Scholarly and Research
Communication
VOLUME 8 / ISSUE 1 / 2017

# Automatically Determining Versions of Scholarly Articles

Daniel Rothchild & Stuart Shieber, *Harvard University*

## ABSTRACT

***Background***: Repositories of scholarly articles should provide authoritative information about the materials they distribute and should distribute those materials in keeping with pertinent laws. To do so, it is important to have accurate information about the versions of articles in a collection.

***Analysis:*** This article presents a simple statistical model to classify articles as author manuscripts or versions of record, with parameters trained on a collection of articles that have been hand-annotated for version. The algorithm achieves about 94 percent accuracy on average (cross-validated).

***Conclusion and implications:*** The average pairwise annotator agreement among a group of experts was 94 percent, showing that the method developed in this article displays performance competitive with human experts.

***Keywords:*** Article versions; Document classification; Open access; Tools; Workflow management

CISP Press
Scholarly and Research Communication
Volume 8, Issue 1, Article ID 268, 13 pages
doi: 10.22230/src2017v8n1a268
Journal URL: www.src-online.ca

Rothchild, Daniel, & Shieber, Stuart. (2017). Automatically Determining Versions of Scholarly Articles. *Scholarly and Research Communication*, *8*(1): 268, 13 pp.

## INTRODUCTION



Scholars are increasingly distributing their scholarly articles through means beyond the traditional static curated journal. For example, online open access repositories, such as arXiv.org and the authors' own institutional repository, Digital Access to Scholarship at Harvard (DASH), accept for distribution scholarly articles that authors typically submit to curated journals as well.

It is useful for both the administrators and patrons of these online repositories to know the version status of the articles in their collection – for instance, whether the articles are author manuscripts (here abbreviated as AM), generated and typeset by the authors themselves, or versions of record (VoR), typeset in final form by a publisher. In addition to serving as important metadata in its own right, this information can be useful, for example, when deciding how to distribute the article: if an article is a VoR, it is possible that further distribution restrictions may inhere in the article as compared to an AM.

Of greatest relevance here, many publishing agreements prohibit the public distribution of VoRs, yet authors often distribute their articles in violation of these agreements. A study of self-archiving at Carnegie Mellon University found that, of the articles that faculty made publicly available online, only half were posted in accordance with the publishers' policies (Covey, 2009). Bo-Christer Björk, Mikael Laakso, Patrik Welling, and Patrik Paetau (2014) found that many authors upload publisher-typeset versions of their articles to the Web even though very few publishers allow authors to distribute this version. In order to comply with publisher policies on the distribution of articles, institutional repositories such as DASH need to reliably determine the version of articles submitted for distribution.

Officials at Harvard's Office for Scholarly Communication (OSC), who administer DASH, currently determine by hand the version of each article that they add to the repository – a time-consuming process. Based on estimates from three officials at the OSC, it takes on average 14 seconds per article to determine an article's version. In other words, some 120 hours would be spent to hand-annotate each article in a repository of 30,000 articles, which is approximately the current size of DASH.

While the OSC uses a variety of version designations – roughly corresponding to guidelines published by the National Information Standards Organization (2008) – we use the term "Author's Manuscript" to refer to those articles marked as "Author's Original" or "Accepted Manuscript" and the term "Version of Record" to refer only to those articles marked as "Version of Record." (We did not consider the tiny minority of articles in DASH with a version designation other than these three.)

To streamline this process of version determination, we developed simple statistical models to perform the classification of articles in PDF format (the format of the vast majority of distributed articles), with parameters trained on a collection of articles that have been hand-annotated for version. Our algorithm achieves about

94 percent accuracy on average and about 98 percent average accuracy on the three-fourths of articles for which its prediction is most confident. On held-out test data, our method achieved 92 percent average accuracy. In comparison, the average pairwise annotator agreement among a group of experts was 94 percent, showing that our method displays performance competitive with human experts.

In this article, we describe our models, including the article set that we used for training, validation, and testing; the features of the PDFs that informed the model; and the results of our testing of the model alone and in comparison to human performance on the same task.

This article is a contribution to the extensive literature on document classification, which is surveyed by Nawei Chen and Dorothea Blostein (2007), especially efforts (such as ours) that make use of image features for categorizing documents. The types of features we use here are by now standard. For instance, Christian Shin, David Doermann, and Azriel Rosenfeld (2001) use features similar to our appearance features, including many that are considerably more sophisticated, to label documents in a variety of classification tasks, and Charles Smutz and Angelos Stavrou (2012) use PDF metadata (as does our method) to classify PDF documents as malicious or benign. However, we know of no prior work applying these techniques to the problem of version classification.

## TRAINING AND TEST DATA

Training a model to perform version classification requires a corpus of scholarly articles hand-labelled for the classification of interest. For this purpose, we made use of the tens of thousands of articles in the DASH repository, spanning many disciplines in the natural sciences, social sciences, and humanities. Starting with a snapshot of 28,012 documents in the repository as of June 2015, we excluded documents under the following conditions:[1]

- There were multiple documents associated with the same DASH entry (in which case we ignored all documents except the one labelled as the primary PDF): 217 eliminated.
- The article PDF caused our PDF parsing tools to crash or hang: 13 eliminated.
- The article had a version other than "Version of Record," "Author's Original," or "Accepted Manuscript" in DASH: 291 eliminated.
- The DASH repository had no hand-labelled version attached to the article:[2] 4,396 eliminated.

The remaining data set consists of 23,095 articles.

We held out a balanced sample of 2,000 articles consisting of 1,000 AMs and 1,000 VoRs, leaving 21,095 PDFs as a training set on which we report unbiased results using cross-validation.

## METHODS

We use machine learning algorithms to learn parameters for a logistic regression model to predict whether previously unseen articles are AMs or VoRs. Logistic regression is a simple statistical model that can be used to predict binary labels based on continuous independent variables. In machine learning, these independent variables are called "features" and the binary labels, "class labels." For the present case, the two classes are AM and VoR, and the features we used are described in the sections below.[3] We found that other algorithms yield similar results, as is typical for many machine learning tasks where the choice of features dominates the choice of algorithm in determining accuracy.[4] We pursued logistic regression for the final model because of its conceptual simplicity.

We use three types of features to feed into the logistic regression model, each of which is described in more detail below:

**Document information dictionary features**. The frequency of words appearing in various fields in the PDF's document information dictionary – a key-value store associated with the entire PDF.

**Word features**. Frequencies of a set of automatically discovered words found in the text of the articles themselves that are most indicative of version.

**Appearance features**. Various image properties of a very low-resolution rendering of the first page of each article, which capture aspects of the appearance of the page.

During development of the model, we used a standard technique called $n$-fold cross-validation: the training data is divided into $n$ equal-sized partitions, and the model is trained $n$ times, each time on a different set of $n - 1$ partitions of the training data, testing on the remaining partition to measure the model's accuracy after each training run. Cross-validation allows us to obtain an average and variance for the accuracy every time we train the model.

Because logistic regression provides not only a classification of the input PDF but also a confidence in that classification, we can examine accuracy for subsets of the test data for which the classification is of a certain confidence. In effect, the model is allowed to abstain from guessing on any given PDF if it is not sufficiently confident in the class label it was going to assign. By doing so, we allow the model to obtain higher accuracies while still making a prediction for most of the supplied PDFs.

In the case at hand, we used logistic regression with 5-fold cross-validation. Although the full set of 23,095 articles that we used from DASH was comprised of about three-quarters VoR and one-quarter AM, we balanced the class sizes during training using a directive[5] to the learning software and during testing by calculating accuracies in each class separately and averaging the accuracies together. Doing

so eliminates the bias introduced by unequal class sizes in the data set, so that an algorithm that guesses the most common class on all articles would get an accuracy of about 50 percent instead of the most common class's frequency in the data set, which in this case is about 75 percent.

We turn now to the features of articles that serve as the input to the classification method.

**Document information dictionary features**. The PDF file format allows documents to include a document information dictionary – a key-value store associated with the entire PDF containing fields such as "Title," "Author," and "Subject." Not every PDF has the same set of keys (fields) and the values of each field also vary across the PDFs, but the values of certain important fields can be useful in classifying articles. We consider every field that appears in at least one percent of the training PDFs, and for each such field, we consider the 100 most common "words"[6] that appear in the values across all the PDFs. For each of these field-word pairs, we include a feature that is the frequency of that word in the field. We also include a feature that takes the value 1 when the field is not present in the PDF document information dictionary and 0 otherwise.

Certain document information dictionary fields and words proved especially useful for distinguishing between AMs and VoRs. For example:

- The word "publisher" appears in the "Creator" field of 3,628 VoRs and of only 15 AMs (for example, the "Creator" field of one VoR is "Arbortext Advanced Print Publisher 9.0.226/W").
- The word "InDesign" appears in the "Creator" field of 1,330 VoRs and of only 11 AMs.

Words such as these, characteristic of one class or the other, can be found for other document information dictionary fields as well. These features are determined automatically by training on the training set of documents, and require no hand selection by experts. Among other advantages, the automatic process means that if document information dictionary values change over time, retraining will find new indicative features.

**Top word features.** Although the main text of an article does not usually change substantially between the author's manuscript and the publisher's version, there are certain key words that, if present on the first page of a scholarly article, often imply a particular class label. To automatically find these words, we calculate a separation score for each commonly occurring word that appears on the first page of any PDF. For each word $i$, let the mean and standard deviation of the frequency that word $i$ appeared in AMs and VoRs respectively be denoted by $\mu^i_{AM}$, $\mu^i_{VoR}$, and $\sigma^i_{AM}$, $\sigma^i_{VoR}$. Then define a separation score for the $i^{\text{th}}$ word as:

$$S_i = \left( \frac{\mu^i_{AM} - \mu^i_{VoR}}{\sigma^i_{AM} + \sigma^i_{VoR}} \right)^2$$

We create features for the frequency of each of the 500 words with the highest separation score. Crucially, we did not choose any of these words by hand. Rather, we calculated the separation score for every word that appeared a total of at least ten times across all PDFs, and we created features for the 500 highest-scoring words.

Table 1 provides a sample of the words indicative of each class with the highest separation scores. Note that some of the top words are highly specific to DASH. For example, "02138" is the zip code of Harvard University, and "hks" is an abbreviation for "Harvard Kennedy School." It is therefore advantageous for institutions planning to use our model on their repositories to retrain the model using hand-labelled data from their repository instead of using our pre-trained models directly. See the Appendix for a list of the 100 highest-scoring words.

**Table 1: Words with the highest separation scores**

| *Top VoR Word* | *Separation Score* | *Top AM Word* | *Separation Score* |
|---|---|---|---|
| article | 0.521 | 02138 | 0.0852 |
| creative | 0.433 | paper | 0.0366 |
| license | 0.415 | working | 0.0362 |
| commons | 0.414 | papers | 0.0296 |
| attribution | 0.389 | hks | 0.0294 |

*Note*: On the left are words that appear more often in VoRs than in AMs. On the right are words that appear more often in AMs than in VoRs.

Also of note is that the separation scores for the top VoR words are much higher than the separation scores for the top AM words. This suggests that publishers tend to add words to manuscripts during the typesetting process rather than removing words. The words themselves also suggest this, as most of them are typical of copyright notices, which we would expect to appear rarely in AMs.

**Appearance features**. Perhaps the most obvious difference between AMs and VoRs is their typical layout. Although there is no standardized layout that is conserved across all AMs or VoRs, VoRs tend to contain visual cues such as publisher logos, shaded boxes, multi-column layouts, and horizontal and vertical rules. In order to quantify these visual cues, we render the first page of each PDF (where the layout difference is most obvious) as a very low resolution (20 pixel per inch) image. We then create several features that capture some of the differences between the two classes. These features are as follows:

1. Average grayscale pixel value. Following standard conventions, values range from 0 (black) to 255 (white), with lower values corresponding to darker pixels.
2. Median grayscale pixel value.
3. Proportion of grayscale pixels above (that is, lighter than) certain cut-off values (5, 100, 200, 250, and 254).
4. Proportion of the pixels in the image that are coloured (that is, do not have the same values for the red, green, and blue channels).
5. Longest contiguous sequence of (grayscale) pixels below a cut-off of 200 in the

horizontal and vertical directions (as a percent of the total width and height respectively of the image). This feature allows the detection of horizontal and vertical rules.

6. Longest contiguous streak of vertical (grayscale) pixels above a cut-off of 200 in the centre of the image (as a percent of the total height of the image). This feature allows the detection of a two-column layout.

The various cut-off values used in these features were chosen somewhat arbitrarily. However, where alternatives were explored, they did not yield significant improvements to the classification accuracy.

Figures 1 and 2 present sample VoR and AM images. Notice that the two VoRs shown have lower (darker) average pixel values, lower cut-off values, higher colour fractions, higher longest dark rows, and higher longest central light columns.

**Figure 1: Sample low-resolution images of the first pages of two VoRs**

(A)

| | |
|---|---|
| **Median pixel value:** | 255 |
| **Average pixel value:** | 223.01 |
| **Color fraction:** | 0.0020023 |
| **Pixels ≥ 100:** | 0.91861 |
| **Pixels ≥ 254:** | 0.65128 |
| **Longest dark col:** | 0.072464 |
| **Longest dark row:** | 0.85526 |
| **Longest central light col:** | 0.20290 |

(B)

| | |
|---|---|
| **Median pixel value:** | 254 |
| **Average pixel value:** | 205.07 |
| **Color fraction:** | 0.0022193 |
| **Pixels ≥ 100:** | 0.86987 |
| **Pixels ≥ 254:** | 0.49422 |
| **Longest dark col:** | 0.036364 |
| **Longest dark row:** | 0.83529 |
| **Longest central light col:** | 0.66818 |

**Figure 2: Sample low-resolution images of the first pages of two AMs**

(A)

| | |
|---|---|
| **Median pixel value:** | 255 |
| **Average pixel value:** | 247.40 |
| **Color fraction:** | 0 |
| **Pixels ≥ 100:** | 0.99989 |
| **Pixels ≥ 254:** | 0.85348 |
| **Longest dark col:** | 0.013636 |
| **Longest dark row:** | 0.13529 |
| **Longest central light col:** | 0.18636 |

(B)

| | |
|---|---|
| **Median pixel value:** | 255 |
| **Average pixel value:** | 246.27 |
| **Color fraction:** | 0 |
| **Pixels ≥ 100:** | 0.99877 |
| **Pixels ≥ 254:** | 0.83639 |
| **Longest dark col:** | 0.013636 |
| **Longest dark row:** | 0.23529 |
| **Longest central light col:** | 0.26818 |

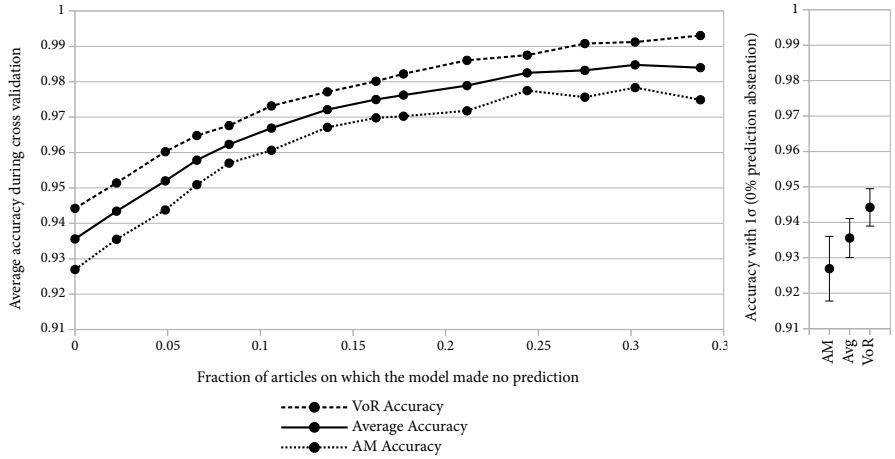
## RESULTS

**Inter-annotator agreement**. In order to provide a benchmark against which the model's performance can be compared, we enlisted three expert annotators at Harvard's Office for Scholarly Communication to hand-label a balanced random sample of 200 PDFs from the DASH repository (100 each of AMs and VoRs). All three annotators agreed on the classification of 91 percent of the articles in the sample. Inter-annotator agreement as measured by Fleiss' Kappa was 0.88, and the average pairwise annotator agreement was 94 percent. These measurements provide a benchmark of expert human performance for the task; we would expect accuracies of 94 percent to represent expert human performance.

The accuracy of our model on the test set of 2,000 articles is 92.3 percent. The average accuracy on the training set using the cross-validation method is 93.5 ± 0.4 percent. This accuracy suggests that our model is achieving expert human performance. We would not expect any model to perform significantly better than the 94 percent pairwise annotator agreement described above, because the article version information that we used was itself generated by human experts at the OSC, so we would expect our training data to have an error rate in the version labels of around six percent. A model that achieves performance significantly higher than 94 percent would most likely be overfitting the data.

We can also allow the classifier to abstain from giving a prediction if it is not confident in its prediction. For varying rates of prediction abstention, we achieve higher overall accuracies, as shown in Figure 3. For instance, our method achieves about 98 percent accuracy on the three-fourths of data for which it is most confident.

**Figure 3**




*Left*: Model accuracy classifying AMs and VoRs for varying rates of prediction abstention. The average of the two accuracies is also shown. The points are not evenly spaced along the horizontal axis since we cannot choose the prediction abstention rate directly: the model outputs the confidence it has in its prediction as a number between 0 and 1 and we arbitrarily threshold that confidence to yield different prediction abstention rates. *Right*: Model accuracy classifying AMs and VoRs when the model was asked to predict every article. Error bars show one standard deviation. (Error bars for accuracies where the model was allowed to abstain from predicting are qualitatively similar.)

To determine the predictive power of each type of feature, we present the performance of the model for each feature class individually, as well as the two most predictive feature classes together, namely the document information dictionary and appearance features. The resulting cross-validated accuracies are shown in Table 2.

**Table 2. Cross-validated accuracy when the model was run using various subsets of the features**

| *Features Used* | *Accuracy ± one std. dev.* |
|---|---|
| Top word only | 86.5 ± 0.5% |
| Appearance only | 87.4 ± 0.6% |
| Doc info only | 91.4 ± 0.3% |
| Doc info and appearance | 93.4 ± 0.2% |
| All three | 93.5 ± 0.4% |

Unsurprisingly, our model tends to have the most trouble with PDFs that were typeset by a publisher but are designated as AMs. (Such examples exist, as publishers sometimes provide typeset articles explicitly labelled by the publisher as author manuscripts as a courtesy to authors.) These PDFs are especially difficult to classify because their document information dictionary and image features make them look like VoRs but they are technically AMs.

A natural question is whether the model's errors occur on articles that human experts also find difficult. To gain insight into the question, we trained the model on all the training data except for the 200 articles that were used for the inter-annotator agreement study. We used the model thus trained to predict the classification of those 200 articles, in order to see whether the model is more likely to make errors on PDFs whose versions the expert annotators disagreed upon. The model misclassified 13 out of the 200 PDFs (93.5% accuracy). Of these misclassifications, two (15%) were PDFs whose versions the annotators disagreed upon. If the misclassifications were distributed independently of the annotator disagreements, we would expect on average that 1.2 of the misclassifications would coincide with the annotator disagreements. There is thus no evidence that the model fails on the same kinds of articles that the human experts find difficult. That the model is making mistakes on articles that the annotators all agree on suggests there may still be room for improvement in the model's performance.

## DISCUSSION

Our trained model does well enough compared to human annotators that we believe the task of classifying PDFs as AMs or VoRs can be completely automated with little loss of accuracy, and that substantial efficiencies could be gained by deploying the method on the large fraction of confidently labelled articles, deferring the others for a quick human verification.

Nonetheless, there are several potential avenues for improvement. Currently our model can only handle scholarly articles in PDF format. We found that limiting ourselves to PDF documents was not overly restrictive given our data set, but extending the model to accept other file types might be desirable. It would be simple to extend the word features and appearance features to other document types (such as LATEX and Microsoft Word), but there may not be an analogue of

the document information dictionary in these formats that provides as much useful information for version classification. There may be useful information contained in the articles that we currently do not extract as features (such as page size or other PDF XMP metadata streams). Creating more features might help address this problem. Our method of deciding on features to use is ad hoc (for example the grayscale thresholds, the horizontal/vertical rule detection parameters, the resolution of the first page images we extract, and the prevalence cut-offs for document information dictionary fields). Where alternatives were explored, we found no significant increase in performance, but our model might be improved by automating our feature selection (for example by using a neural network). Finally, it might be useful to explore more fine-grained version classification, for instance by differentiating AMs between accepted author manuscripts, NISO's AM, and earlier versions, NISO's "Author Original" (AO) and "Submitted Manuscript Under Review" (SMUR).

There are many ways the model might be used in practice. Repository administrators could use the model to completely automate the version classification process. Alternatively, the model could be used only when it is very confident in its classification, with the remaining articles labelled by hand. The model could also be used during the article uploading process to warn uploaders if the repository is soliciting a different version from the one being uploaded, or to reject such uploads outright. Lastly, the model could be used to make suggestions when a human annotator is determining an article's version by hand.

## ACKNOWLEDGEMENTS

This work was made possible in part by a grant from the Arcadia Fund to the Harvard Library. We thank Harvard's Office for Scholarly Communication for aid that made this study possible, especially William McKinney for facilitating the data access, Ben Steinberg, Rebecca Cremona, and Colin Lukens for their help in the annotation experiments, and Peter Suber for additional comments and insights.

## NOTES

1. Our PDF parsing tools were unable to parse some of the PDFs in DASH. In particular, we were unable to extract images of the first pages of 57 articles, the plaintext of the first pages of 144 articles, and the PDF document information dictionary from 472 articles. We set the value of features we were unable to extract to zero (or to one for the binary "document information dictionary field not present" feature) during training and testing.

2. Many articles in DASH do not have a hand-labelled version because the OSC did not originally include a "version" field in the DASH metadata. Indeed, one of the prime applications of the model we provide is to help retrospectively label these currently unlabelled articles.

3. We train our models using the Python package `scikit-learn` (Buitinck, Louppe, Blondel, Pedregosa, Mueller, Grisel, Niculae, Prettenhofer, Gramfort, Grobler, Layton, Vanderplas, Joly, Holt, & Varoquaux, 2013), in particular, the liblinear solver (Machine Learning Group, n.d.) with L2 regularization. Our code is available online (Zenodo, 2017).

4. By way of comparison, a support vector machine classifier yielded an accuracy of 91 percent; a decision tree classifier yielded an accuracy of 91 percent; and a random forest classifier yielded an accuracy of 93 percent. Compare with the 93.5 percent accuracy of the logistic regression model as noted in Table 2.

5. We set the `scikit-learn` parameter `class_weight` to `balanced`, which weights the training instances from the less common class more heavily, thereby compensating for the unbalanced training set.

6. When dividing text into words, we converted the text to lower case, removed all characters besides letters, numbers, and underscores, and then created word boundaries at every sequence of white-space characters.

## WEBSITES

arXiv, http://arXiv.org/
Digital Access to Scholarship at Harvard, http://dash.harvard.edu/

## REFERENCES

Björk, Bo-Christer, Laakso, Mikael, Welling, Patrik, & Paetau, Patrik. (2014). Anatomy of green open access. *Journal of the Association for Information Science and Technology*, *65*(2), 237-250. URL: http://dx.doi.org/10.1002/asi.22963 [1/28/2017].

Buitinck, Lars, Louppe, Gilles, Blondel, Mathieu, Pedregosa, Fabian, Mueller, Andreas, Grisel, Olivier, Niculae, Vlad, Prettenhofer, Peter, Gramfort, Alexandre, Grobler, Jaques, Layton, Robert, Vanderplas, Jake, Joly, Arnaud, Holt, Brian, & Varoquaux, Gaël. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD workshop: Languages for data mining and machine learning* (pp. 108-122).

Chen, Nawei, & Blostein, Dorothea. (2007). A survey of document image classification: problem statement, classifier architecture and performance evaluation. *International Journal of Document Analysis and Recognition (IJDAR)*, *10*(1), 1-16. URL: http://dx.doi.org/10.1007/s10032-006-0020-2 [1/28/2017].

Covey, Denise Troll (2009). Self-archiving journal articles: A case study of faculty practice and missed opportunity. *portal: Libraries and the Academy*, *9*(2), 223-251. URL: https://muse.jhu.edu/article/262847 [1/28/2017].

Machine Learning Group. (n.d.). *LIBLINEAR—A library for large linear classification*. Taipei City, TW: National Taiwan University. URL: http://www.csie.ntu.edu.tw/~cjlin/liblinear/ [1/28/2017].

NISO/ALPSP Journal Article Versions (JAV) Technical Working Group. (2008, April). *Journal Article Versions (JAV): Recommendations of the NISO/ALPSP JAV Technical*

*Working Group* (Tech. Rep.). URL: http://www.niso.org/publications/rp/RP-8-2008.pdf [1/28/2017].

Shin, Christian, Doermann, David, & Rosenfeld, Azriel. (2001). Classification of document pages using structure-based features. *International Journal on Document Analysis and Recognition*, *3*(4), 232-247. URL: http://dx.doi.org/10.1007/PL00013566 [1/28/2017].

Smutz, Charles, & Stavrou, Angelos. (2012). Malicious PDF detection using metadata and structural features. In *Proceedings of the 28th annual computer security applications conference* (pp. 239-248). New York, NY: ACM. URL: http://dl.acm.org/citation.-cfm?id=2420987 [1/28/2017].

Zenodo. (2017, January 7). *dhroth/article_classification: Publication version*. URL: https://zenodo.org/record/232938#.WIbsTIWcAyl [1/28/2017].

## AUTHORS

**Daniel Rothchild** is an undergraduate at Harvard University studying physics and computer science. Email: drothchild@college.harvard.edu .

**Stuart Shieber** is James O. Welch, Jr. and Virginia B. Welch Professor of Computer Science and faculty director of the Office for Scholarly Communication at Harvard University. Email: shieber@seas.harvard.edu .

## APPENDIX: TOP 100 WORDS

A list of the 100 words with the highest calculated separation scores follows. The words are ordered by separation scores top to bottom and left to right. Note that we did not choose these words by hand. Rather, we calculated separation scores for every word that appeared at least ten times in any article and then chose the words with the highest scores. This method finds expected words such as publisher names and URLs, but also includes other words that one might not initially think of as having high discriminatory power. Every word in this list except "02138," which is Harvard University's zip code, appears more often in VoRs than in AMs. (The word marked with † is "httpcreativecommonsorglicensesby20".)

| | | | |
|---|---|---|---|
| article | cited | clinical | been |
| creative | licensee | health | 2013 |
| license | boston | interests | available |
| commons | were | declared | issue |
| attribution | distributed | competing | increased |
| permits | funders | no | analysis |
| reproduction | biomed | by | may |
| unrestricted | study | studies | collection |
| original | publish | with | results |
| received | was | access | for |
| and | hospital | source | including |
| published | any | school | central |
| accepted | medicine | 2014 | research |
| of | role | 02138 | or |
| medium | al | associated | 2012 |
| provided | et | have | diseases |
| citation | united | doi | is |
| credited | open | the | background |
| properly | funding | preparation | conclusions |
| medical | ltd | to | center |
| in | editor | had | author |
| under | disease | massachusetts | cells |
| openaccess | wwwplosoneorg | this | |
| plos | http...† | from | |
| distribution | use | correspondence | |
| terms | america | patients | |