Reporting Statistical Validity and Model Complexity in Computational Studies

Babatunde Kazeem Olorisade
Keele University
Keele, Staffs ST5 5BG
b.k.olorisade@keele.ac.uk

Pearl Brereton
Keele University
Keele, Staffs ST5 5BG
o.p.brereton@keele.ac.uk

Peter Andras
Keele University
Keele, Staffs ST5 5BG
p.andras@keele.ac.uk

ABSTRACT

Background: : Statistical validity and model complexity are both important concepts to enhanced understanding and correctness assessment of computational models. However, information about these are often missing from publications applying machine learning.

Aim:The aim of this study is to show the importance of providing details that can indicate statistical validity and complexity of models in publications in the context of citation screening automation using machine learning techniques.

Method: We built 15 Support Vector Machine (SVM) models each developed using word2vec (average word) features — and data for 15 review topics from the Text REtrieval Conference (TREC) 2004 dataset.

Results: The word2vec features were found to be sufficiently linearly separable by the SVM and consequently we used the linear kernels. In 11 of the 15 models, the negative (majority) class used over 80% of its training data as support vectors (SVs) and approximately 45% of the positive training data.

Conclusions: In this context, exploring the SVs revealed that the models are overly complex against ideal expectations of not more than 2%-5% (and preferably much less) of the training vectors.

CCS CONCEPTS

• **General and reference** → **Experimentation**; *Empirical studies*; • **Computing methodologies** → *Support vector machines*; Supervised learning by classification;

KEYWORDS

Model complexity, statistical validity, CS automation, systematic reviews, model selection, text mining

ACM Reference format:
Babatunde Kazeem Olorisade, Pearl Brereton, and Peter Andras. 2017. Reporting Statistical Validity and Model Complexity in Computational Studies. In *Proceedings of International Conference on Evaluation and Assessment in Software Engineering, Karlskrona, Sweden, June 2017 (EASE'17),* 6 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

EASE'17, June 2017, Karlskrona, Sweden

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Empirical software engineering is currently witnessing an increased number of studies reporting models from computational research based on machine learning algorithms. A particular example is the use of text mining to automate the Citation Screening (CS) phase of the Systematic Review (SR) process. Systematic review is a rigorous review approach used in software engineering [22] and other disciplines (particularly medicine and education). CS is the process of deciding which of the papers found in the search phase of a SR are relevant and hence should be included in a review and which are not.

There is an on-going campaign on the need for reporting basic information in publications based on computation to ensure that independent researchers will be able to reproduce the results of these studies. The same cannot be said about the statistical validity and complexity of such models. Information that can indicate the statistical validity and complexity of models is rarely reported in studies.

In this study, we explore the need for the explicit provision of statistical validity and complexity details of proposed models in computational studies. In general, a machine learning based model should not be assumed to be statistically valid and/or robust without assessing its complexity — even if such a model is reported to have high performance according to the measures used.

We conduct this study in the context of the automation of CS in SRs using text mining techniques. We build multiple support vector machines (SVMs) using average word-to-vector (word2vec) features, for binary classification of citations (i.e. to automate the inclusion/exclusion of papers). According to Olorisade et al. [29], SVM based models have been proposed in 31% of the studies on the automation of CS in SRs between 2006 — 2014; making it the most used algorithm in the field. Thus, the choice of the SVM algorithm for this study. The datasets are those used for 15 reviews from the Text Retrieval and Evaluation Conference (TREC) 2004 datasets [13]. The datasets are part of the Drug Evaluation Review Program (DERP) reports made available through the collaboration between the Cochrane Centre and the Evidence based Practice Centres of the Agency for Healthcare Research and Quality (AHRQ) [13].

Various measures have been proposed in the context of model complexity, each adopting different information criterion statistics. Some of the early ones are: Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC); recently Minimum Description Length (MDL) and Occam's razor from information theory have become popular. In general, statistical validity requires tight constraints on a model in terms of variation of parameters, less constraint and more possible variation of parameter values

implies lower validity of the model [16]. This implies that models with higher statistical validity are more likely to be replicated with small variation in their parameters.

Computational (model) complexity in the context of machine learning can be simply viewed in terms of resources — the number of required examples, elementary components of a hypothesis etc. In general, a model aims to achieve the best possible data description performance (e.g. classification in the context of making inclusion/exclusion decisions as part of an SR), however at the same time its complexity should be kept as low as possible. Often, in the case of multi-component models, it is difficult to establish confidence intervals for the model predictions due to the complicated and non-trivial joint effects of the multiple components. However, in such cases the model complexity combined with the model's data description performance provide a proxy for the estimation of the model's statistical validity. In general, following the Occam's razor principle, if two models have comparable data description performance, the least complex is assumed to be statistically more valid [15]. Thus, assessing model complexity for multi-component models is critical for the estimation of the statistical validity of the model. Information regarding statistical validity and complexity is generally missing from computational studies in the context of machine learning applications for text classification in SRs.

Viewing complexity as above, what translates to complexity in each model differs. In this study, we illustrate with SVM models where complexity is characterised by the number of support vectors involved in the SVM classifier and is controlled through its hyper-parameters — C, gamma and kernel type. In the rest of the paper, section 2 presents a brief introduction to the Occam's razor principle for choosing a less complex model and a review of the studies that have proposed SVM models in CS automation. The conduct of our experiment is the subject of section 3. Section 4 presents the results and discussion, while section 5 discusses some threats to validity. The conclusions are presented in section 6.

2 BACKGROUND

2.1 Model Complexity

The option of selecting the best model in machine learning is not usually a straightforward one. The rule of thumb is to select a model with the least generalization error. According to Nannen, a good model is the one with low generalization error and low tendency to overfit [27]. However, given two possible representations or models of data, Occam's razor dictates that other things being equal, the simpler or less complex of the two should be preferred [14, 15]. Though, the understanding of this principle has generated a few controversies based on different interpretations and drawing of unsupported conclusions between simplicity and accuracy [15]. Simplicity in this context refers to the representation generated from a less complex hypothesis [7, 15], which may be easier to understand, and/or to explain.

There are a number of ways to determine the complexity of a model, such as minimum description length (MDL) [17, 18] and Kolmogorov complexity [8, 23, 32]. The MDL seeks a model that yields a suitable balance between model accuracy and complexity given the sample size and data complexity [5]. Kolmogorov complexity is the length of the shortest program a finite string can be computed from [17]. Originally used in information theory, it is lately becoming more popular in computational studies. In essence, applied to computational models it implies the preference for the simplest hypothesis that represents or approximates the data. In the case of models with multiple components, the often adopted measure of complexity is the measure of structural complexity, which is given by the number of components of the model. This approach is valid in particular, when each component of the model can be expected to have the same level of complexity as any other component of the model.

2.2 Citation Screening

The CS (or study selection) process is a key activity in SR where the relevant documents are separated from the irrelevant ones. This phase is one of the most time consuming activity of the SR process. It is therefore not surprising that it has attracted the most attention in terms of automating an individual phase of the SR process [25]. Most of the research on automating CS has centered on text mining techniques. These are explored in the context of developing models based on machine learning algorithms to ease the task of selecting the relevant studies from the results of the study search [29].

2.3 SVM Based Citation Screening Studies

In the field of automation of CS, the SVM approach has been widely used since it was proposed by Aphinyanaphongs and Alferis [2]. Cohen et al. have since published a number of studies based on different TREC and other datasets, which show that the SVMs record acceptable performances, usually a recall performance of 95% and over [9–11, 13]; and are used also to track newly published articles relevant to the same study [12]. Wallace et al. have also published a series of articles using different datasets, some of which show the ability of the SVM to separate well non—relevant articles from relevant ones [31, 33, 34]. They have since proposed a CS system — ABSTRACKR — using SVM and an active learning algorithm [34]. Other studies that have published SVM based classifiers within the CS context are [1, 6, 20, 21, 35]. A comprehensive review of these studies and more is presented in [30].

SVM has been widely used in text mining studies about screening citations automatically during SR but the studies are devoid of information about the complexity of the proposed models or about whether they are statistically better than other possible models. Though, all of the studies selected their best models through cross validation. Cross validation is another way to eliminate random performance and establish a reliable predictive performance of a model. In cross validation, the whole input is divided into equal sized subsets, the model will be trained on all but one subset and tested using the one left out [24]. This process is continued until each part has been used for testing. The results of the different runs are then averaged to get the mean performance of the model. We note that cross-validation by itself does not take into account the complexity of the model.

3 METHODS

We ran experiments to show the importance of providing details that characterise the complexity and statistical validity of computational models. We developed SVM models with word2vec (using average word vectors) representations, using the 15 reviews from [13].

The vector space model represents (embeds) words in a continuous vector space where words closer together are adjudged to share semantic meaning more than those farther away [4]. Word2vec is a predictive model for learning word embedding from raw text by first creating a vocabulary from the training text data and then learning the vector representation of words incorporating an understanding of when and how often words are used together in the representation [26]. The average word2vec incorporates the average of each word over the given corpus.

We retrieved our version of the data from the TREC 2004 raw data from http://skynet.ohsu.edu/trec-gen/data/2004/ after unsuccessful attempts to get a copy of those used by earlier researchers. For unknown reasons we could not retrieve exact counts for each of the reviews as reported in [13]. The missing data was directly retrieved from the 'pubmed' database. The studies and the proportion of negative and positive examples in each study is presented in Table 1.

To generate the word2vec representation, we tokenized the corpus and removed stopwords with the Stanford's nltk package. We then trained a word to vectors model with the aid of the Word2Vec method in the genism package. This model is used to transform the corpus to 'average word feature'. There was no stemming in the feature preparations. We reduce the dimensionality of the resulting sparse vectors using the χ^2 method in the sklearn's model selection routine to select top features that were found to be significant at 0.05 α level as reported in [13]. The number of total features and top features retained is shown in Table 2. We split each review corpus into training and testing datasets in the ratio 70:30 respectively using the train_test_split method with seed value of 37. The seed values were chosen randomly for the purpose of the study and to ensure reproduction. None of the previous studies reported their seed values. This ratio was changed to 80:20 in order to increase the training set in corpus where the data size was deemed small — usually below 1000. Table 4 shows the values used for the individual review training and testing sets.

Information necessary for the reproducibility of this experiment is provided below; software environment information is shown in Table 3.

- Initial dataset shuffle seed: 29
- Train-test split seed: 37, 71, 21, 61, 55
- SVM parameters:
  - Gamma: auto
  - C: 1, 10, 100, 1000, 10000
  - Kernel: Linear
  - Model random state: 37, 71, 21, 61, 55
  - Sample weight: 1:4
  - Class weight: balanced
- Word2Vec model
  - Features: as in Table 2.
  - minimum word count: 10
  - context window: 15

Table 1: Number of Retrieved Documents per Review

Review	Retrieved corpus size	Negative samples	Positive samples
ACEinhibitor	2544	2503	41
ADHD	851	831	20
Antihistamines	310	294	16
AtypicalAntipsychotics	1120	748	146
BetaBlockers	2072	1897	42
CalciumChannelBlockers	1218	1118	100
Estrogens	368	288	80
NSAIDs	393	352	41
Opioids	1915	1900	15
OralHypoglycemics	503	367	136
ProtonPumpInhibitors	1333	1282	51
SkeletalMuscleRelaxants	1643	1634	9
Statins	3465	3380	85
Triptans	671	647	24
UrinaryIncontinence	327	287	40

We trained a battery of 15 SVM models with the chosen models using stratified 5x2 folds cross validation. The dataset is split with different seed values on each run to ensure randomization. The recall, precision, accuracy and number of support vectors were accumulated and averaged.

- Recall is the fraction of correctly classified positive examples by the total positive examples in the whole corpus [34].
$$recall = \frac{tp}{tp + fn}$$
- Precision is the ratio of actual positive examples and the total positive prediction [34].
$$recall = \frac{tp}{tp + fp}$$
- Accuracy is the fraction of the total correct negative and correct positive prediction by corpus size [34].
$$recall = \frac{tp + tn}{tp + fp + tn + fn}$$

where,
tp →true positive fp →false positive
tn →true negative fn →false negative

In CS for SR, full recall of all relevant studies is the primary target. Thus, we chose the models with highest recall for the positive class. The results for the models given for the 15 reviews are shown in Table 4. The training and testing data sizes presented in the tables are the average over five runs. Similarly, the performance metrics — recall, accuracy and precision, are also the mean and standard deviation values over the five runs. Similarly, the performance metrics âĂŞ recall, accuracy and precision, are also the mean and

Table 2: Top Selected Features

Review	Feature size	Selected top features
ACEinhibitor	5754	210
ADHD	3591	80
Antihistamines	2105	29
AtypicalAntipsychotics	4131	381
BetaBlockers	5567	194
CalciumChannelBlockers	4111	329
Estrogens	2489	233
NSAIDs	2409	242
Opioids	5512	55
OralHypoglycemics	2759	234
ProtonPumpInhibitors	3942	206
SkeletalMuscleRelaxants	5835	11
Statins	7240	467
Triptans	3035	121
UrinaryIncontinence	2315	215

standard deviation values over the five runs. Apart from the usual recall and precision metrics we also show the mean and standard deviation of the number of support vectors that each of the models used in making its classification judgements — this characterises the complexity of the SVM classifiers.

4 RESULTS AND DISCUSSION

CS automation in SR is one of the software engineering fields where machine learning based techniques are currently being applied, thus its choice for this study. We chose to experiment with the linear kernel because it is simpler than the non-linear kernels. We also experimented with the binary, term frequency (tf), term frequency-inverse document frequency (tfidf) and word2vec features.

The word2vec features showed better performance with the linear kernel comparable to what is obtainable with other feature representations and nonlinear kernels. Following Occam's principle, if two models exhibit similar performance, the least complex should be chosen [14, 15]. Thus, our choice to experiment with the nonlinear kernel and word2vec features.

Viewing complexity as illustrated in section 1, what translates to complexity in each model differs. In this study, we illustrate with SVM models where complexity is characterised by the number of support vectors involved in the SVM classifier and is controlled through its hyper-parameters — C, gamma and kernel type

Table 4 shows that the linear kernel SVM models have relatively high recall performance but the number of support vectors is generally high, above 80% of the negative examples of the training dataset in 11 of the studies and 30% to 75% of the positive examples in all the reviews. In support vector models, the number of support vectors is indicative of the statistical validity and complexity of the models. We note that the number of support vectors reduces as the value of 'C' increases (i.e. this is the weight of the complexity penalty in the optimisation of the SVM) in the models.

The statistical theory of the SVM is based on the assumption that the algorithm uses as few support vectors as possible to make its decision [28]. This is the underlying reason for the reported advantage of the SVM algorithm - it is robust to small sample sizes or situations where the number of features is more than the number of samples because it needs only a few of the samples as support vectors [3, 19]. Ideally, we would expect a well optimized SVM model to use at most between 2%-5% of its total training data vectors (and preferably much less than 2% in the case of large volumes of data) as support vectors.

The fact that we find typically many more support vectors in our SVM classifiers may mean that in our case, the SVM optimisation is complicated and slow, which eventually leads to an early stop of the optimisers before achieving any significant optimisation.

Consequently, the statistical validity of these results is likely to be relatively limited, or in other words the likely error bounds are large and the likelihood of wrong classifications is also relatively high.

In the course of this study, we conducted similar experiment for the term frequency (tf), binary and the term frequency—inverse document frequency (tfidf) feature representations as well. However, we found that the average word vector based SVM classification lead to better performance results without requiring further preprocessing of the data and we chose these simpler approaches for the work presented in this paper. Also, we did not optimize the models beyond choosing a set of 'C' values and a set of kernel options since this was sufficient to explore the issue of statistical validity and complexity of data models that we address in this paper.

Ordinarily, only one parameter of the machine learning model is reported in most studies. Here we explore the potential of models considering several parameters before optimizing the result of the best. We used average word vector for feature representation modelled by the linear kernel SVMs for each review topic. Taking

Table 3: Software information

S/N	Software and packages	Version
1	Python	2.7.12 64bit
2	Ipython	5.1.0
3	Scipy	0.18.1
4	Numpy	1.11.3
5	Sklearn	0.18.1
6	Pandas	0.19.2
7	NLTK	3.2.2
8	Gensim	1.0.1
9	Matplotlib	1.5.3

Table 4: Word2Vec Linear Kernel (W2V-L)

Review	Train/Test size		Mean Performance (5x2 folds CV)			Support vectors		configuration
	neg	pos	precision	recall	accuracy	neg	pos	parameters
ACEinhibitor	1252	21	0.07 ± 0.02	0.94 ± 0.04	0.78 ± 0.05	595 ± 88	7 ± 1	linear, 1.0
ADHD	415	10	0.08 ± 0.01	0.93 ± 0.09	0.75 ± 0.01	251 ± 43	4 ± 1	linear, 1.0
Antihistamines	147	8	0.06 ± 0.00	0.90 ± 0.12	0.22 ± 0.06	141 ± 6	4 ± 1	linear, 40.0
AtypicalAntipsychotics	487	73	0.17 ± 0.02	0.92 ± 0.05	0.40 ± 0.09	420 ± 43	25 ± 4	linear, 1000
BetaBlockers	1015	21	0.05 ± 0.01	0.89 ± 0.07	0.63 ± 0.05	675 ± 68	8 ± 2	linear, 1.0
CalciumChannelBlockers	559	50	0.12 ± 0.01	0.92 ± 0.06	0.45 ± 0.07	477 ± 45	20 ± 2	linear, 100
Estrogens	144	40	0.32 ± 0.01	0.92 ± 0.06	0.56 ± 0.03	121 ± 8	12 ± 2	linear, 1000
NSAIDs	176	21	0.15 ± 0.02	1.00 ± 0.00	0.38 ± 0.06	157 ± 5	5 ± 1	linear, 1.0
Opioids	950	8	0.03 ± 0.00	0.81 ± 0.11	0.76 ± 0.05	482 ± 44	4 ± 1	linear, 1.0
OralHypoglycemics	184	68	0.28 ± 0.01	0.98 ± 0.02	0.30 ± 0.02	182 ± 2	33 ± 2	linear, 10000
ProtonPumpInhibitors	641	26	0.06 ± 0.01	0.91 ± 0.07	0.47 ± 0.09	542 ± 56	9 ± 1	linear, 1.0
SkeletalMuscleRelaxants	817	5	0.01 ± 0.01	0.66 ± 0.23	0.53 ± 0.11	610 ± 83	4 ± 0	linear, 1.0
Statins	1690	43	0.05 ± 0.00	0.92 ± 0.04	0.56 ± 0.06	1250 ± 83	14 ± 2	linear, 1.0
Triptans	324	12	0.06 ± 0.00	0.97 ± 0.06	0.44 ± 0.07	286 ± 16	4 ± 1	linear, 1.0
UrinaryIncontinence	143	20	0.20 ± 0.03	0.93 ± 0.07	0.53 ± 0.11	122 ± 15	6 ± 1	linear, 100

statistical validity into account and the principles of model selection — Occam's razor, MDL and Kolmogorov complexity — the linear kernel models should be preferred.

In SVM, the higher the number of support vectors, the more complex the model is, the higher the possibility of misclassification error and over-fitting. According to [5], learning in models is a function of the hypothesis, representation and optimization. There is hardly any optimization done by the model, when (almost) all the dataset acts as support vectors in an SVM model. Such models are almost equivalent of a nearest neighbour classifier using all available training data. Consequently, the statistical validity of SVM models, where a large fraction of the training data constitute support vectors, is comparable to the statistical validity of nearest neighbour classifiers based on the full training data.

5 VALIDITY THREATS

We present here, only the result of the linear kernel SVM, we have not considered the result of the nonlinear kernel SVMs in this study. The performance of the SVMs reported in this study is particular to the models generated by the datasets used. It should be noted that the sample sizes used are quite small with considerably imbalanced classes. We did not make any extra attempt to improve the performance of the SVM models beyond the feature types used and tuning of the parameters. Additional tuning may have changed the outcome of the study.

6 CONCLUSIONS

In this study, we developed SVM models with linear kernels to automatically screen citations for inclusion and exclusion in an SR scenario based on 15 DERP SR dataset to explore complexity and statistical validity issues surrounding machine learning models. Apart from reporting the performance — recall, precision and accuracy — results 5x2 cross validation, we also explore the number of support vectors for each model. The models show relatively acceptable recall performance which is the target in SR but the support vectors are relatively high. This may raise suspicion about the statistical validity and complexity of the models.

This work has shown that, in addition to performance results, information that reflects how well a model complies with the principles of its underlying theory and complexity are also important to be provided in study reports. This will give the reader a better understanding of the model and more grounds for comparability and improvement. The specific complexity or statistical validity details differ from model to model, we only illustrate this with SVMs in this paper.

7 ACKNOWLEDGMENT

B.K. Olorisade thank National Information Technology Development Agency (NITDA) for providing funds to sponsor his PhD research.

REFERENCES

[1] JJ García Adeva, JM Pikatza Atxa, M Ubeda Carrillo, and E Ansuategi Zengotitabengoa. 2014. Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications* 41, 4 (2014), 1498–1508.

[2] Yindalon Aphinyanaphongs, Ioannis Tsamardinos, Alexander Statnikov, Douglas Hardin, and Constantin F Aliferis. 2005. Text categorization models for high-quality article retrieval in internal medicine. *J. Am. Med. Informatics Assoc.* 12, 2 (2005), 207–216.

[3] Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. 2010. A Review of Machine Learning Algorithms for Text-Documents Classification. *J. Adv. Inf.*

Technol. 1, 1 (2010), 4–20. https://doi.org/10.4304/jait.1.1.4-20

[4] Marco Baroni, Georgiana Dinu, and German Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist.* (2014), 238–247. https://doi.org/10.3115/v1/P14-1023

[5] Andrew Barron, Jorma Rissanen, and Bin Yu. 1998. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory* 44, 6 (1998), 2743–2760.

[6] Tanja Bekhuis and Dina Demner-Fushman. 2010. Towards automating the initial screening phase of a systematic review. *Stud. Health Technol. Inform.* 160, PART 1 (2010), 146–150. https://doi.org/10.3233/978-1-60750-588-4-146

[7] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. 1987. Occam's razor. *Information processing letters* 24, 6 (1987), 377–380.

[8] Gregory J Chaitin. 1969. On the length of programs for computing finite binary sequences: statistical considerations. *Journal of the ACM (JACM)* 16, 1 (1969), 145–159.

[9] Aaron M Cohen. 2006. An effective general purpose approach for automated biomedical document classification. In *AMIA Annual Symposium Proceedings*, Vol. 2006. American Medical Informatics Association, 161.

[10] Aaron M Cohen. 2008. Optimizing feature representation for automated systematic review work prioritization. In *AMIA annual symposium proceedings*, Vol. 2008. American Medical Informatics Association, 121.

[11] Aaron M Cohen, Kyle Ambert, and Marian McDonagh. 2009. Cross-topic learning for work prioritization in systematic review creation and update. *Journal of the American Medical Informatics Association* 16, 5 (2009), 690–704.

[12] Aaron M Cohen, Kyle Ambert, and Marian McDonagh. 2012. Studying the potential impact of automated document classification on scheduling a systematic review update. *BMC medical informatics and decision making* 12, 1 (2012), 33.

[13] Aaron M Cohen, William R Hersh, K Peterson, and Po-Yin Yen. 2006. Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association* 13, 2 (2006), 206–219.

[14] Pedro Domingos. 1999. The role of Occam's razor in knowledge discovery. *Data mining and knowledge discovery* 3, 4 (1999), 409–425.

[15] Pedro Domingos. 2012. A few useful things to know about machine learning. *Commun. ACM* 55, 10 (2012), 78–87.

[16] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin.

[17] Peter Grünwald. 2000. Model selection based on minimum description length. *Journal of Mathematical Psychology* 44, 1 (2000), 133–152.

[18] Mark H Hansen and Bin Yu. 2001. Model selection and the principle of minimum description length. *J. Amer. Statist. Assoc.* 96, 454 (2001), 746–774.

[19] M Ikonomakis, S Kotsiantis, and V Tampakas. 2005. Text classification using machine learning techniques. *WSEAS transactions on computers* 4, 8 (2005), 966–974.

[20] Seunghee Kim and Jinwook Choi. 2012. Improving the performance of text categorization models used for the selection of high quality articles. *Healthcare informatics research* 18, 1 (2012), 18–28.

[21] Seunghee Kim and Jinwook Choi. 2014. An SVM-based high-quality article classifier for systematic reviews. *Journal of biomedical informatics* 47 (2014), 153–159.

[22] Barbara Ann Kitchenham, David Budgen, and Pearl Brereton. 2015. *Evidence-Based Software engineering and systematic reviews*. Vol. 4. CRC Press.

[23] Andrei N Kolmogorov. 1965. Three approaches to the quantitative definition ofinformation'. *Problems of information transmission* 1, 1 (1965), 1–7.

[24] Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. 2006. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review* 26, 3 (2006), 159–190.

[25] Christopher Marshall and Pearl Brereton. 2013. Tools to support systematic literature reviews in software engineering: A mapping study. In *Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on*. IEEE, 296–299.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[27] Volker Nannen. 2010. A short introduction to model selection, Kolmogorov complexity and Minimum Description Length (MDL). *arXiv preprint arXiv:1005.2364* (2010).

[28] Christianini Nello and Shawe-Taylor John. 2000. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*.

[29] Babatunde Kazeem Olorisade, Ed de Quincey, Pearl Brereton, and Peter Andras. 2016. A critical analysis of studies that address the use of text mining for citation screening in systematic reviews. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*. ACM, 14.

[30] Alison O'Mara-Eves, James Thomas, John McNaught, Makoto Miwa, and Sophia Ananiadou. 2015. Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Systematic reviews* 4, 1 (2015), 5.

[31] Kevin Small, Byron Wallace, Thomas Trikalinos, and Carla E Brodley. 2011. The constrained weight space svm: learning with ranked features. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 865–872.

[32] Ray J Solomonoff. 1964. A formal theory of inductive inference. Part I. *Information and control* 7, 1 (1964), 1–22.

[33] Byron C Wallace, Kevin Small, Carla E Brodley, Joseph Lau, and Thomas A Trikalinos. 2010. Modeling annotation time to reduce workload in comparative effectiveness reviews. In *Proceedings of the 1st ACM International Health Informatics Symposium*. ACM, 28–35.

[34] Byron C Wallace, Kevin Small, Carla E Brodley, Joseph Lau, and Thomas A Trikalinos. 2012. Deploying an interactive machine learning system in an evidence-based practice center: abstrackr. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*. ACM, 819–824.

[35] Wei Yu, Melinda Clyne, Siobhan M Dolan, Ajay Yesupriya, Anja Wulf, Tiebin Liu, Muin J Khoury, and Marta Gwinn. 2008. GAPscreener: an automatic tool for screening human genetic association literature in PubMed using the support vector machine technique. *BMC bioinformatics* 9, 1 (2008), 205.