Evaluation of classification models for retrieving experimental sections from full-text publications

A. Lefebvre

J. Berendsen

M.R. Spruit

Technical Report UU-CS-2019-002
March 2019

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

ISSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

Evaluation of classification models for retrieving experimental sections from full-text publications

Armel Lefebvre[1], Jorrit Berendsen[1] and Marco Spruit[1]
[1]*Department of Information and Computing Science, Utrecht University, Princentonplein 5, Utrecht, The Netherlands*
a.e.j.lefebvre@uu.nl, j.h.berendsen@students.uu.nl, m.r.spruit@uu.nl

Keywords: Text Mining, Classification, Reproducible Experiments, Performance

Abstract: In recent years, reporting scientific experiments became a challenge for scientists working data-intensive research fields. One of these challenges is to accurately report experimental work relying on computational activities. In this report, an exploratory computational experiment is conducted. We evaluate the performance of a set of classification models to extract experimental paragraphs from full-text scientific publications in an unsupervised fashion. The results show that the best performing classification model (Multinomial Naive Bayes) trained on 30 publications in the Proteomics domain achieves a Recall of 87.12% and an Accuracy of 80.63%. Successful unsupervised extraction of experimental paragraphs from reports can considerably reduce the noise present in full-text publications. This approach could be beneficial to automatically generate domain specific vocabulary describing experimental designs and experimental processes. As such, this work contributes to the identification of NLP techniques automatizing the extraction of domain-specific paragraphs which relate to experimental work.

1 INTRODUCTION

Since the early 2000s concrete steps have been taken towards making scientific knowledge more accessible to scientists, the industry and citizens. There are two main driving forces that led to these circumstances. One driving force is open access (OA), the other one is reproducible research (RR). OA started originally with the Budapest initiative (Chan et al., 2002). The signatories proposed to broaden the access to peer-reviewed journal articles to a wider audience than scholars. After the Budapest initiative, the scope of OA was extended in the Bethesda statement (Suber et al., 2003) and Berlin declaration (Braarvig et al., 2003). After this point, the initial scope of OA was enlarged to supplemental material which was redefined in the Berlin declaration as being original scientific research results, raw data and meta-data, source materials, digital representations of pictorial and graphical materials and scholarly multimedia material (Braarvig et al., 2003).

The other driving force, Reproducible Research (RR) (McNutt, 2014; Peng et al., 2006) started with some serious concerns about the increased amount of errors or misinterpretations of statistical analyses and reproduction failures of peer-reviewed academic work (Donoho, 2010). These concerns led several academics and publishers to request a better scrutiny of reported results or to promote replication studies in different fields, including information systems (Casadevall and Fang, 2010; Fanelli, 2013; Laine et al., 2007; Nekrutenko and Taylor, 2012). RR attempts to deal with apparent mistakes in data analyses by recommending researchers to hand over enough content next to the published paper to facilitate verification of the results by independent reviewers. Therefore, the software and the parameters applied by data processing algorithms are expected to be part of the supplemental material and preserved accordingly.

As an example of public concern, a New York Times article from 2006 (Bosman, 2006) explains that news organizations are becoming more skeptical about research's results due to the increasing discovery of fraudulent papers. The reasons for this increase varies, as it could be due to an increase in the amount of published fraudulent papers, or due to an increase in the efficiency of detecting fraudulent papers (Fanelli, 2013; Fang et al., 2012; Steen et al., 2013). In both cases, skepticism of research increases, and should be tackled by improving reproducibility in order to present trustworthy results to the public (Laine et al., 2007).

One of the challenges in reproducing research, is the quality of the research report (Nicholls et al., 2016). Insufficiently reported research experiments can lead to unsuccessful attempts of reproducing research results, which is both a waste of money and time. For instance, failed attempts to reproduce some of the top cancer studies are due to insufficient details in the reports about experimental resources used (Begley and Ellis, 2012). To improve the quality of research experiment reporting, new conventions and guidelines are proposed (Bandrowski and Martone, 2016). Nevertheless, the analysis of experimental reports can be tedious and steps towards automatic analysis of experimental reports are welcome. Here, we propose a first step to automatically identify paragraphs that are related to experimental designs in published articles. Our goal here is to avoid to be dependent on specific templates used in publications and reduce noise from full-text publications given as input to Natural Language Processing (NLP) algorithms.

We apply NLP techniques as NLP became more mature in the last couple of years and research on this topic keeps increasing within the Information Systems field (Liu et al., 2017; Debortoli et al., 2016). Previous research has already proven that applying NLP technologies to unstructured data can result in prediction Accuracy rates of over 90% (Chokshi et al., 2017; Lacey et al., 2017a; Lacey et al., 2017b). Nevertheless, to be able to transform an experimental report in a standardized format, domain knowledge is assumed to be necessary. Named Entity Recognition analysis can be used for such a task, e.g. to extract resources from the experiment like used software or laboratory instruments, but its effectiveness correlates negatively with the noisiness of the text (Derczynski et al., 2015). Therefore, this research focuses on reducing the noise in the data by selecting experimental paragraphs and dismiss other type of paragraphs found in scientific articles (e.g. references, results, etc.). This is necessary to avoid further analyses on the content of these paragraphs to be biased by sections not containing specific experimental work.

This technical report will first elaborate the related NLP background in Section 2. Thereafter, the experiment itself will be described in section 3. The results from the experiment are presented in Section 4 and discussed in Section 5. Finally, we conclude in Section 6 that our approach considerably reduces noise in Full-Text publication but is not yet sufficient to extract a list of named entities which are useful to build a vocabulary of scientific experimentation.

2 Background: NLP for Paragraph Extraction

To be able to extract the paragraphs in a paper which contain (parts of) the experiment, an algorithm is required to classify the paragraphs from a paper into experimental and non-experimental paragraphs. To identify relevant classification algorithms, previous literature is consulted. Aggarwal and Zhai (Aggarwal and Zhai, 2012) published a survey of text classification algorithms. The authors state that before any classification task feature selection and document representation are crucial. Two representations are explained, the common bag-of-words representation and the strings representation. A bag-of-word represents the text as a multiset of tokens, including the frequency of each tokens. This is in contrast with the text-as-strings representation representing text as characters strings, keeping the sequence of tokens unaltered.

Next, Aggarwal and Zhai introduce two feature selections methods: stop-word removal and stemming (Aggarwal and Zhai, 2012). By stemming words, the words are reduced to their ”stem”. Another variant of stemming is lemmatizing, which also takes the context and Part-Of-Speech into account when reducing a word to its ”lemma” (Jivani et al., 2011). A disadvantage of stemming is that the generated words are not always correct, i.e. stemming the word ”cycling” to ”cycl”. This problem is addressed by lemmatizing, as long as there is an exhaustive lexicon covering domain-specific words, as only words within the lexicon can be lemmatized properly. Multiple studies applying these techniques on different languages proved that lemmatizing can improve performance compared to stemming (Ingason et al., 2008; Korenius et al., 2004).

While Aggarwal and Zhai (Aggarwal and Zhai, 2012) describe several classification algorithms including Decision Trees, Pattern Rule-based Classifiers, Linear Classifiers, and Probabilistic Classifiers, older studies have shown that Support Vector Machine (SVM) classifiers, and, thereafter, Logistic Regression (LR) and Naive Bayes classifiers (NB) perform better than the other classifiers (Colas and Brazdil, 2006; Genkin et al., 2007; Manevitz and Yousef, 2001). At the same time, these results appear to be unstable, the Accuracy of SVM classifiers were dependent on the data set: e.g., SVM outperformed CNN on one of these datasets.

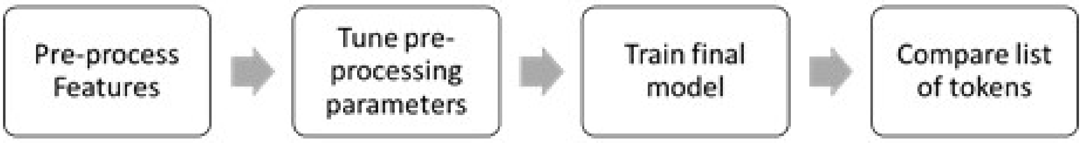

Figure 1: The steps we followed to select the most optimal classifier for extracting experimental paragraphs

3 Experimental Method

As this is an exploratory research, this experiment evaluates standard classification methods for extracting experimental paragraphs. Fig. 1 depicts the main step of the analysis workflow. In short, all documents are transformed into a bag-of-words representation. A bag-of-words is a simple structure which is not requiring intense preprocessing of the documents. In short, the experiment reported here consists of four parts: (1) testing the feature selection parameter values, (2) testing the classification models, (3) train and test the best performing model using the most optimal values for the feature selection parameters and the best performing classification model, and (4) testing the effectiveness of the classification in terms of noise filtered out. In this technical report we give an example of the reduction obtained on one paper using the best performing classification model.

The main goal of developing a classification model is to filter out non-relevant paragraphs from a document, which decreases the corpus from which the experimental steps and used resources of a paper can be extracted. Therefore, the performance of the different feature selection methods and classification models are not evaluated by the Accuracy, but by Recall. The Recall is the percentage of positive elements that are predicted as positive, thus the number of true positives (TP) divided by the number of true positives and false negatives (FN), so Recall = TP/(TP+FN). In this research that would be the percentage of the experimental paragraphs that are identified as such. Accuracy is the percentage of correctly classified paragraphs as experimental and non experimental.

Regarding evaluation of the classification models, a 10-fold cross-validation method is used to test the performance of their performance for each configuration of the processor (Wong, 2015). The first 30 papers are used in the cross-validation, and the last 12 papers for testing the final model. The numbers of experimental paragraphs and non-experimental paragraphs are unbalanced in scientific publications and consequently in the training set. Hence, the number of non-experimental paragraphs for each paper is reduced to the number of experimental paragraphs in the train set. The non-experimental paragraphs that are filtered out of the train set are chosen randomly for each paper. Without re-balancing the cases the Recall of the models decreased by 50% compared to the final results.

We used Python (3.6) and NLTK (3.2.5), a natural language processing module in Python, to implement the experiment. The code is deposited in Zenodo, a repository funded by the European Commission: `https://doi.org/10.5281/zenodo.1202310`.

Table 1: Confusion matrix final model (Nave Bayes)

	Predicted Experimental	**Predicted Other**	**Total**
Experimental [True]	426	63	*489*
Other [True]	327	1197	*1524*
Total	*753*	*1260*	*2013*

3.1 Pre-process Features

The corpus was built from 42 full full-text PDFs (i.e. published articles from one laboratory in the Netherlands) transformed into text files. Next, the experimental sections of each paper were manually extracted from the full text version to serve as training instances. In the end, each publication resulted into three text files: (1) full text, (2) manually extracted experimental section and (3) remaining paragraphs. To transform the paragraphs into features that can be used by the classification models, the text documents need to be split up into paragraphs. Then, features must be selected for each paragraph. For this, a feature selection class was build which can transform the text of a paragraph into word features with the corresponding frequency. While exploring the basic features through word tokenizing the paragraphs, the features showed noise like single (punctuation) characters, numeric values, and words that were hyphenated in the text resulting in two different words. Therefore, for each paragraph the punctuation characters were removed, and hyphenated words were combined. For this experiment, three feature selection methods are examined: (1) removal of a single character words, (2) removal of numeric values, and (3) replacement of numeric values by the string float. Numeric values contain very specific values like number of runs, output from instruments, etc. Therefore, removing them might prevent over-fitting on numeric values. At the same time, as the number of numeric values in a paragraph might depend on whether the paragraph is experimental or not, replacing the numeric values with one string might improve the model Accuracy artificially.

Additionally, as discussed in section 2, two main feature selection methods are stop-word removal and stemming/lemmatizing (Aggarwal and Zhai, 2012). To test whether one of these methods improves the classification Accuracy, three feature selection methods are added. Removing stop words, stemming words, and lemmatizing words. These parameters are used to configure the text pre-processor. The methods removing or replacing floats and, stemming or lemmatizing words are both combined into a single parameter as they logically cannot co-exist.

3.2 Parameters tuning

After the processor class has processed the paragraphs, a list of features with their corresponding frequency in the paragraph is created and linked to the classification of the paragraph. This is used to train the classification models and test their performance. Naive Bayes, Logistic Regression, SVM, and CNN models have shown to be very accurate for sentiment analysis [1]. For selecting the models that are included in this experiment two criteria were used:

1. The model must be relatively simple and transparent. It implies that features must interpretable (Rudin, 2014)
2. The model must be able to work with a bag-of-words representation of the paragraphs.

Using the first criteria, CNN models is excluded due to their complexity. The Linear SVM model is chosen as it is more easily interpretable. Using the second criteria, the Multinomial Naive Bayes model works with the frequency of the features. Thus, the experiment included the Multinomial Naive Bayes model, the Linear SVM model, and the Logistic Regression model. To compare the models, a 10-fold cross-validation method is used [36].In each iteration of the cross-validation, the models are trained and their performance is tested with the selected fold. For evaluating the performance on the test set both the Accuracy and the Recall are calculated.

3.3 Final model selection

Model training is the final stage of the worklfow. The best performing model, Naive Bayes, see Table 1 was trained. For each feature selection parameter, the value with the highest Recall is chosen. Then, the classification model with the highest Recall is trained. The training set used for the 10-fold cross-validation was used for training this model. Then, the test set was used for testing the model. The test set has an unbalanced ratio between experimental and non experimental paragraphs to test the performance of the Naive Bayes classifier on data which is representing unprocessed full-text publications. The final model is trained using a Multinomial Naive Bayes model. It is the model which obtained the highest Recall, including the feature selection methods. The final model is trained on the first 30 papers, with a balanced ratio between the experimental and non-experimental paragraphs and tested on the last 12 papers with all of their paragraphs included.

3.4 Named-Entity Recognition

A simple name entity recognition algorithm has been implemented for the purpose of this experiment. The implementation is based on the *ne_chunk* class from NLTK. The aim is to be able to visualize and compare the entities extracted from full-text, manually extracted experimental paragraphs and predicted experimental paragraphs.

4 Results

The results of our computational experiment show data for 36 different configurations of the feature selection process.For each configuration method a 10-fold cross-validation method was used. In total, 3 different classification models were evaluated, which sums up to 1080 unique tests. The values shown in Table 2 are the mean Accuracy and the mean Recall obtained during the training of the three classifiers. As can be seen, the impact on Accuracy and Recall under different settings (e.g. with or without stop words, with stems, with lemmas etc.) is extremely subtle. The values differ only by +/- 1%. This indicates that the used feature selection methods do not have a significant impact on the performance of the classifier in our case.

4.1 Final model selection

The results of the Naive Bayes classifier on the unbalanced test set show an Accuracy of 80.63% and a Recall of 87.12%. Table 2 shows the confusion matrix of the final model on test data. In total, the model was tested on 2013 paragraphs: 489 experimental and 1524 non-experimental paragraphs. The specificity (true negative rate) of the final model is

Table 2: Feature selection where tuning different parameters appears to have limited impact on Accuracy and Recall for all possible configurations (n= 1080)

Filter	Value	Mean Accuracy	Mean Recall	# Tests
Stop word	False	0.825	0.82	540
Stop word	True	0.817	0.814	540
Single character	False	0.821	0.816	540
Single character	True	0.821	0.819	540
Filter and Replace float	False	0.82	0.814	360
Replace float	True	0.819	0.814	360
Filter float	True	0.825	0.824	360
Stem and Lemmatize	False	0.821	0.815	360
Stem	True	0.823	0.819	360
Lemmatize	True	0.82	0.818	360

78.54%, which indicates that almost 4 out of 5 non-experimental paragraphs are identified as such.

4.2 Named-Entities comparison

As an example of NE extraction, a single paper is used. NEs are recognized for all (full-text) paragraphs, manually extracted experimental paragraphs, and the predicted experimental paragraphs. The last column is using the final trained model to filter out non-experimental paragraphs. In total, the paper has 155 para-graphs, with 30 observed experimental paragraphs, and 68 predicted experimental paragraphs. The results show that in total 182 tokens are recognized in all the para-graphs, and 49 of them are also recognized in the observed experimental paragraphs. Thus, the tokens of all the paragraphs consists for 73% of noise. The tokens recognized in the predicted experimental paragraphs contain 79.6% of the tokens, retrieved from manually extracted experimental paragraphs, and, 6.8% of the noise. In other words, although the noise has been reduced with 93.2%, 20.4% of the tokens from the manually extracted experimental paragraphs has been lost using the final model.

5 Discussion and Limitations

For stemming and lemmatizing words, the results showed that stemming words leads to the highest Recall and Accuracy, and lemmatizing words has a lower Accuracy compared to leaving the words intact. As mentioned in section 2, lemmatizing words only works when the lexicons are studied exhaustively [18], which is not the case for proteomics jargon. The most informative features from a simple Naive Bayes classifier (e.g. the, of, and, in, a, to, were, with, for, wa, **protein**, use, at, by, **peptid**, mm, as, **cell**, from, is) show that features captured proteomics domain specific (stemmed) words predicting experimental paragraphs. These features are not, however, specific to experimental paragraphs.

From the classification models, the Multinomial Naive Bayes model outperformed the other models in mean Recall, but Logistic Regression performed significantly better in Accuracy than the other two classifiers. The Linear SVM model performed the worst of the three in general, which might be explained by the fact that the training set contains more features than paragraphs.

This also explains why the final results show a higher Recall and a lower Accuracy, as the data on which the model is trained is larger and the ratio between experimental paragraphs and non-experimental paragraphs is not balanced. Models are expected to perform better when trained on a larger data set, but as Multinomial Naive Bayes is better in predicting experimental paragraphs than non-experimental paragraphs, its Accuracy logically drops as there are more non-experimental paragraphs in the test set. Yet, the goal of reducing the corpus (and consecutively the noise in the result lists of tokens), is achieved with this model, and the noise has still been reduced by 93% for the tested paper.

There are several limitations to our approach. First, we used a limited corpus from a specific domain. indeed, our goal is more focused on exploring the feasibility of such an approach than to develop a complete analysis pipeline. Next, another limitation is that we did not explore sufficiently further analysis steps. We reached a level where the output of the classification algorithm can be used in future processing and analysis activities without having tested further. A first improvement is to apply other unsupervised techniques such as latent-semantic analysis (LDA) and entity-recognition (ER) algorithms to extract experimental procedures and knowledge from full-text publications. Examples of knowledge can be specific software, data sets, repositories, availability information directly from a reduced corpus. In that respect, our approach is relevant for decreasing the amount of text from publications as the amount of articles published is increasing. In the Proteomics domain, the number of articles doubled in ten years (2004-2014, source Scopus , query: *TITLE-ABS-KEY (proteomics) AND (LIMIT-TO (SUBJAREA , "BIOC")))* . Therefore, it might be unpractical to train topic modeling algorithms and entity recognition of the complete set of paragraphs of each publication, knowing that experimental paragraphs form a minority of the document.

6 Conclusion

A Multinomial Naive Bayes model was trained using the training set of 30 papers and tested on the resulting 12 papers. The words in the paragraphs of each paper were stemmed and floats where filtered out. The final model shows an Accuracy 80.63% and a Recall of 87.12% on the test set. A simple named-entity recognition test on one of the test papers shows that using the final model, the noise in the recognized tokens from all the paper's paragraphs is reduced by 93.2% when only using the extracted experimental paragraphs. But, at the same time, 20.4% of the recognized tokens from the paper's observed experimental paragraphs are lost.

The main challenges encountered during this experiment is a classification problem on an unbalanced data set due to the fact that experimental sections yield a minority of paragraphs in publications. Nevertheless, for reproducing published experiments, the main information is described in those paragraphs. We showed that it is feasible to extract experimental paragraphs with a simple and transparent classifier. Despite the limitations, such as not achieving 100% recall, filtering out non-experimental paragraphs can generate more useful lists of named entities to extract experimental resources and methods from scientific publications.

REFERENCES

Aggarwal, C. C. and Zhai, C. (2012). A survey of text classification algorithms. *Mining text data*, pages 163–222.

Bandrowski, A. E. and Martone, M. E. (2016). Rrids: A simple step toward improving reproducibility through rigor and transparency of experimental methods. *Neuron*, 90(3):434–436.

Begley, C. G. and Ellis, L. M. (2012). Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391):531–533.

Bosman, J. (2006). Reporters find science journals harder to trust, but not easy to verify. *New York Times*, 13.

Braarvig, J., Bréchot, C., Bullinger, H.-J., Einhäupl, K. M., Elkana, Y., Gaehtgens, P., Galluzzi, P., Geisselmann, F., Gruss, P., Guédon, J.-C., Henkel, H.-O., Kröll, W., Larrouturou, B., Leon, J. M. R., Mittelstraß, J., Roth, M., Schirmbacher, P., Simon, D., and Winnacker, E.-L. (2003). Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities.

Casadevall, A. and Fang, F. C. (2010). Reproducible science. *Infection and immunity*, 78(12):4972–5.

Chan, L., Cuplinskas, D., Eisen, M., Friend, F., Genova, Y., Guédon, J.-C., Hagemann, M., Harnad, S., Kupryte, R., Johnson, R., Manna, M. L., Rév, I., Segbert, M., de Souza, S., Suber, P., and Velterop, J. (2002). Budapest Open Access Initiative — Budapest Open Access Initiative.

Chokshi, F., Shin, B., Lee, T., Lemmon, A., Necessary, S., and Choi, J. (2017). Natural language processing for classification of acute, communicable findings on unstructured head ct reports: Comparison of neural network and non-neural machine learning techniques. *bioRxiv*, page 173310.

Colas, F. and Brazdil, P. (2006). Comparison of svm and some older classification algorithms in text classification tasks. *Artificial Intelligence in Theory and Practice*, pages 169–178.

Debortoli, S., Müller, O., Junglas, I., and vom Brocke, J. (2016). Text mining for information systems researchers: An annotated topic modeling tutorial. *Communications of the Association for Information Systems*, 39(1):110–135.

Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., Petrak, J., and Bontcheva, K. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49.

Donoho, D. L. (2010). An invitation to reproducible computational research. *Biostatistics (Oxford, England)*, 11(3):385–8.

Fanelli, D. (2013). Why growing retractions are (mostly) a good sign. *PLoS medicine*, 10(12):e1001563.

Fang, F. C., Steen, R. G., and Casadevall, A. (2012). Misconduct accounts for the majority of retracted scientific publications. *Proceedings of the National Academy of Sciences*, 109(42):17028–17033.

Genkin, A., Lewis, D. D., and Madigan, D. (2007). Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304.

Ingason, A. K., Helgadóttir, S., Loftsson, H., and Rögnvaldsson, E. (2008). A mixed method lemmatization algorithm using a hierarchy of linguistic identities (holi). In *Advances in Natural Language Processing*, pages 205–216. Springer.

Jivani, A. G. et al. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938.

Korenius, T., Laurikkala, J., Järvelin, K., and Juhola, M. (2004). Stemming and lemmatization in the clustering of finnish text documents. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 625–633. ACM.

Lacey, A., Lyons, J., Akbari, A., Turner, S. L., Walters, A. M., Fonferko-Shadrach, B., Pickrell, O., Rees, M. I., Lyons, R. A., Ford, D. V., et al. (2017a). Codifying unstructured data: A natural language processing approach to extract rich data from clinical letters. *International Journal for Population Data Science*, 1(1).

Lacey, A. S., Fonferko-Shadrach, B., Lyons, R. A., Kerr, M. P., Ford, D. V., Rees, M. I., and Pickrell, O. W. (2017b). Obtaining structured clinical data from unstructured data using natural language processing software. *International Journal for Population Data Science*, 1(1).

Laine, C., Goodman, S. N., Griswold, M. E., and Sox, H. C. (2007). Reproducible research: Moving toward research the public can really trustmoving toward research the public can trust. *Annals of Internal Medicine*, 146(6):450–453.

Liu, D., Li, Y., and Thomas, M. A. (2017). A roadmap for natural language processing research in information systems. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.

Manevitz, L. M. and Yousef, M. (2001). One-class svms for document classification. *Journal of Machine Learning Research*, 2(Dec):139–154.

McNutt, M. (2014). Journals unite for reproducibility. *Science*, 346(6210):679–679.

Nekrutenko, A. and Taylor, J. (2012). Next-generation sequencing data interpretation: enhancing reproducibility and accessibility.

Nicholls, S. G., Langan, S. M., Benchimol, E. I., and Moher, D. (2016). Reporting transparency: making the ethical mandate explicit. *BMC medicine*, 14(1):44.

Peng, R. D., Dominici, F., and Zeger, S. L. (2006). Reproducible epidemiologic research.

Rudin, C. (2014). Algorithms for interpretable machine learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, pages 1519–1519.

Steen, R. G., Casadevall, A., and Fang, F. C. (2013). Why has the number of scientific retractions increased? *PloS one*, 8(7):e68397.

Suber, P., Brown, P., Cabell, D., Chakravarti, A., and Cohen, B. (2003). Bethesda statement on open access publishing.

Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(9):2839–2846.