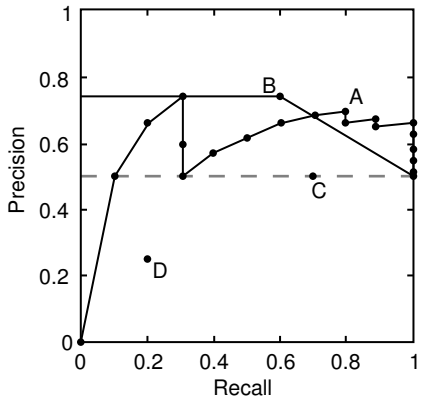
Detector A Prediction	Actual P	Actual N
P	8	3
N	2	7

Detector B Prediction	Actual P	Actual N
P	6	2
N	4	8

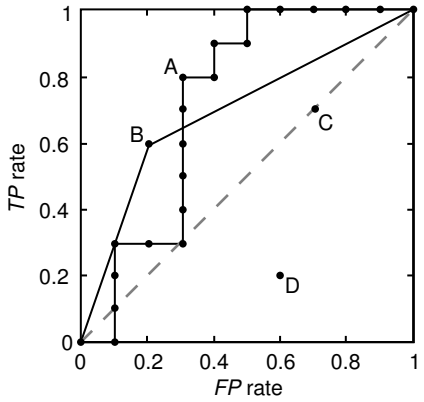
Detector C Prediction	Actual P	Actual N
P	7	7
N	3	3

Detector D Prediction	Actual P	Actual N
P	2	6
N	8	4

(b)



(c)



(d)

Figure 1. (a) A set of test edits, their actual classes, and predictions for them from a vandalism detector A which employs a continuous classifier. (b) Confusion matrices of four vandalism detectors A, B, C, and D. For A, threshold $\tau = 0.58$ is assumed, whereas B, C, and D employ discrete classifiers. (c) Precision-recall space that illustrates the performances of the four detectors. The precision-recall curve for A is given. (d) ROC space that illustrates the performances of the four detectors. The ROC curve of A is given, and for B an ROC curve is induced.

spaces are exemplified. Figure 1a lists 20 test edits along with the fact whether or not they are vandalism. For a vandalism detector A its predictions with regard to the classes of every edit are given. Figure 1b shows the confusion matrix of detector A when τ is set to 0.58 as well as the confusion matrices of three additional detectors B, C, and D. Note that every confusion matrix corresponds to one point in both spaces; Figures 1c and 1d show the precision-recall space and the ROC space:

– *Precision-Recall Space.* The corners of precision-recall space denote extreme cases: at (0,0) none of the edits classified as vandalism are in fact vandalism, at (1,1) all edits classified as vandalism are vandalism; close to (1,0) all edits are classified as vandalism, and close to (0,1) all edits are classified as being regular. Observe that the latter two points are gaps of definition and therefore unreachable in practice: when constructing a test set to approach them, the values of the confusion matrix

become contradictory. The dashed line shows the expected performances of detectors that select classes at random. Note that the classifier characteristics shown in precision-recall space depend on the class distribution in the test set.

– *ROC Space.* The corners of ROC space denote extreme cases: at (0,0) all edits are classified as regular, at (1,1) all edits are classified as vandalism; at (1,0) all edits are classified correctly, at (0,1) all edits are classified incorrectly. The diagonal from (0,0) to (1,1) shows the expected performances of detectors that select classes at random; the ROC space is symmetric about this diagonal by flipping a detector's decisions from vandalism to regular and vice versa. Note that classifier characteristics shown in ROC space are independent of the class distribution in the test set.

Changing the threshold τ of detector A will lead to a new confusion matrix and, consequently, to a new point in precision-recall space and ROC space respectively. By varying τ between 0 and 1 a curve is produced in both spaces, as shown in Figures 1c and 1d. Note that in precision-recall space such curves have sawtooth shape, while in ROC space they are step curves from (0,0) to (1,1). In information retrieval, precision-recall curves are smoothed, which, however, is unnecessary in large-scale classification tasks, since the class imbalance is not as high as in Web search. By measuring the area under a curve, AUC, a single performance value is obtained that is independent of τ. The better a detector performs, the bigger its AUC. Observe that maximizing the ROC-AUC does not necessarily maximize the precision-recall-AUC [4]. For discrete classifiers, such as B, the curves can be induced as shown. The ROC-AUC is the same as the probability that two randomly sampled edits, one being regular and one vandalism, are ranked correctly. Ideally, AUC values are measured more than once for a detector on different pairs of training sets and test sets, so that variance can be measured o determine whether a deviation from the random baseline is in fact significant. Due to the limited size of the available corpus, and the nature of a competition, however, we could not apply this strategy.

From the above it becomes clear that detector A performs best in this example, closely followed by detectors B and D, which perform equally well. Detector C is no better than a random detector that classifies an edit as vandalism with probability 0.7.

2 Survey of Detection Approaches

Out of 9 groups, 5 submitted a report describing their vandalism detector, while 2 sent brief descriptions. This section surveys the detectors in a unified manner. We examine the edit model used, and the machine learning algorithms that have been employed to train the classifiers.

An edit model function α is made up of features that are supposed to indicate vandalism. A well-chosen set of features makes the task to train a classifier that detects vandalism much easier, whereas a not so well-chosen set of features forestalls a better-than-chance detection performance. Hence, feature engineering is crucial to the success of a vandalism detector. Note in this connection that no single feature can be expected to separate regular edits from vandalism perfectly. Instead, a set of features does the

trick, where each feature highlights different aspects of vandalism, and where the subsequently employed machine learning algorithm is left with using the information provided by the feature set to train a classifier.

We organize the features employed by all detectors into two categories: features based on an edit's content (cf. Table 1) and features based on meta information about an edit (cf. Table 2). Each table row describes a particular kind of feature. Moreover,

Table 1. Features based on an edit's textual difference between old and new article revision.

Feature	Description	References
Character-level Features		
Capitalization	Ratio of upper case chars to lower case chars (all chars).	[6, 9]
	Number of capital words.	[12, 14]
Digits	Ratio of digits to all letters.	[9]
Special Chars	Ratio of non-alphanumeric chars to all chars.	[6, 9, 12]
Distribution	Kullback-Leibler divergence of the char distribution from the expectation.	[9]
Diversity	Length of all inserted lines to the (1 / number of different chars).	[9]
Repetition	Number of repeated char sequences.	[5, 6, 12]
	Length of the longest repeated char sequence.	[9]
Compressibility	Compression rate of the edit differences.	[9, 12]
Spacing	Length of the longest char sequence without whitespace.	[9]
Markup	Ratio of new (changed) wikitext chars to all wikitext chars.	[3, 8, 12, 14]
Word-level Features		
Vulgarism	Frequency of vulgar words.	[3, 5, 6, 9, 12, 14]
	Vulgarism impact: ratio of new vulgar words to those present in the article.	[9]
Pronouns	Frequency (impact) of personal pronouns.	[9]
Bias	Frequency (impact) of biased words.	[9]
Sex	Frequency (impact) of sex-related words.	[9]
Contractions	Frequency (impact) of contractions.	[9]
Sentiment	Frequency (impact) of sentiment words.	[5, 12]
Vandal words	Frequency (impact) of the top-k words used by vandals.	[3, 6, 9, 14]
Spam Words	Frequency (impact) of words often used in spam.	[12]
Inserted words	Average term frequency of inserted words.	[9]
Spelling and Grammar Features		
Word Existence	Ratio of words that occur in an English dictionary.	[6]
Spelling	Frequency (impact) of spelling errors.	[5, 9, 12]
Grammar	Number of grammatical errors.	[5]
Edit Size Features		
Revision size	Size difference ratio between the old revision and the new one.	[9, 12, 14]
Distance	Edit distance between the old revision and the new revision.	[1, 5]
Diff size	Number of inserted (deleted, changed) chars (words).	[3, 5, 9, 12]
Edit Type Features		
Edit Type	The edit is an insertion, deletion, modification, or a combination.	[5]
Replacement	The article (a paragraph) is completely replaced, excluding its title.	[14]
Revert	The edit reverts an article back to a previous revision.	[14]
Blanking	Whether the whole article has been deleted.	[3, 12, 14]
Links and Files	Number of added links (files)	[12]

Table 2. Features based on meta information about an edit.

Feature	Description	References
Edit Comment Features		
Existence	A comment was given.	[3, 6]
Length	Length of the comment.	[1, 9, 12, 14]
Revert	Comment indicates the edit is a revert.	[3, 14]
Language	Comment contains vulgarism or wrong capitalization.	[3, 8]
Bot	Comment indicates the edit was made by a bot.	[3]
Edit Time Features		
Edit time	Hour of the day the edit was made.	[1]
Successiveness	Logarithm of the time difference to the previous edit.	[1]
Article Revision History Features		
Revisions	Number of revisions.	[3]
Reverts	Number of reverts.	[3]
Regular	Number of regular edits.	[3]
Vandalism	Number of vandalism edits.	[3]
Editors	Number of reputable editors.	[3]
Article Trustworthiness Features		
Suspect Topic	The article is on the list of often vandalized articles.	[12]
WikiTrust	Values from the WikiTrust trust histogram.	[1]
	Number of words with a certain WikiTrust reputation score.	[1]
Editor Reputation Features		
Anonymous	Anonymous editor.	[1, 5, 6, 8, 9, 12, 14]
Known Editor	Editor is administrator (on the list of reviewers)	[12]
Edits	Number of previous edits by the same editor.	[5, 8, 14]
	Number of previous edits by the same editor on the same article.	[5]
Reputation	Scores that compute a user's reputation based on previous edits.	[8]
Reverts	Number of reverted edits, or participation in edit wars.	[3, 14]
Vandalism	Editor vandalized before.	[14]
Registration	Time the editor was registered with Wikipedia.	[5, 14]

the right table column indicates who employed which feature in their detectors. Note that our descriptions are not as detailed as those of the original authors, and that they have been reformulated where appropriate in order to highlight similar feature ideas.

Content-based features as well as meta information-based features further subdivide into groups of similar kinds. Content-based features on character-level aim at vandalism that sticks out due to unusual typing, whereas features on word-level use dictionaries to quantify the usage of certain word classes and words often used by vandals. Some features even quantify spelling and grammar mistakes. The size of an edit is measured in various ways, and certain edit types are distinguished. The meta information-based features evaluate the comment left by an editor, and the time-related information about an edit. Other features quantify certain characteristics about the edited article in order to better inform the machine learning algorithm about the prevalence of vandalism in an article's history. Moreover, information about an editor's reputation is quantified assuming that reputable editors are less likely to vandalize.

Finally, all groups, who submitted a description of their approach, employed decision trees in their detectors, such as random forests, alternating decision trees, naive Bayes decision trees, and C4.5 decision trees. Two groups additionally employed other classifiers in an ensemble classifier. The winning detector uses a random forest of 1000 trees at 5 random features each.

3 Evaluation Results

In this section we report on the detection performances of the vandalism detectors that took part in PAN. To determine the winning detector, their overall detection performance is measured as AUC in ROC space and precision-recall space. Moreover, the detectors' curves are visualized in both spaces to gain further insight into their performance characteristics. Finally, we train and evaluate a meta detector which combines the predictions made by the individual detectors to determine what performance can be expected from a detector that incorporates all of the aforementioned features. We find that the meta detector outperforms all of the other detectors.

3.1 Overall Detection Performance

Table 3 shows the final ranking among the 9 vandalism detectors according to their area under the ROC curve. Further, each detector's area under the precision-recall curve is given as well as the different ranking suggested by this measure. Both values measure the detection performance of a detector on the 50% portion of the PAN-WVC-10 corpus that was used as test set, which comprises 17 443 edits of which 1481 are vandalism. The winning detector is that of Mola Velasco [9]; it clearly outperforms the other detectors with regard to both measures. The performances of the remaining detectors vary from good to poor performance. As a baseline for comparison, the expected detection performance of a random detector is given.

Table 3. Final ranking of the vandalism detectors that took part in PAN 2010. For simplicity, each detector is referred to by last name of the lead developer. The detectors are ranked by their area under the ROC curve, ROC-AUC. Also, each detector's area under the precision-recall curve, PR-AUC, is given, along with the ranking difference suggested by this measure. The bottom row shows the expected performance of a random detector.

ROC-AUC	ROC rank	PR-AUC	PR rank		Detector
0.92236	1	0.66522	1	–	Mola Velasco [9]
0.90351	2	0.49263	3	↓	Adler et al. [1]
0.89856	3	0.44756	4	↓	Javanmardi [8]
0.89377	4	0.56213	2	↑↑	Chichkov [3]
0.87990	5	0.41365	7	↓↓	Seaward [12]
0.87669	6	0.42203	5	↑	Hegedűs et al. [6]
0.85875	7	0.41498	6	↑	Harpalani et al. [5]
0.84340	8	0.39341	8	–	White and Maessen [14]
0.65404	9	0.12235	9	–	Iftene [7]
0.50000	10	0.08490	10	–	Random Detector

3.2 Visualizing Detection Performance in Precision-Recall Space and ROC Space

Figures 2 and 3 show the precision-recall space and the ROC space, and in each space the respective curves of the vandalism detectors are plotted. Note that all detectors supplied predictions for every edit in the test set, however, some detectors' prediction values are less fine-grained than those of others, which can be also observed by looking at the smoothness of a curve.

In precision-recall space, the detector of Mola Velasco is the only detector that achieves a nearly perfect precision at recall values smaller than 0.2. All other curves have lower precision values to begin with, and they fall off rather quickly with recall increasing to 0.2. An exception is the detector of Chichkov. While the curves of the detectors on ranks 5–8 behave similar at all times, those of the top 4 detectors behave different up to a recall of 0.7, but similar onwards. Here, the detectors of Chichkov and Javanmardi outperform the winning detector to some extent. Altogether, the winning

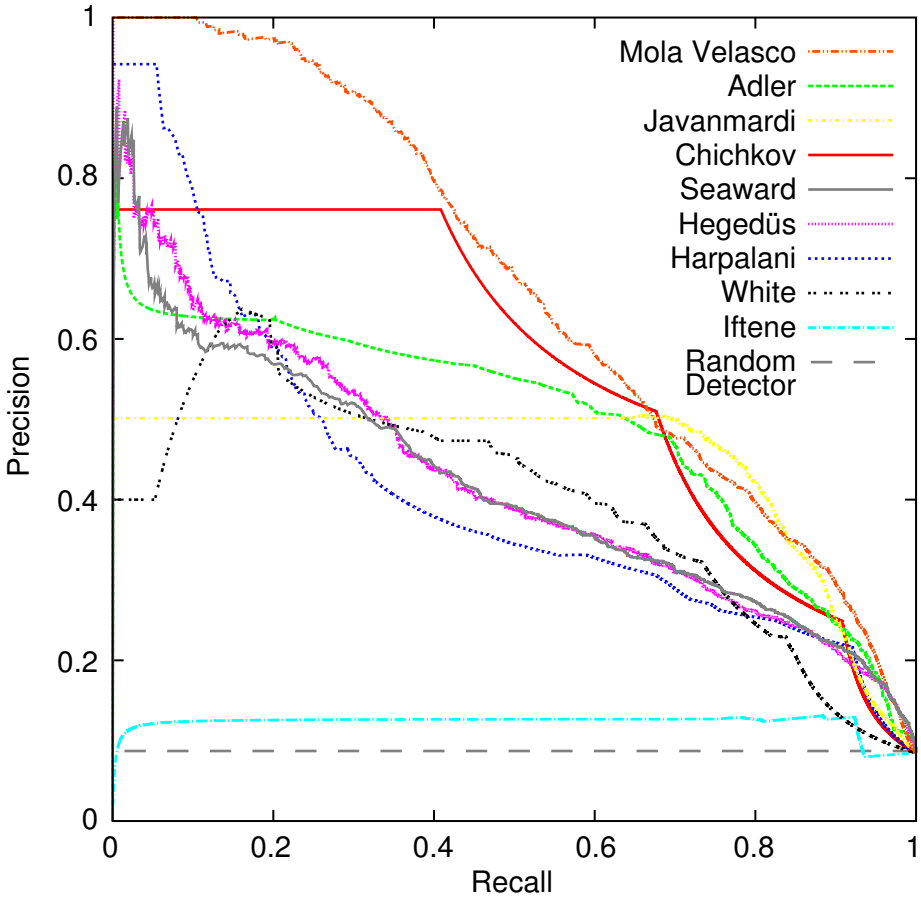


Figure 2. Precision-recall curves of the vandalism detectors developed for PAN. The key is sorted according to the final ranking of the vandalism detectors.

detector clearly outperforms the other detectors by far in precision-recall space, but it does not dominate all of them, which shows possibilities for improvements. Nevertheless, its threshold can be adjusted so that 20% of the vandalism cases will be detected with virtually perfect precision, i.e., it can be used without constant manual double-checking of its decisions. This has serious practical implications and cannot be said of any other detector in the competition.

By contrast, in ROC space, the detectors' curves appear to be much more uniform. Still, some detectors perform worse than others, but differences are less obvious. The top 4 detectors and the detectors on ranks 5–8 behave similar at FP rates below 0.4. The winning detector is outperformed by those of Chichkov and Javanmardi at FP rates between 0.1 and 0.2, as well as those of Adler et al., Hegedűs et al., and Seaward at FP rates above 0.6. Altogether, this visualization supports the winning detector but

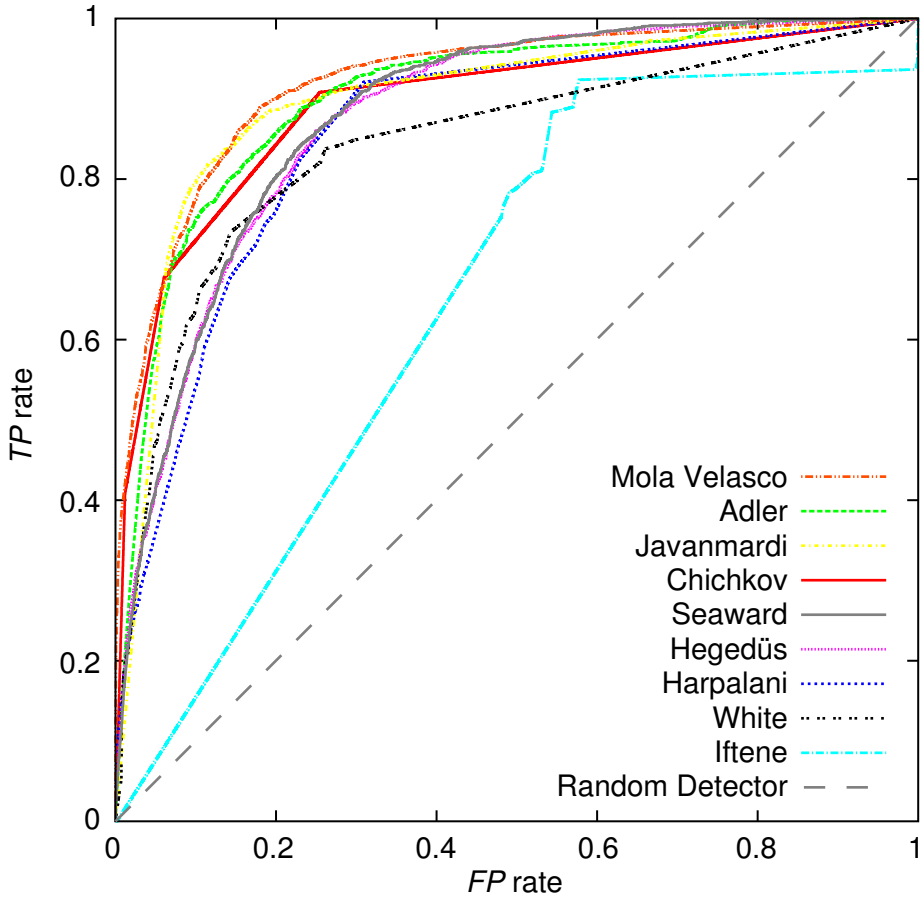


Figure 3. ROC curves of the vandalism detectors developed for PAN. The key is sorted according to the final ranking of the vandalism detectors.

it does not set it apart from the rest, which may lead to the conclusion that the different approaches and feature sets employed are not so different, after all.

Discussion. The differences between precision-recall space and ROC space underline that they indeed possess unique properties, but they also raise the question, who's right? To answer this question for a particular classification task, it has to be determined whether the precision or the FP rate is more important. For vandalism detection, due to the class imbalance between regular edits and vandalism edits, precision may be more important, which questions our decision made before the competition to use the ROC-AUC to rank vandalism detectors.

3.3 Combining all Vandalism Detectors: The PAN'10 Meta Detector

Our evaluation shows that there is definite potential to improve vandalism detectors even further: the winning detector does not dominate all other detectors, and more importantly, no detector uses all features, yet. In what follows, we report on an experiment to determine what the performance of a detector that incorporates all features would be. To this end, we have set up the PAN'10 meta detector that trains a classifier based on the predictions of all vandalism detectors for the set of test edits. The meta detector thus combines the feature information encoded in the detectors' predictions.

Let E_c denote the PAN-WVC-10 corpus of edits whose classification is known, and let C denote the set of detectors developed for PAN, where every $c \in C$ maps an edit model $\alpha_c(e) = \mathbf{e}$, $e \in E_c$, onto $[0, 1]$. E_c was split into a training set $E_{c|\text{train}}$ and a test set $E_{c|\text{test}}$. In the course of the competition, every $c \in C$ was trained based on $E_{c|\text{train}}$ and then used to predict whether or not the edits in $E_{c|\text{test}}$ are vandalism. Instead of analyzing those predictions to determine the performance of the detectors in C—as was done in the previous section—$E_{c|\text{test}}$ is split again into $E_{c|\text{test}|\text{train}}$ and $E_{c|\text{test}|\text{test}}$. The former is used to train our new meta detector c_{PAN}, while the latter is used to test its performance. For c_{PAN} an edit $e \in E_{c|\text{test}}$ is modeled as a vector $\mathbf{e}$ of predictions made by the detectors in C: $\mathbf{e} = (c_1(\alpha_{c_1}(e)), \ldots, c_{|C|}(\alpha_{c_{|C|}}(e)))$ where $c_i \in C$. That way, without re-implementing the detectors, it is possible to test the impact of combining the edit models of all detectors. To train c_{PAN} we employ a random forest of 1000 trees at 4 random features each. $E_{c|\text{test}|\text{train}}$ and $E_{c|\text{test}|\text{test}}$ both comprise 8721 edits of which 713 and 768 are vandalism, respectively.

Table 4. Detection performance of the PAN'10 meta detector and the top 4 detectors in the competition, measured by the areas under the ROC curve and the precision-recall curve, PR-AUC.

ROC-AUC	PR-AUC	Detector
0.95690	0.77609	PAN'10 Meta Detector
0.91580	0.66823	Mola Velasco [9]
0.90244	0.49483	Adler et al. [1]
0.89915	0.45144	Javanmardi [8]
0.89424	0.56951	Chichkov [3]
0.50000	0.08805	Random Detector

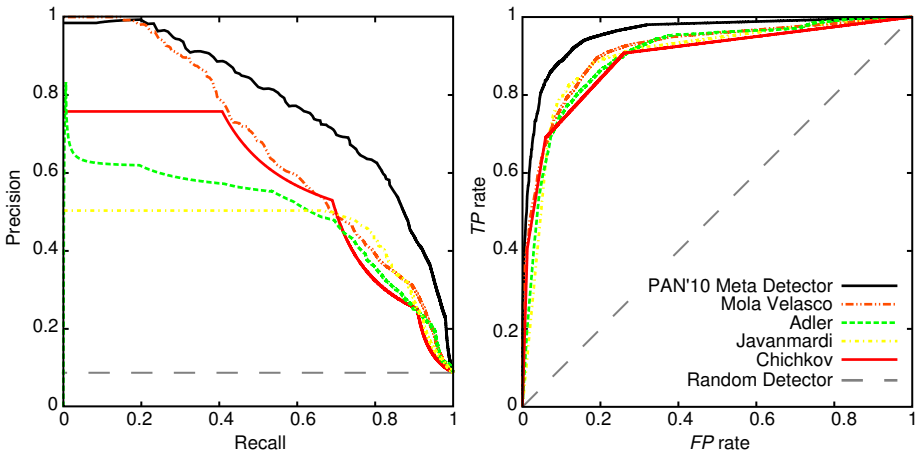


Figure 4. Precision-recall curves and ROC curves of the PAN'10 meta detector and the top 4 vandalism detectors in the competition.

Table 4 contrasts the overall performance of our meta detector with the top 4 vandalism detectors in the competition: the meta detector outperforms the winning detector by 5% ROC-AUC and by 16% PR-AUC. Note that, in order to make a fair comparison, we have recomputed both measures for the top 4 detectors based only on $E_{c|\text{test}|\text{test}}$. Figure 4 visualizes precision-recall space and ROC space for the 5 detectors. In both spaces, the meta detector's curves stick out notably. Observe that, in precision-recall space, the meta detector is still outperformed by the winning detector at recall values below 0.2. While in ROC space, the meta detector's curve lies uniformly above the others, the respective curve in precision-recall space shows that the meta detector gains more performance at recall values above 0.4. This shows that none of the detectors provide the meta detector with additional information to correct errors in high-confidence predictions, whereas, a lot of errors are corrected in low-confidence predictions.

4 Conclusion

In summary, the results of the 1st international competition on vandalism detection are the following: 9 vandalism detectors have been developed, which include a total of 55 features to quantify vandalism characteristics of an edit. One detector achieves outstanding performance which allows for its practical use. Further, all vandalism detectors can be combined into a meta detector that even outperforms the single best performing detector. This shows that there is definite potential to develop better detectors.

Lessons learned from the competition include that the evaluation of vandalism detectors cannot be done solely based on the receiver operating characteristic, ROC, and the area under ROC curves. Instead, an evaluation based on precision and recall provides more insights. Despite the good performances achieved, vandalism detectors still have a long way to go, which pertains particularly to the development of vandalism-indicating features. It is still unclear, which features contribute how much to the detection performance. Finally, the corpora used to evaluate vandalism detectors require

further improvement with regard to annotation errors. Future evaluations of vandalism detectors will have to address these shortcomings.

Bibliography

[1] B. Thomas Adler, Luca de Alfaro, and Ian Pye. Detecting Wikipedia Vandalism using WikiTrust: Lab Report for PAN at CLEF 2010. In Braschler et al. [2]. ISBN 978-88-904810-0-0.

[2] Martin Braschler, Donna Harman, and Emanuele Pianta, editors. *Notebook Papers of CLEF 2010 LABs and Workshops, 22-23 September, Padua, Italy*, 2010. ISBN 978-88-904810-0-0.

[3] Dmitry Chichkov. Submission to the 1st International Competition on Wikipedia Vandalism Detection, 2010. SC Software Inc., USA.

[4] Jesse Davis and Mark Goadrich. The Relationship Between Precision-Recall and ROC curves. In *ICML'06: Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143874.

[5] Manoj Harpalani, Thanadit Phumprao, Megha Bass, Michael Hart, and Rob Johnson. Wiki Vandalysis—Wikipedia Vandalism Analysis: Lab Report for PAN at CLEF 2010. In Braschler et al. [2]. ISBN 978-88-904810-0-0.

[6] István Hegedűs, Róbert Ormándi, Richárd Farkas, and Márk Jelasity. Novel Balanced Feature Representation for Wikipedia Vandalism Detection Task: Lab Report for PAN at CLEF 2010. In Braschler et al. [2]. ISBN 978-88-904810-0-0.

[7] Adrian Iftene. Submission to the 1st International Competition on Wikipedia Vandalism Detection, 2010. From the Universtiy of Iasi, Romania.

[8] Sarah Javanmardi. Submission to the 1st International Competition on Wikipedia Vandalism Detection, 2010. From the Universtiy of California, Irvine, USA.

[9] Santiago M. Mola Velasco. Wikipedia Vandalism Detection Through Machine Learning: Feature Review and New Proposals: Lab Report for PAN at CLEF 2010. In Braschler et al. [2]. ISBN 978-88-904810-0-0.

[10] Martin Potthast. Crowdsourcing a Wikipedia Vandalism Corpus. In Hsin-Hsi Chen, Efthimis N. Efthimiadis, Jaques Savoy, Fabio Crestani, and Stéphane Marchand-Maillet, editors, *33rd Annual International ACM SIGIR Conference*, pages 789–790. ACM, July 2010. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835617.

[11] Martin Potthast, Benno Stein, and Robert Gerling. Automatic Vandalism Detection in Wikipedia. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White, editors, *Advances in Information Retrieval: Proceedings of the 30th European Conference on IR Research (ECIR 2008)*, volume 4956 LNCS of *Lecture Notes in Computer Science*, pages 663–668, Berlin Heidelberg New York, 2008. Springer. ISBN 978-3-540-78645-0. doi: http://dx.doi.org/10.1007/978-3-540-78646-7_75.

[12] Leanne Seaward. Submission to the 1st International Competition on Wikipedia Vandalism Detection, 2010. From the Universtiy of Ottawa, Canada.

[13] Andrew G. West, Sampath Kannan, and Insup Lee. Detecting Wikipedia Vandalism via Spatio-Temporal Analysis of Revision Metadata. In *EUROSEC '10: Proceedings of the Third European Workshop on System Security*, pages 22–28, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0059-9. doi: 10.1145/1752046.1752050.

[14] James White and Rebecca Maessen. ZOT! to Wikipedia Vandalism: Lab Report for PAN at CLEF 2010. In Braschler et al. [2]. ISBN 978-88-904810-0-0.