The use of text-mining and machine learning algorithms in systematic reviews: reducing workload in preclinical biomedical sciences and reducing human screening error

Bannach-Brown, A.[1,5] (https://orcid.org/0000-0002-3161-1395) , Przybyła, P.,[2] Thomas, J.[3] (https://orcid.org/0000-0003-4805-4190), Rice, A.S.C.[4] (https://orcid.org/0000-0002-1237-4769) , Ananiadou. S.[2], Liao, J.[1], Macleod, M.R.[1] (https://orcid.org/0000-0001-9187-9839)

1 Centre for Clinical Brain Sciences, University of Edinburgh.

2 National Centre for Text Mining, School of Computer Science, University of Manchester.

3 EPPI-Centre, Department of Social Science, University College London.

4 Pain Research, Department of Surgery and Cancer, Imperial College.

5 Translational Neuropsychiatry Unit, Aarhus University.

Abstract:

Background: In this paper we outline a method of applying machine learning (ML) algorithms to aid citation screening in an on-going broad and shallow systematic review, with the aim of achieving a high performing algorithm comparable to human screening.

Methods: We tested a range of machine learning algorithms. We applied ML algorithms to incremental numbers of training records and recorded the performance on sensitivity and specificity on an unseen validation set of papers. The performance of these algorithms was assessed on measures of recall, specificity, and accuracy. The classification results of the best performing algorithm was taken forward and applied to the remaining unseen records in the dataset and will be taken forward to the next stage of systematic review. ML was used to identify potential human errors during screening by analysing the training and validation datasets against the machine-ranked score.

Results: We found that ML algorithms perform at a desirable level. Classifiers reached 98.7% sensitivity based on learning from a training set of 5749 records, with an inclusion prevalence of 13.2% (see below). The highest level of specificity reached was 86%. Human errors in the training and validation set were successfully identified using ML scores to highlight discrepancies. Training the ML algorithm on the corrected dataset improved the specificity of the algorithm without compromising sensitivity. Error analysis sees a 3% increase or change in sensitivity and specificity, which increases precision and accuracy of the ML algorithm.

Conclusions: The technique of using ML to identify human error needs to be investigated in more depth, however this pilot shows a promising approach to integrating human decisions and automation in systematic review methodology.

Key-words: machine learning, systematic review, analysis of human error, citation screening, automation tools

Introduction:

The rate of publication is increasing exponentially within biomedical research [1]. Researchers are finding it increasingly difficult to keep up with new findings and discoveries even within one biomedical domain, an issue that has been emerging for a number of years [2]. Synthesising research – either informally or through systematic reviews to provide an unbiased summary also becomes an increasingly resource intensive task as search strings retrieve larger and larger corpuses of potentially relevant papers for reviewers to screen for relevance to the research question at hand.

Within the animal model literature we see this increase in rate of publication. In an update to a systematic review of animal models of neuropathic pain, 11,880 further unique records were retrieved in 2015, in addition to the 33,184 unique records identified initially in the search in 2012. In the field of animal models of depression, the amount of unique records retrieved from the search has increased in 1 year from 70,365 (May 2016) to 76,679 (August 2017).

The use of text-mining tools and machine learning (ML) algorithms to aid systematic review is becoming an increasingly popular tool to reduce human burden and monetary resources required and increase efficiency to complete systematic reviews as well as increase the speed at which results are produced [3; 4; 5]. ML algorithms are primarily employed at the screening stage in the systematic review process. This screening stage involves categorising records identified from the search into 'Relevant' or 'Not-Relevant' to the research question, typically carried out to the gold standard by two independent human reviewers. This decision is typically made using the title and abstract of an article in the first instance. In previous experience at CAMARADES, screening a preclinical systematic review with 33,184 independent hits took 18 person months in total, which can be divided between the number of reviewers available on a project. With two reviewers working concurrently, it took 9 months to screen 33,184 records. Based on this, we have estimated that a systematic review with roughly 10,000 publications retrieved takes an estimated 40 weeks. In clinical systematic reviews, Borah and colleagues [6] showed the average clinical systematic review registered on PROSPERO takes

an average 67.3 weeks to complete. ML algorithms are employed to learn this categorisation ability, based on training instances that have been screened by human reviewers. This approach is an example of using supervised machine learning for automatic classification in systematic review screening, where the algorithm learns the classification decision based on example information given by human screeners. Other machine learning approaches for screening in systematic reviews include supervised learning for document prioritisation, often used for rapid evidence assessments, where the most potentially relevant documents are presented to reviewers first for screening [7; 5]. Further, active learning, which involves the algorithm selecting data that it is trained on, as opposed to a random sample. The concept being that the classifier presents the expert reviewer with the next record that it 'thinks' it will learn most from [8; 9]. Different approaches can be taken on how to use the machine learning decision, whether, with high enough performance, review teams accept the decision of the machine, or whether to use the machine as a second screener.

In this paper we outline the approach taken to screen a novel corpus for a broad and shallow systematic review question, non-human animal models of depression, a corpus of 70,365 records retrieved from two online biomedical databases. In the context of systematic review, ML algorithms have often been tested in projects, where a large corpus of 'already-classified' data is available, for example in the case of updating a systematic review [3]. *Here, the aim was to identify the amount of training data required for an algorithm to achieve the level of performance equivalent to the gold standard of two independent human screeners.* The goal performance for the ML algorithms was performance at the human gold standard of 95% sensitivity and to maximise performance for specificity.

Sena and colleagues developed guidelines for the appraisal of systematic reviews of animal studies [10]. These guidelines consider dual extraction or two independent human reviewers as a feature of a high quality review. The inter-screener agreement in systematic reviews at CAMARADES is estimated to be between 95% and 99%. This leaves a 1-5% human error rate. This can include both random error

due to fatigue or distraction of human screeners, or – more consequentially - a systematic error, which, if included in a training set, might be propagated into a ML algorithm. Sources of systematic error at the screening stage could be if there are differences in the way screeners interpret inclusion criteria. As far as we are aware, the nature of this 5% residual human error in systematic review methodology has not been formally investigated. The training data used for ML categorisation is based on training instances that has been screened by two independent human screeners. *We therefore aimed to explore the use of ML algorithms as part of a systematic review framework at the classification stage, to investigate if the ML algorithms could be used to improve the human gold standard by identifying human screening errors and thus improve the overall performance of ML.*

Methods:

The methodology of this paper is outlined as follows. Firstly, different machine learning algorithms were applied to assist in the screening of a large systematic review (> 70,000 records identified). Three sub-sets of the 70,365 records were screened by two human reviewers. These sub-sets were used to train the algorithms. Performance of the classifiers was assessed on a validation set of unseen documents, and a number of different metrics were used. Secondly, the best performing algorithm was used to identify human error in the training and validation sets. The error analysis is assessed on the net reclassification index.

Step 1: Application of ML tools to screening of a large preclinical systematic review.

Training Sets:

70,365 potentially relevant records were identified from Pubmed and EMBASE (for search strings see [11]). The training sets were chosen at random from the 70,365 by assigning each record a random number using the RAND function in excel and ranking them from smallest to largest. The first 1993

records were chosen for the first training set. The second training set consisted of adding the next 996 records, bringing the training set size to 2989. The third training set consisted of the adding the next 2760 records, bringing the total to 5749 records. These datasets were intended to be round numbers (i.e.2000 and 1000), however data disposition errors occurred when transferring the records between databases, hence the non-even numbers of records in each sub-dataset. The total number of training records was 5749. The training sets were screened by two independent human screeners. (Datasets are available on Zenodo, see "Availability of Data & Materials" below). Performance was assessed at each level on a validation set of unseen records. All training and validation sets were selected consecutively from the initial random ordering. The first training set had a validation set of 996 records. The second training set had a validation set of 1010 records. And for the full training set of 5749 records, the validation set was the subsequent 1251 records. This validation set had more than 150 "included" records, which can give reasonably precise 95% confidence intervals for sensitivity and specificity.

Feature Generation:

First, documents in the training set were transformed into a representation that is appropriate for the machine learning algorithms. Namely, every case (document) is represented by a fixed number of features, i.e., numerical quantities describing certain properties that could be used by the classifier to extract rules and make predictions about it. The classifiers described below used similar approaches, though there were differences.

The "bag-of-words" model was used to characterise document features – their titles and abstracts – in both classifiers. This approach to translating text to a numeric representation is commonly used in document classification and, at its most basic, is simply the frequency count of each word in a document. In order to account not only for the relative importance of words within a document, 'Term Frequency – Inverse Document Frequency – is often used. This helps the classifier to focus on terms

which help to distinguish between documents, rather than on terms which occur frequently [12]. Technically, each word is known as a 'gram' – and the simple indexing of individual words would be described as being a 'uni-gram' bag of words. 'n-gram' bag of words are usually used – where 'n' indicates the number of words that are included (e.g. 'bi-gram' bag of words approaches index every word, and also every pair of words). Lastly 'stopword' lists are used, to remove frequently occurring words which provide little relevant information for classification purposes.

Bag of words representation is straightforward and easy to implement, but has a serious drawback: it generates as many unigram features as there are words in the collection (typically at least several thousand), and much more when using higher-order n-grams. This is known to be a problem for general-purpose classifiers, which have to be extended with additional feature selection stages. Moreover, this multi-dimensionality is related to high redundancy, as quite often different words refer to the same underlying concept, e.g. in case of synonymy or morphological variants. That is why several techniques have been developed for representing textual data in a more efficient way.

One such option is Latent Semantic Indexing (LSI) [13]. In LSI the training set is represented as a matrix, where rows correspond to documents, columns to terms (words or n-grams), while cells contain frequency or TF/IDF score of a given term in a given document. The matrix is then decomposed using a general matrix factorisation technique: Singular Value Decomposition (SVD) and truncated to the first *n* dimensions. Thanks to the properties of SVD, the new features will be such linear combinations of features of the old space that minimise the differences between the original and the transformed space. In case of textual data it means that those words that frequently occur in the same documents (probably because of the similar meaning) will be treated in the same way. The *n* is set a-priori to a reasonably low value – usually a few hundred.

Latent Dirchlet Allocation (LDA) is another method aiming to provide lower-dimensional representation of documents by exploiting distributional similarities between words, but based on explaining document contents using a Bayesian network [14]. This method is based on the premise

that every document is a mixture of topics, which in turn consist of related words. The correspondence between documents and topics and between topics and words could be inferred via Gibbs sampling process. As a result, similarly to LSI, every document is represented by a sequence of *n* numbers, indicating how related it is to every topic [15]. As well as being used to identify distinct (and overlapping) clusters of documents, topic model representation can be included as feature space for document classification tasks, including screening [9]. Other related methods exploiting vocabulary similarities for document representation include paragraph vectors and descriptive clustering [16; 17].

Classifiers:

Following the transformations made in feature selection, the documents are then used to train the machine learning classifier. One of the most commonly used document classifiers is the Support Vector Machine (SVM). SVM is a supervised learning algorithm, learning to classify new documents based on a training set of labelled documents [18]. This algorithm represents training documents as points in a feature space and seeks the optimal hyper-plane separating positive and negative cases, i.e. included and excluded documents. The new, unseen, documents are then ranked by their distance from the optimal hyper-plane. Logistic regression is a similar linear classifier, which instead of hyperplane, seeks such coefficients of a linear combination of feature values that will give high values for positive cases (included documents) and low for negative (excluded documents). Both of these approaches could be enriched with feature selection elements to mitigate the problems with multitude of features.

After testing a number of machine learning approaches, these two algorithms (below) performed the best for this dataset of 70,365 records, on the broad topic of preclinical animal models of depression.

Linear classifiers:

Classifier 1, 3, & 5:

Classifier one, three, and five used a tri-gram 'bag-of-words' model for feature selection and used a simple linear classifier. The classifier selected applied Stochastic Gradient Descent as implemented in the SciKit-Learn python library [19]. This classifier was chosen as previous internal evaluations showed its results to be indistinguishable from SVM; moreover, it is efficient, scales well to large numbers of records, and provides an easily interpretable list of probability estimates when predicting class membership (i.e. scores for each document lying between 0 and 1). Efficiency and interpretability are important, as this classifier is deployed in a large systematic review platform [20], and any deployed algorithm therefore needs not to be too computationally demanding, and its results understood by users who are not machine learning specialists. The tri-gram feature selection approach without any additional feature engineering also reflects the generalist need of deployment on a platform used in a wide range of reviews: the algorithm needs to be generalisable across disciplines and literatures, and not 'over-fitted' to a specific area. This approach aims to give the best compromise between reliable performance across a wide range of domains and that achievable from a workflow that has been highly tuned to a specific context.

Classifier 2 & 4:

Classifier two and four used a regularised logistic regression model built on LDA and SVD features. Namely, the document text (consisting of title and abstract) was first lemmatised with GENIA tagger [21] and then converted into bag of words representation of unigrams, which was then used to create two types of features. First, the word frequencies were converted into a matrix TF/IDF scores, which was then decomposed via SVD implemented in scikit-learn library and truncated to the first 300

dimensions. Second, an LDA model was built using MALLET library [22], setting 300 as a number of topics. As a result each document was represented by 600 features, and an L1-regularised logistic regression model was built using glmnet package [23] in R statistical framework [24].

Thanks to this procedure every document is represented with a constant, manageable number of features, no matter what corpus or vocabulary size is. As a result, we can use a relatively simple classification algorithm and obtain good performance with short processing time even for very large collections. This feature is particularly useful when running the procedure numerous times in cross-validation mode for error analysis (see below).

Algorithms assign each record a score of predicted probability of being relevant and an optimal cut-off score that gives the best performance is calculated.

Assessing Machine Learning Performance:

The facets of a machine learning algorithm performance that would be beneficial to this field of research are high sensitivity, comparable to the current gold standard, two independent human screeners, which we estimate at 95%. We accept an error rate of 5%, missing out on 5% of relevant studies, is permissible. We therefore aim to achieve at least 95% sensitivity, including the lower 95% confidence bound. Once the level of sensitivity has been reached, the aim is to maximise specificity, in order to reduce the amount of irrelevant records included by an algorithm.

Performance metrics:

Performance was assessed using sensitivity (or recall), specificity, precision, accuracy, and Work Saved over Sampling (WSS) (see table 1), carried out in R (R version 3.4.2; [24]) using the 'caret' package [25]. 95% Confidence Intervals were calculated using the efficient-score method [26].

235

Table 1. Equations used to assess performance of machine learning algorithms

Sensitivity or Recall	TP / (TP+FN)
Specificity	TN / (TN+FP)
Precision	TP / (TP+FP)
Accuracy	(TP+TN) / (TP+FP+FN+TN)
WSS@95%	((TN+FN) / N) – (1.0 – 0.95)

All equations from [5].

236

237

238 **Step 2: Application of ML tools to training datasets to identify human error.**

239 **Error Analysis Methods:**

240 The methodology for the error analysis was outlined in an *a priori* protocol, published on the
241 CAMARADES website 18th December 2016 [27]. To generate the machine learning scores for training
242 set records, the non-exhaustive cross-validation method, k-fold validation, was used. This method
243 involved randomly partitioning the test set into *k* number of equal sized subsamples. One subsample
244 was set aside for validation, and the remaining k -1 subsamples were used to train the algorithm [28].
245 Here k = 5. This process was carried out until all records in the training set had a machine learning
246 algorithm assigned score. The scores were used to draw attention to discrepancies or disagreements
247 between machine decision and human decision, by ranking the machine assigned labels in order of
248 predictive probability, from most likely to be relevant to least likely to be relevant, and highlighting
249 potential false positive and potential false negative cases, i.e. errors in the human decision. In order
250 to avoid reassessing the full 5749 record dataset for false positive and false negative results, a stopping
251 rule was established. If the initial human decision was correct five consecutive times, further false
252 positive, or false negative, records were not reassessed.

253 After the errors in the training set were investigated and corrected, a new model was built on the
254 updated training data. The outcome of error analysis is presented as reclassification tables and the
255 net reclassification index [29] is used to compare the performance of the classifier built on the

updated training data with the performance of the classifier built on the original training data. The following equation was used:

$$NRI_{\text{binary outcomes}} = (\text{Sensitivity} + \text{Specificity})_{\text{second test}} - (\text{Sensitivity} + \text{Specificity})_{\text{first test}}$$

[30]

Further, we applied the same technique as above to identify human screening errors in the validation dataset. Due to the small number of records in the validation set (1251 records), it was assumed that every error would be likely to impact performance, therefore, the manual screening of the validation set involved revisiting every record where the human and machine decision were incongruent. The number of reclassified records was noted.

Results:

In this section we display the performance from the ML algorithms. We display the results from the analysis of human error. And we show the performance of the ML algorithm after human errors in the training and validation set have been corrected.

Performance of Machine Learning Algorithms

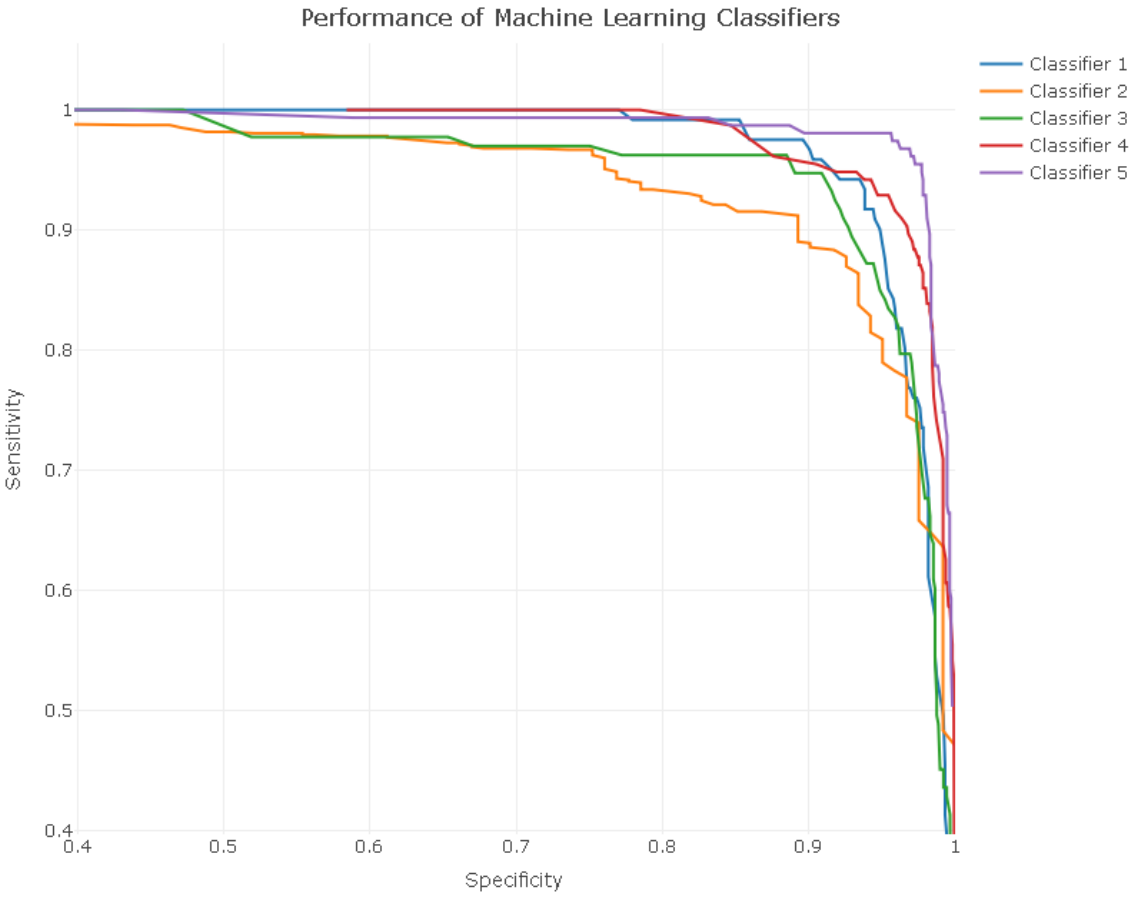
Table 2 shows the performance of the two best machine learning approaches from the SLIM collaboration. The desired sensitivity of 95% has been reach by both classifiers. Both classifiers reached 98.7% sensitivity based on learning from a training set of 5749 records, with an inclusion prevalence of 13.2% (see below). Classifier 5 reached a higher specificity level of 86%.

Table 2. Performance of machine learning approaches on depression training dataset.

	Classifier 5	Classifier 4
Training Set Size	**5749**	**5749**
Optimal Cut-Off Score	**0.1**	**0.07**
Sensitivity	**98.7%**	**98.7%**
Upper 95% CI	**0.997**	**0.997**
Lower 95% CI	**0.949**	**0.949**
Specificity	**86.0%**	**84.7%**
Precision	**50%**	**47.66%**
Accuracy	1096/1251 = **87.6%**	1081/1251= **86.4%**
WSS@95%	**0.705**	**0.693**

The performance of the classifiers altered with increasing numbers of training records, and can be visualised as a learning curve (figure 1). Here we see a trade-off between sensitivity and specificity. With increasing amounts of records, specificity is increased for Classifier 5, the sensitivity increases for Classifier 4 to reach above the desired 95%.

Figure 1. Performance of classifiers with increasing amounts of training records. Classifiers 1 & 2 are trained on the first training set of 1993 records. Classifier 3 is trained on the second training set of 2989 records. Classifier 4 & 5 are trained on the third and full dataset of 5749 records. For the interactive version of this plot with cut-off values, see code and data at https://github.com/abannachbrown/The-use-of-text-mining-and-machine-learning-algorithms-in-systematic-reviews/blob/master/Performance%20of%20Machine%20Learning%20Classifiers.html



290

291

292 **Error Analysis & Reclassification**

293 To assess whether machine learning algorithms can identify human error and therefore improve the
294 training data, error analysis was conducted. Seventy-five papers out of 5749 papers were identified
295 by the machine learning algorithm as potential human errors (i.e. where ML classification was correct,
296 human was wrong) and were rescreened by a human. Out of 75 rescreened papers, the machine
297 corrected the human decision 47 times. The machine was wrong, or the initial human decision was
298 correct, 28 times. The validation set was also rescreened. Ten papers out of the 1251 records were
299 identified as potential human errors. Out of 10 errors, the machine corrected 8 human decisions.
300 These 8 records were all falsely excluded by the human and were now included. The initial human
301 decision was correct twice.

To calculate human error in the training set, the number of errors identified (47) out of the training set (5749 records) was calculated to be 0.8%. Of the 47 records reclassified, 11 records were falsely included in the original screening process and were now correctly excluded, and 36 records were falsely excluded in the original screening process and were now correctly included. The machine correctly identified human screening errors, which were calculated to be just under 1% of the dual screened training set. By looking at the prevalence of inclusion in this training set, which is 13.2% (760 out of the 5749), it highlights that it essential to correctly identify relevant papers. Therefore any missing relevant papers, false negatives, or including irrelevant papers, false positives, will have a large impact on the learning of the ML algorithm. Forty-seven papers out of 760 were 'correctly' reclassified, 6% of the included papers.

Similarly, the human error rate in the validation set (1251 records) was 0.6%. Again looking at the prevalence of inclusion in this dataset (155/1251), which is 12.4%, the 8 records of out the now 163 were correctly reclassified which is 4.9% reclassified. All 8 records we falsely excluded in the original screening process and are now correctly included.

Test 1: 98.7% + 86% = 184.7%

Test 2: 98.2% + 89.3% = 187.5%

NRI = 3.2%

We consider the updated validation set to be the new gold standard as 8 records were now included. The confusion matrix for the performance of the machine learning algorithm after the error analysis update on the training records is displayed below in table 3.

Table 3. Reclassification of records in validation after error analysis.

	Test 1 – Original Machine Learning Algorithms results		
Test 2 – Post-error analysis ML results	In	Out	Total
In	153 160	153 116	306 276
Out	2 3	943 972	945 975
Total	155 163	1096 1088	1251

By analysing the human errors identified by the machine learning algorithm, correcting for these errors and re-teaching the algorithm, you can see an increase in performance of the algorithm, particularly sensitivity and thus precision. This can save considerable human time in the screening stage of a systematic review. Consider the remaining approximately 64,000 papers, if the ML algorithm results are 3% more accurate, that is approximately 2000 papers that are correctly 'excluded' that a human reviewer does not need to screen through.

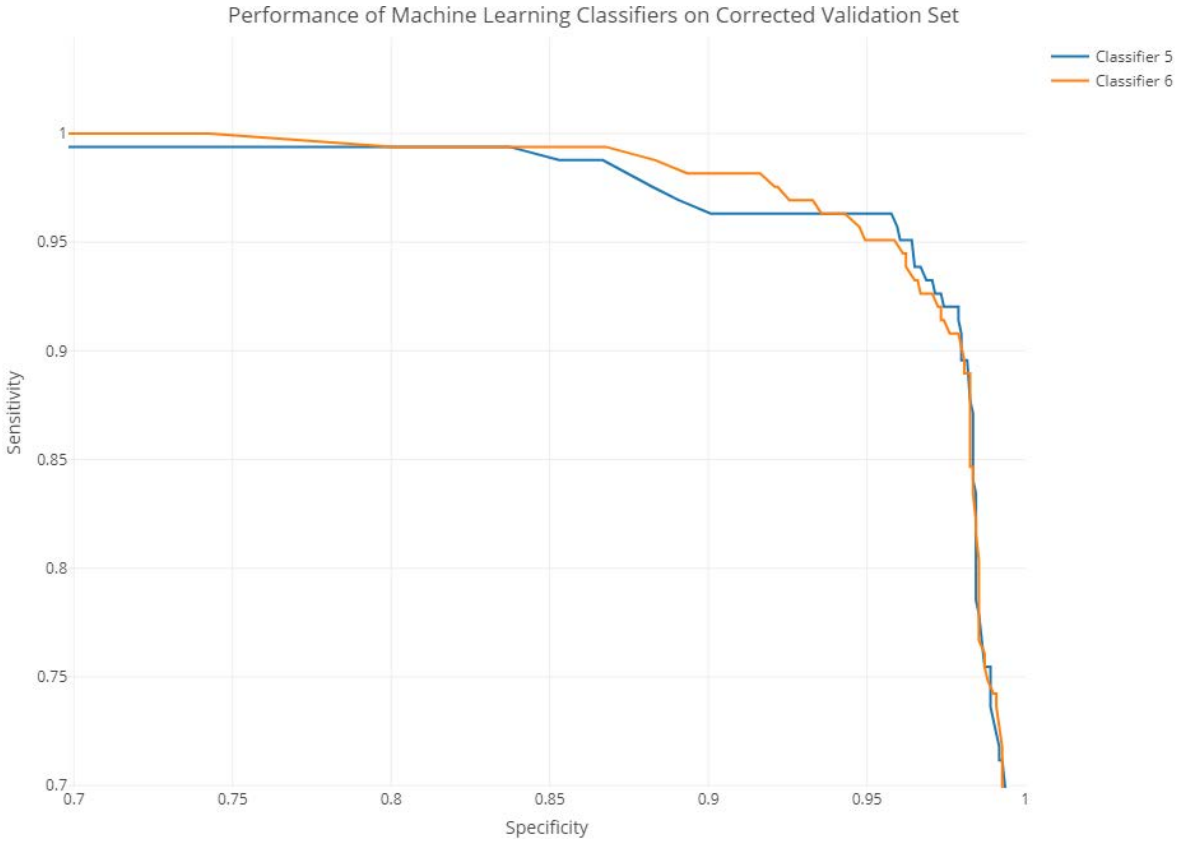
After Error Analysis: Improving Machine Learning

Using the error analysis technique above, of the 47 errors identified in the full training dataset of 5749 records, 0.8% were corrected. We retrained classifier 5 on the corrected training set and measured performance on the corrected validation set of 1251 records as we consider this to be the 'new' gold standard. The performance of the original classifier 5 and classifier 6 was assessed on the corrected validation set of 1251 records. The performance of this retrained algorithm in comparison to the performance of the original classifier 5 on the updated validation set is shown in table 4.

Table 4. Performance of machine learning classifier after error analysis.

	Classifier 6	Classifier 5
Cut-Off	**0.09**	0.10
Sensitivity	**98.7%**	98.7%
Upper 95% CI of Sensitivity	**0.997**	0.997
Lower 95% CI of Sensitivity	**0.949**	0.949
Specificity	**88.3%**	86.7%
Precision	**55.9%**	52.61%
Accuracy	**89.7%**	88.2%
WSS@95%	**961/ 1251 – (0.05) = 0.718**	945/1251 – (0.05) = 0.705

Figure 2. Performance of classifiers after error analysis. Classifier 6 is classifier 5 retrained on the corrected training set after error analysis correction. Performance on both Classifier 5 and 6 is measured on the corrected validation set (with error analysis correction). For the interactive version of this plot with exact cut-off values, see code and data at https://github.com/abannachbrown/The-use-of-text-mining-and-machine-learning-algorithms-in-systematic-reviews/blob/master/Performance-after-error-analysis.html



We compared the area under the ROC curve for classifier 5 and 6. The AUC for Classifier 5 was 0.9272 (95% CI calculated using DeLong method; 0.914-0.9404). The AUC for Classifier 6 was 0.9355 (95% CI calculated using DeLong method; 0.9227-0.9483). DeLong's test to compare the AUC between the ROC of the two classifiers was applied ('pROC' package in R; [31]), Z = -2.3685, p = 0.0178.

Discussion:

Document Classification:

Overall, machine learning algorithms are shown here to have high levels of performance, with a sensitivity at least comparable to two independent human screeners. The decision making process for selecting classifiers in this project was to maximise sensitivity, in order that the minimum of potentially relevant papers or research are missed. Thereafter, algorithms were then chosen for improved/highest specificity. This was to reduce the subsequent human time required to sort through and assess papers.

The two best performing classifiers have similar performance. The slight differences may reflect the method of feature generation. These algorithms have high performance on this specific topic of animal models of depression. As demonstrated previously, the performance of various classifiers can alter depending on the topic and specificity of the research question [3].

In a similar broad preclinical research project, neuropathic pain, it took 18 person months to screen 33,814 unique records – based on these numbers it would take an estimated 40 person months to screen 70,365 unique records. Performance of machine learning tools demonstrated in this paper can greatly reduce the amount of human resource needed for initial title and abstract screening of a large corpus of records retrieved from a broad search.

We have applied the algorithm to the full dataset (remaining 63,365 records) and are in the process of full-text screening. Following this process, it will allow a more in depth learning of the machine that it can apply to any updates to the search.

Error Analysis:

By using the ML algorithm to classify the likelihood of inclusion for each record in the training set, we ranked and highlighted discrepancies between the human inclusion or exclusion decision and the machine decision. Using this technique we identified human errors, which were then corrected to update the training set.

We have successfully identified human screening errors which were calculated to be just under 1% of the training set which was dual screened by two independent human reviewers. This error analysis results in a 3% increase or change in sensitivity and specificity, which has increased precision, accuracy, and work saved over sampling of the algorithm. We observed an increase in specificity of 1.6% without compromise to specificity. In a systematic review with this number of records, or larger, this saves considerable human resources as the number of records required to screen reduces by at least 1125.

This was an initial pilot with stopping criteria where if the initial human decision was correct five consecutive times, further records were not reassessed. However with a more in-depth analysis of the training dataset, investigating every instance where the human and machine decision were incongruent, it is possible that more errors could be identified and correcting them would increase the precision and accuracy of machine learning classifiers, further reducing human resources required for this stage of systematic review

Limitations & Future Directions:

Here we display the best performing algorithms for this dataset with a broad research question. Other dissimilar research questions or topics may require different levels of training data to achieve the same levels of performance, or may require different topic modelling approaches or classifiers. The best performing algorithm, outlined in this paper, is being applied in an ongoing research project, therefore the 'true' inclusion and exclusion results for the remaining 63365 records is not yet known. The 'true' results will unfold with the fullness of time.

These machine learning algorithms are deployed in an existing systematic review online platform, EPPI-Reviewer [20] and are in the process of being integrated into the SyRF tool, which is focused on the preclinical domain (www.app.syrf.org). This will improve the ease of use of machine learning functions for systematic reviewers, increase the usage of machine learning algorithms for systematic review and significantly reduce the amount of human resources required to conduct systematic review across a range of topics. By allowing a degree of user control over which classifiers and the levels of performance are required for each specific research project. With a broad collaboration such as SLIM we aim to test many ML algorithms across a range of research topics to identify which classifiers perform best under which circumstances, to be able to provide recommendations to users of SyRF.

This paper outlines a pilot approach to using machine learning algorithms to identify human errors in current systematic review methodology. Future research can investigate this concept more thoroughly by setting up a more comprehensive experimental design. After further investigation into the extent of human error in dual reviewing, the picture will be clearer as to the scale of human error and to what extent a machine learning algorithm can identify and aid in rectifying this. These tools can could be integrated into systematic review platforms, such as SyRF (www.app.syrf.org), and may provide feedback to the systematic reviewer during screening, and could ultimately flag incorrectly screened records as the human screens them for inclusion in a dataset for machine training.

Conclusions:

We have demonstrated that machine learning techniques can be successfully applied to an ongoing, broad pre-clinical systematic review. We have demonstrated that machine learning techniques can be used to identify human errors in the training and validation datasets. We have demonstrated that updating the learning of the algorithm after error analysis improves performance. This error analysis technique requires further detailed elucidation and validation. These machine learning techniques are in the process of being integrated into existing systematic review applications to enable more wide-spread use. In future, machine learning and error analysis techniques that are optimised for different types of review topics and research questions can be applied seamlessly within the existing methodological framework.

References:

[1] Bornmann, L., & Mutz, R. (2015). Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, *66*(11), 2215-2222.

[2] Cohen, A. M., Adams, C. E., Davis, J. M., Yu, C., Yu, P. S., Meng, W., ... & Smalheiser, N. R. (2010, November). Evidence-based medicine, the essential role of systematic reviews, and the need for automated text mining tools. In *Proceedings of the 1st ACM international Health Informatics Symposium* (pp. 376-380). ACM.

[3] Howard, B.E., Phillips, J., Miller, K., Tandon, A., Mav, D., Shah, M.R., Holmgren, S., Pelch, K.E., Walker, V., Rooney, A.A. and Macleod, M., 2016. SWIFT-Review: a text-mining workbench for systematic review. *Systematic reviews*, *5*(1), p.87.

[4] Tsafnat, G., Glasziou, P., Choong, M.K., Dunn, A., Galgani, F. and Coiera, E., 2014. Systematic review automation technologies. *Systematic reviews*, *3*(1), p.74.

[5] O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M., & Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic reviews*, *4*(1), 5.

[6] Borah, R., Brown, A.W., Capers, P.L*., et al.* (2017). Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the PROSPERO registry. *BMJ Open*;**7:**e012545. doi: 10.1136/bmjopen-2016-012545

[7] Thomas, J., McNaught, J., Ananiadou, S. (2011). Applications of text mining within systematic reviews. Res Synth Methods,2(1):1–14. 10.1002/jrsm.27

[8] Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2010). Active learning for biomedical citation screening. In Proceedings of the 16th ACM SIGKDD International conference on Knowledge Discovery and Data mining (pp. 173-182). ACM.

[9] Miwa, M., Thomas, J., O'Mara-Eves, A. and Ananiadou, S., 2014. Reducing systematic review workload through certainty-based screening. *Journal of biomedical informatics*, *51*, pp.242-253.

[10] Sena, E. S., Currie, G. L., McCann, S. K., Macleod, M. R., & Howells, D. W. (2014). Systematic reviews and meta-analysis of preclinical studies: why perform them and how to appraise them critically. *Journal of Cerebral Blood Flow & Metabolism*, *34*(5), 737-742.

[11] Bannach-Brown, A., Liao, J., Wegener, G., & Macleod, M.R. (2016). Understanding in vivo modelling of depression in non-human animals: a systematic review protocol. *Evidence-based Preclinical Medicine*, *3*(2), 20-27.

[12] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval, Cambridge University Press: USA.

[13] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391.

[14] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993-1022.

[15] Mo, Y., Kontonatsios, G., & Ananiadou, S. (2015). Supporting systematic reviews using LDA-based document representations. BMC Systematic Reviews, 4(1), 1.

[16] Hashimoto, K., Kontonatsios, G., Miwa, M., & Ananiadou, S. (2016). Topic detection using paragraph vectors to support active learning in systematic reviews. *Journal of biomedical informatics*, *62*, 59-65.

478 [17] Mu, T., Goulermas, J. Y., Korkontzelos, I., & Ananiadou, S. (2016). Descriptive document
479 clustering via discriminant learning in a co-embedded space of multilevel similarities. *Journal of the*
480 *Association for Information Science and Technology*, *67*(1), 106-133.

481 [18] Mertsalov, K., & McCreary, M. (2009). Document classification with support vector machines.
482 Rational Enterprise: White Paper. Accessed from:
483 http://www.rationalenterprise.com/assets/content/files/Classification_with_Support_Vector_Machi
484 nes.pdf , on: 05/09/2016.

485 [19] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P,
486 Weiss R, Dubourg V, Vanderplas J. (2011). Scikit-learn: Machine learning in Python. Journal of
487 machine learning research.12(Oct):2825-30.

488 [20] Thomas, J., Brunton, J., Graziosi, S., (2010). EPPI-Reviewer 4.0: software for research synthesis.
489 EPPI-Centre Software. London: Social Science Research Unit, Institute of Education.

490 [21] Tsuruoka, Y., Tateishi, Y., Kim, J. D., Ohta, T., McNaught, J., Ananiadou, S., & Tsujii, J. I. (2005).
491 Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic Conference on*
492 *Informatics* (pp. 382-392). Springer, Berlin, Heidelberg.

493 [22] McCallum, Andrew Kachites. (2002). "MALLET: A Machine Learning for Language Toolkit."
494 http://mallet.cs.umass.edu.

495 [23] Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear
496 models via coordinate descent. *Journal of statistical software*, *33*(1), 1.

497 [24] R Core Team, “R: A Language and Environment for Statistical Computing.” Vienna, Austria, 2013.

498 [25] Kuhn, M., (2017) “The caret package”. https://topepo.github.io/caret/

499 [26] Newcombe, R.G. (1998)."Two-Sided Confidence Intervals for the Single Proportion: Comparison
500 of Seven Methods," *Statistics in Medicine,* **17**, 857-872

[27] Bannach-Brown, A., Thomas, J., Przybyła, P., Liao, J., (2016). "Protocol for Error Analysis: Machine learning and text mining solutions for systematic reviews of animal models of depression". Published on CAMARADES Website. www.CAMARADES.info. Direct Access: https://drive.google.com/file/d/0BxckMffc78BYTm0tUzJJZkc1alk/view

[28] Rodriguez, J. D., Perez, A., & Lozano, J. A. (2010). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, *32*(3), 569-575.

[29] Kerr, K. F., Wang, Z., Janes, H., McClelland, R. L., Psaty, B. M., & Pepe, M. S. (2014). Net Reclassification Indices for Evaluating Risk-Prediction Instruments: A Critical Review. *Epidemiology (Cambridge, Mass.)*, *25*(1), 114–121. http://doi.org/10.1097/EDE.0000000000000018

[30] Pencina, M.J., D'Agostino, R.B. and Vasan, R.S., (2008). Evaluating the added predictive ability of a new marker: from area under the ROC curve to reclassification and beyond. *Statistics in medicine*, *27*(2), 157-172.

[31] Robin, X. (2017). "pROC" Package. https://cran.r-project.org/web/packages/pROC/pROC.pdf

Declarations:

Availability of Data & Materials:

The training and validation datasets, error analysis datasheets, as well as all the records in the depression systematic review are available on Zenodo: DOI 10.5281/zenodo.60269

The protocol for the systematic review of animal models of depression is available from: http://onlinelibrary.wiley.com/doi/10.1002/ebm2.24/pdf

The protocol for the Error Analysis is available via the CAMARADES website and can be accessed directly from this link: https://drive.google.com/file/d/0BxckMffc78BYTm0tUzJJZkc1alk/view

The results of the classification algorithms and the R code used to generate the results is available on GitHub: https://github.com/abannachbrown/The-use-of-text-mining-and-machine-learning-algorithms-in-systematic-reviews.

Competing Interests:

The authors declare that they have no competing interests.

Funding:

This work is supported by a grant from the Wellcome Trust & Medical Research Council (Grant Number: MR/N015665/1). ABB is supported by a scholarship from the Aarhus-Edinburgh Excellence in European Doctoral Education Project.

Authors' Contributions:

ABB screened and analysed the datasets. JT & PB conducted feature selection and built the classifiers. ABB, JT & PB wrote the manuscript. ABB, JT, PB, MRM, JL, AR & SA devised the study. JL, MRM & SA supervised the study. All authors edited and approved the final manuscript.

Acknowledgements:

Thank you to Kaitlyn Hair & Paula Grill for their assistance in second screening the training and validation datasets.