Manuscript drafts

Gerbrich Ferdinands

03 May, 2020

Keywords: Human-Machine interaction, active learning, systematic reviews, text classification

Introduction

Methods

Results

Discussion

Introduction

Systematic reviews are top of the bill in building evidence in research. A systematic review brings together all studies relevant to answer a specific research question [1]. Systematic reviews inform practice and policy [2] and are key in developing clinical guidelines [3].

However, systematic reviews are costly because they involve the manual screening of thousands of titles and abstracts to identify publications relevant to answering the research question. An experienced reviewer takes on average 30 seconds to screen one title and abstract, whereas an inexperienced reviewer takes even longer [4]. Conducting a systematic review typically requires over a year of work by a team of researchers [5]. Nevertheless, systematic reviewers are often bound to a limited budget and timeframe. Currently, the demand for systematic reviews exceeds the available time and resources by far [6]. Especially when the need for guidelines is urgent - such as in the context of the current COVID-19 crisis - it is almost impossible to provide a review that is both timely and comprehensive. To ensure a timely review, reducing workload in systematic reviews is essential.

With advances in Artificial Intelligence (AI), there has been wide interest in tools to reduce workload in systematic reviews [7]. Various learning models have been proposed, aiming to predict whether a given publication is relevant or irrelevant to the systematic review. Findings suggest that such models potentially reduce workload with 30-70% at the cost of losing 5% of relevant publications, i.e. 95% recall [8].

A well-established approach in increasing efficiency in title and abstract screening is screening prioritization [9,10]. In screening prioritization, the learning model reorders publications to be screened by their likeliness to be relevant. The model presents the reviewer with the publications which are most likely to be relevant first, thereby expediting the process of finding all of the relevant publications. Such an approach allows for substantial time-savings in the screening process. Reviewing relevant publications early facilitates a faster transition of those publications to the next steps in the review process [9]. ~~Additionally, although outside the scope of the current study, active learning models have the potential to reduce the number of publications needed to screen when combined with some sort of stopping criterion [11].~~

Recent studies have demonstrated the effectiveness of screening prioritization by means of active learning models [4,11–15]. Active learning is when the model can iteratively improve its predictions by allowing the model to choose the data from which it can learn [16]. Active learning has proven to be an efficient strategy in large datasets where labels are scarce, which makes title and abstract screening an ideal candidate for such models. When applied in screening prioritization, the reviewer screens publications that are presented by an active learning model. Subsequently, the active learning model learns from the reviewers' decision ('relevant', 'irrelevant') and uses this knowledge in selecting the next publication to be screened by the reviewer.

Although the application of learning models in reducing workload of systematic reviews has been extensively studied [4,11–13], the complex nature of the field is making it difficult to draw overarching conclusions about best practice [8]. First, whilst previous studies have evaluated models in many forms and shapes [4,12,12,13], all models adopt the same classification technique. Even though a plethora of classification techniques exists [17], no study in this area has made a comparison between models adopting different classification techniques. Second, the lack of model replication on reviews from varying research contexts makes it impossible to draw conclusions about the general effectiveness of active learning models [8,18]. As far as known to the authors, Miwa et al [13] were the only researchers to make a direct comparison between systematic reviews from different research areas, namely the social and the medical sciences. They found that active learning was more difficult on data from the social sciences due to the nature of the different vocabularies used. Therefore, it is of interest to evaluate model performance across different research contexts.

Taken together, evaluations of active learning models are required across different classification techniques (1) and review contexts (2). The current study aims to address these issues by answering the following research questions:

RQ1 What is the performance of active learning models across different classification techniques?

RQ2 Does performance of active learning models differ across systematic reviews from different research areas?

The purpose of the current paper is to increase the evidence base on active learning models for reducing workload in title and abstract screening in systematic reviews. Combining latest insights from this area, we propose seven different active learning models for the purpose of identifying relevant publications in systematic review datasets. The models were chosen to maximize the number of identified relevant publications, while minimizing the number of publications needed to screen. Working towards a general consensus in this emerging field, model performance was assessed by conducting a retrospective simulation on six systematic review datasets. Datasets were collected from the fields of medicine, software engineering, psychology, behavioural public administration, and virology to assess generalizability of the models across research contexts. The models, datasets and simulations are implemented in a pipeline of active learning for screening prioritization, called `ASReview` [19]. `ASReview` is an open source and generic tool such that users can adapt and add modules as they like, encouraging fellow researchers to replicate findings from previous studies. All scripts and data used are openly published to facilitate usability and acceptability of AI-assisted title and abstract screening in the field of systematic review.

The remaining part of this paper is organized as follows. The Methods section elaborates on the setup of the current study, starting with a task description, followed the Technical Details section which will cover the workings of active learning models for study selection in systematic reviews on a conceptual level. The results section reports ... The discusssion section summarises the findings, comments on them and summarises the main findings, discusses discuss limitations, draw conclusion and

Methods

Task Description

The screening process of a systematic review starts with all publications obtained in the search. The task is to identify which of these publications are relevant, by screening them at the title and abstract level. In active learning for screening prioritization, the screening process proceeds as follows:

- Start with the set of all unlabeled (titles and) abstracts, $\mathcal{U}$.
- The reviewer provides a label for a few records $x \in \mathcal{U}$, creating an set of labeled titles and abstracts $\mathcal{L}$. The label can be either relevant x_R or irrelevant x_I.
- The active learning cycle starts:
    1. A classifier is trained on the labeled titles and abstracts, $C = \texttt{train}(\mathcal{L})$
    2. The classifier predicts labels for all unlabelled titles and abstracts, $C(\mathcal{U})$
    3. Based on the predictions by C, the model selects the most relevant title and abstract $x^* \in \mathcal{U}$
    4. The model asks the reviewer to screen this title and abstract, $x^*_?$
    5. The reviewer screens the title and abstract and provides a label, x^*_R or x^*_I
    6. The labeled title and abstract is added to the training data, $x_{R \texttt{ xor } I} \in \mathcal{L}$
    7. Back to step 1.
- In this active learning cycle, the model can incrementally improve its predictions on the remaining unlabeled title and abstracts. The relevant titles and abstracts are identified as early in the process as possible. The reviewer and the model keep interacting until the reviewer decides to stop or until all title and abstracts have been labelled.

Technical details

A more detailed account of the active learning models is given in the following section. The structure and functions of the key components of the models will be introduced to clarify the choices made in the design of the current study.

Classification

To make predictions on the unlabeled publications, a classifier is trained on features from the set of previously labeled publications. A technique widely used in classification tasks is the Support Vector Machine (SVM). SVMs separate the data into classes by finding a multidimensional hyperplane [20,21]. SVMs have been proven to be effective in active learning models for screening prioritization [[12]; Miwa2014]. Moreover, SVMs are the currently the only classifier implemented in ready-to-use software tools implementing active learning for screening prioritization (Abstrackr [22], Colandr [23], FASTREAD [12], Rayyan [24], and RobotAnalyst [25]).

Whilst the performance of several classification techniques has been investigated in the AI-aided title-and-abstract screening field in general [26,27], the relatively new subfield of active learning for screening prioritization has not yet studied the performance of classifiers other than SVMs [4,12,12–15]. The current study aims to address this gap by exploring performance of three classifiers besides SVM:

- L2-regularized Logistic Regression (LR) models the probabilities describing the possible outcomes by a logistic function. The L2 penalty is imposed on the coefficients to reduce the number of features upon which the given solution is dependent [28].
- Naive Bayes (NB) is a supervised learning algorithm often used in text classification. Based on Bayes' Theorem, with the 'naive' assumption that all features are independent given the class value [29].
- Random Forests (RF) is a supervised learning algorithm where a large number of decision trees are fit on bootstrapped samples of the original data. All trees cast a vote on the class, which are aggregated into a class prediction for each instance [30].

These three classification techniques were selected because they are widely adopted methods in text classification [17]. Moreover, these techniques can be run on a personal computer as they require a relatively low amount of processing power.

Class imbalance problem

There are two classes in the dataset: relevant and irrelevant publications. Typically, the inclusion rate is low (<10%) as only a fraction of the publications belong to the relevant class. This poses a problem for training a classifier as there are far fewer examples of relevant than irrelevant publications to train on [8]. Moreover, classifiers are well-suited to separate data into classes, but not to correctly identify one class [4]. Therefore, the class imbalance problem causes the classifier to miss relevant publications. This is evident in the case of a systematic review dataset where only one percent of publications are relevant. A model would achieve 99% accuracy when classifying all publications as irrelevant, even though none of the relevant papers would have been correctly identified.

Previous studies have addressed the class imbalance problem by rebalancing the training data in different ways [8]. To decrease the class imbalance in the training data, the models in the current study rebalance the training set by Dynamic Supersampling (DS). DS decreases the number of irrelevant publications in the training data, whereas the number of relevant publications are increased (by copy) such that the size of the training data remains the same. The ratio between relevant and irrelevant publications in the training data is not fixed, but dynamically updated and depends on the size of the training data, the total number of publications, and the ratio between the total number of labeled publications.

Word embeddings

To predict publication class, the classifier uses information from the publications in the dataset. Examples of such information are titles and abstracts. However, a model cannot make predictions from the titles and abstracts as they are; their textual content needs to be represented numerically. The textual information needs to be mapped to feature vectors. This process of numerically representing textual content is called 'word embeddings'.

A classical example of word embeddings is a 'bag of words' (bow) representation. For each text in the data set, the number of occurrences of each word is stored. This leads to n features, where n is the number of distinct words in the texts [28]. The bag-of-words method is simplistic and will highly value often occurring but otherwise meaningless words such as "and". A more sophisticated approach is Term-Frequency Inverse Document Frequency (TF-IDF). TF-IDF circumvents this problem by adjusting the term frequency in a text with the inverse document frequency, the frequency of a given word in the entire data set [31]. A downside of TF-IDF and other bow methods is that they do not take into account the ordering of words, thereby ignoring semantics. An example of an approach that aims to overcome this weakness is Doc2Vec (D2V), capable of grasping the relations between words by learning to predict the words in the texts [32].

Miwa et al. found that active learning was more difficult on data from the social sciences compared to data from the medical sciences and were able to link this difficulty to a natural difference in text complexity between these research areas [13]. As the study by Miwa et al. adopted a bow approach [13], we hypothesize that a more sophisticated word embedding strategy has the potential to bridge the performance gap between these research areas.

Query strategy

The active learning model can adopt different strategies in selecting the next publication to be screened by the reviewer. A strategy mentioned before is selecting the publication with the highest probability of being relevant. In the active learning literature this is referred to as certainty-based active learning [16]. Another well-known strategy is uncertainty-based active learning, where the instances that will be presented next will be those instances on which the model's classifications are the least certain, i.e. close to 0.5 probability [16]. Traditionally, this strategy trains the most accurate model because the model can learn the most from instances it is uncertain about. However, a study comparing performance of both strategies in detecting relevant publications found that the accuracy gain of uncertainty-based screening was not significant [13].

Certainty-based active learning is the preferred strategy for the task at hand. Firstly, this strategy is far better suited to the goal of prioritizing relevant publications compared to uncertainty-based active learning, in which the publications are prioritized that the model is most uncertain about. Secondly, certainty-based active learning is far better equipped at dealing with imbalanced data in active learning [33].

Models

The seven models consist of the components described in the Technical Details section, adopting four different classification techniques and two different word embeddings approaches:

First, four models combining every classifier with TF-IDF word embeddings were investigated:

- SVM + TF-IDF
- NB + TF-IDF
- RF + TF-IDF
- SVM + TF-IDF

Second, the classifiers were combined with Doc2Vec word embeddings, leading to the following three models:[1]

- SVM + Doc2Vec
- RF + Doc2Vec
- SVM + Doc2Vec

[1]The combination NB + D2V could not be tested because the Multinomial Naive Bayes classifier (sklearn) can only handle a feature matrix with positive values, whereas the doc2vec word embeddings approach (gensim) produces a feature matrix that can also contain negative values.

Simulation study

For each of the seven models, performance was evaluated by simulating the model on the screening process of six systematic reviews. Performance of the seven models was evaluated by simulating their behaviour in the screening process of six systematic review datasets. Put differently, 42 simulations were carried out. For every model-dataset combination, hyperparameters were optimized[2]. To account for variance, every simulation was repeated for 15 trials. Simulations were run using `ASReview`'s simulation mode [19]. There was no need for a human reviewer as the model could query the labels in the data instead.

Every simulation started with an initial training set of one relevant and one irrelevant publication to represent a 'worst case scenario' where the reviewer has minimal prior knowledge on the publications in the data. To account for bias, the initial training set was randomly sampled from the dataset for every of the 15 runs. Although varying over runs, the initial training sets were kept constant over datasets to allow for a direct comparison of models within datasets. A seed value was set to ensure reproducibility The classifier was retrained every time after a publication had been labeled. The simulation ended after all publications in the dataset had been labeled.

Datasets

The models were simulated on a convenience sample of six systematic review datasets. The data selection process was driven by two factors. Firstly, datasets were selected based on their background, given the need for datasets from diverse research areas. Secondly, datasets were selected by their availability, given the limited timespan of the current project. The datasets were retrieved from a collection of open systematic review datasets to be used for text mining purposes[3].

Three out of six datasets originated from the *medical sciences*: Ace, Wilson, and Virus. The Wilson dataset [34] is on a review on effectiveness and safety of treatments of Wilson Disease, a rare genetic disorder of copper metabolism [35]. The Ace dataset contains publications on the efficacy of Angiotensin-converting enzyme (ACE) inhibitors, a drug treatment for heart disease [36]. The Virus dataset is from a systematic review on studies that performed viral Metagenomic Next-Generation Sequencing (mNGS) in farm animals [37]. From the field of *software engineering*, the Software dataset contains publications from a review on fault prediction in source code [38]. The Nudging dataset [39] belongs to a systematic review on nudging healthcare professionals [40], stemming from the area of *behavioural public administration*, The PTSD dataset contains publications from the field of *psychology*. The corresponding systematic review is on studies applying latent trajectory analyses on posttraumatic stress after exposure to trauma [41]. Of these six datasets, Ace, and Software have been used for model simulations in previous studies on AI-aided title and abstract screening, respectively [36] and [12].

Data were preprocessed from their original source into a test dataset, containing title and abstract of the publications obtained in the initial search. Candidate studies with missing abstracts and duplicate instances were removed from the data. Test datasets were labelled to indicate which candidate studies were included in the systematic review, thereby indicating relevant publications. All test datasets consisted of thousands of candidate studies, of which only only a fraction was deemed relevant to the systematic review. For four of the six datasets inclusion rates were centered around 1-2 percent. For the remaining two datasets, the inclusion rate was about 5 percent (Table 1).

Table 1: Statistics on datasets from original systematic reviews.

```
## Warning in kable_styling(., full_width = TRUE): Please specify format in
## kable. kableExtra can customize either HTML or LaTeX outputs. See https://
## haozhu233.github.io/kableExtra/ for details.
```

[2]see the Appendix for more information

[3]https://github.com/asreview/systematic-review-datasets

```
## Warning in add_header_above(., c("", `Original study` = 3, `Test collection`
## = 3)): Please specify format in kable. kableExtra can customize either HTML or
## LaTeX outputs. See https://haozhu233.github.io/kableExtra/ for details.
```

Dataset	Candidate studies	Relevant studies	Inclusion rate (%)	Candidate studies	Relevant studies	Inclusion rat
Ace	2544	41	1.61	2235	41	
Nudging	2006	100	4.99	1847	100	
PTSD	6185	38	0.61	5031	38	
Software	8911	104	1.17	8896	104	
Virus	2481	120	4.84	2304	114	
Wilson	3453	26	0.75	2333	23	

Evaluating performance

Model performance assessed by three different measures, Work Saved over Sampling (WSS), Relevant References Found (RRF), and Time to Discovery (TTD). Furthermore, model performance was visualized by plotting recall curves. Results were averaged over 15 trials for every simulation.

WSS indicates the reduction in publications needed to be screened, at a given level of recall [36]. Typically measured at a recall level of 0.95 [36], WSS@95 yields an estimate of the amount of work that can be saved at the cost of failing to identify 5% of relevant publications. In the current study, WSS is computed at 0.95 and 1.00 recall level. RRF statistics are computed at 10%, representing the proportion of relevant studies that are after screening 10% of all publications.

Both RRF and WSS are sensitive to random effects as these statistics are strongly dependent on the position of the cutoff value. Moreover, WSS makes assumptions about acceptable recall levels whereas this level might depend on the research question at hand [8].

A statistic that is not dependent on some arbitrary cutoff value is the TTD, which is the average number of publications that needed to be screened to find a relevant publication, divided by the total number of publications in the data. The TTD is proportional to the area above the recall curve.

Plotting recall as a function of the number of screened publications offers insight in model performance throughout the screening process [12,14]. The curves give information in two directions. On the one hand they display the number of publications that need to be screened to achieve a certain level of recall (WSS), but on the other hand they present how many relevant publications are identified after screening a certain proportion of all publications (RRF).

Simulations were run in ASReview, version 0.9.3 [19]. Analyses were carried out using R, version 3.6.1 [42]. All datasets accompanying the systematic reviews are openly published. This study was approved by the Ethics Committee of the Faculty of Social and Behavioural Sciences of Utrecht University, filed as an amendment under study 20-104. Due to their large number, the simulations were carried out on Cartesius, the Dutch national supercomputer.

–> –>

Results

For each of the seven models, performance was evaluated by simulating the model on the screening process of six systematic reviews.

Results were averaged over 15 trials for every simulation. Performance of the seven models was evaluated by simulating their behaviour in the screening process of six systematic review datasets. All results were

To allow for a direct comparison between models, their performance is evaluated within datasets. Every model*data combination (42) was repeated for 15 times. 15 runs started with the same initial publications..

RQ1 What is the performance of active learning models across different classification techniques?

Evaluation on the Ace dataset

Recall curves present recall after averaged over 15 trials with a confidence region ...

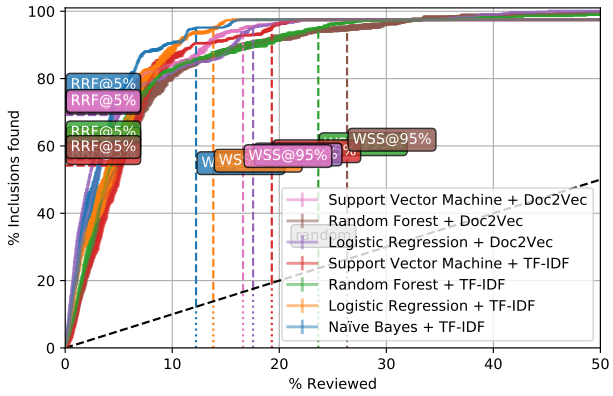

Figure 1: performance on Ace dataset

Evaluation on the Nudging dataset

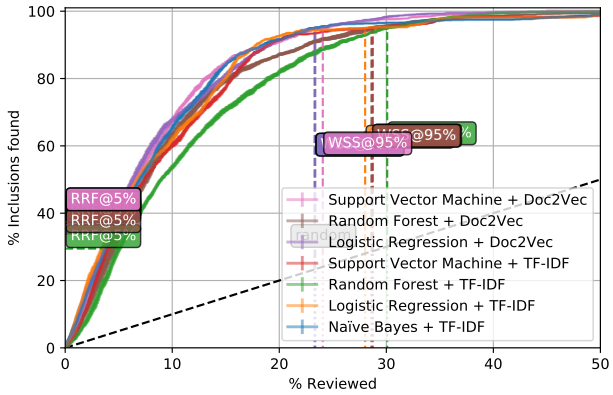

Figure 2: Performance on Nudging dataset

Evaluation on the PTSD dataset

simulation on svm + tfidf is not finished yet

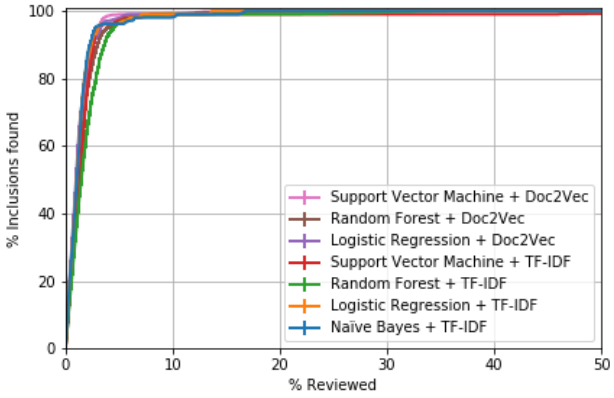

Figure 3: Performance on Software dataset

Evaluation on the Software dataset

Evaluation on the Virus dataset

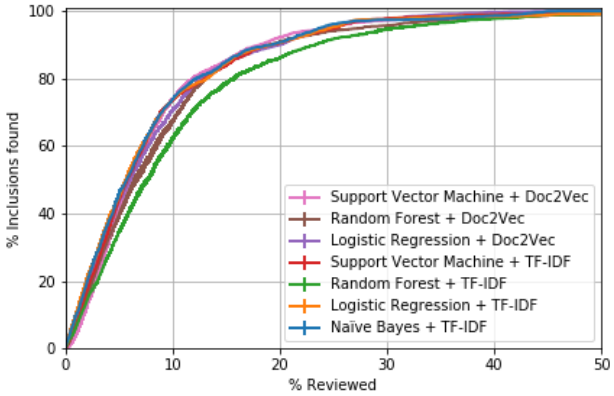

Figure 4: Performance on Virus dataset

Evaluation on the Wilson dataset

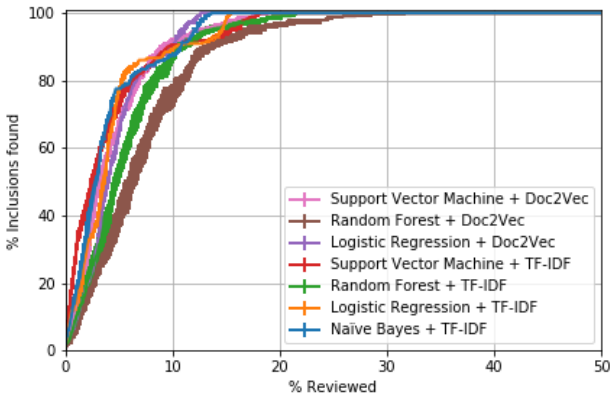

Figure 5: Performance on Wilson dataset

Overall Comparison

- models in general?
- which models perform better in which contexts?

	Ace		
	WSS@95	RRF@10	TTD
NB + TF-IDF	82.94	90.67	4.93
LR + D2V	77.45	81.83	5.49
LR + TF-IDF	81.16	88.50	5.99
RF + D2V	68.67	80.83	7.21
RF + TF-IDF	71.35	82.33	6.89
SVM + D2V	78.39	84.17	6.14
SVM + TF-IDF	75.87	86.33	7.18

Discussion

- we look for final inclusions but we screen only the abstracts (do they satisfy the imformation need (blake (page 19 omara et evs)))

future research: - stopping rule is not discussed - computation/retraining time

strengths:

- open data
- different research areas
- different models on same dataset
- different datasets on same model

limitations

future research - all models save time, difficult to distinguish performance over datasets, especially when applied on a dataset of which no prior information is known (e.g. inclusions isn't known in practice). Perhaps go for other criteria like the fastest model, replicate study with computation time?

Simulating the title and abstract screening process, models are evaluted on their capability/speed of detecting the final inclusions. However, in a manual SR these final inclusions are selected after reading the fulltext. Information the text mining tool does not have. To truly assess the added value of such a tool, models should be evaluated on their capability of detecting the abstract inclusions. Call for systematic reviewers to openly publish need for open data containing abstract inclusions, not only final inclusions!

Appendix A - list of definitions

Feature Extraction Strategies

split_ta = overall hyperparameter

TF-IDF

hyperparameters

```
ngram_max: int
        Can use up to ngrams up to ngram_max. For example in the case of
        ngram_max=2, monograms and bigrams could be used.
```

Doc2Vec Predicts words from context. Aims at capturing the relations between word (man-woman, king-queen). [32]. Using a neural network.

using Continuous Bag-of-Words (CBOW), Skip-Gram model, ... Word vector W and extra: document vector D, trained to predict words in the text.

From gensim [43].

```
    Arguments
    ---------
    vector_size: int
        Output size of the vector.
    epochs: int
        Number of epochs to train the doc2vec model.
    min_count: int
        Minimum number of occurences for a word in the corpus for it to
        be included in the model.
    workers: int
        Number of threads to train the model with.
    window: int
        Maximum distance over which word vectors influence each other.
    dm_concat: int
        Whether to concatenate word vectors or not.
        See paper for more detail.
    dm: int
        Model to use.
        0: Use distribute bag of words (DBOW).
        1: Use distributed memory (DM).
        2: Use both of the above with half the vector size and concatenate
        them.
    dbow_words: int
        Whether to train the word vectors using the skipgram metho
```

SBERT BERT-base model with mean-tokens pooling [44]

embeddingIdf This model averages the weighted word vectors of all the words in the text, in order to get a single feature vector for each text. The weights are provided by the inverse document frequencies

Models

Naive Bayes Naive Bayes assumes all features are independent given the class value. [29]

ASReview uses the `MultinomialNB` from the scikit-learn package [28], that implements the naive Bayes algorithm for multinomially distributed data. `nb`

Hyperparameters

- alpha - accounts for features not present in learning samples and prevents zero probabilities in further computations.

Random Forests A number of decision trees are fit on bootstrapped samples of the original data, [30] RandomForestClassifier from sklearn

Arguments ——— n_estimators: int Number of estimators. max_features: int Number of features in the model. class_weight: float Class weight of the inclusions. random_state: int, RandomState Set the random state of the RNG. """

Support Vector Machine Arguments ——— gamma: str Gamma parameter of the SVM model. class_weight: class_weight of the inclusions. C: C parameter of the SVM model. kernel: SVM kernel type. random_state: State of the RNG.

Logistic Regression

Dense Neural Network

Query Strategies

- Max - Choose the most likely samples to be included according to the model
- Uncertainty - choose the most uncertain samples according to the model (i.e. closest to 0.5 probability) [45]
- Random - randomly selects abstracts with no regard to model assigned probabilities.
- Cluster - Use clustering after feature extraction on the dataset. Then the highest probabilities within random clusters are sampled

The following combinations are simulated:

- cluster
- max
- cluster * random
- cluster * uncertainty
- max * cluster
- max * random
- max * uncertainty

Balance Strategies

amount of training data

- n_instances = number of papers queried each query
- n_queries = number of queries
- n_prior_included: 5
- n_prior_excluded:

Combinations

This leads to 119 combinations of configurations.

- Naive bayes only goes with tfidf feature extraction.
- For the feature extraction strategies we will focus on doc2vec and tfidf. (but will compute all 4)
- This leads to 3 * 7 * 4 * 3 + 1 * 7 * 1 * 3 = 273 combinations.

See appendix A for a table containing all 273 combinations.

Cross-validation

Should give an accurate estimate of maximum performance / future systematic reviews to be performed.

References

1. PRISMA-P Group, Moher D, Shamseer L, Clarke M, Ghersi D, Liberati A, et al. Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. Syst Rev. 2015;4:1.

2. Gough D, Elbourne D. Systematic Research Synthesis to Inform Policy, Practice and Democratic Debate. Social Policy and Society. Cambridge University Press; 2002;1:225–36.

3. Chalmers I. The lethal consequences of failing to make full use of all relevant evidence about the effects of medical treatments: The importance of systematic reviews. Treating individuals—from randomised trials to personalised medicine. Lancet; 2007. pp. 37–58.

4. Wallace BC, Trikalinos TA, Lau J, Brodley C, Schmid CH. Semi-automated screening of biomedical citations for systematic reviews. BMC Bioinformatics. 2010;11:55.

5. Borah R, Brown AW, Capers PL, Kaiser KA. Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the PROSPERO registry. BMJ Open. British Medical Journal Publishing Group; 2017;7:e012545.

6. Lau J. Editorial: Systematic review automation thematic series. Systematic Reviews. 2019;8:70.

7. Harrison H, Griffin SJ, Kuhn I, Usher-Smith JA. Software tools to support title and abstract screening for systematic reviews in healthcare: An evaluation. BMC Medical Research Methodology. 2020;20:7.

8. O'Mara-Eves A, Thomas J, McNaught J, Miwa M, Ananiadou S. Using text mining for study identification in systematic reviews: A systematic review of current approaches. Systematic Reviews. 2015;4:5.

9. Cohen AM, Ambert K, McDonagh M. Cross-Topic Learning for Work Prioritization in Systematic Review Creation and Update. J Am Med Inform Assoc. Oxford Academic; 2009;16:690–704.

10. Shemilt I, Simon A, Hollands GJ, Marteau TM, Ogilvie D, O'Mara-Eves A, et al. Pinpointing needles in giant haystacks: Use of text mining to reduce impractical screening workload in extremely large scoping reviews. Research Synthesis Methods. 2014;5:31–49.

11. Yu Z, Menzies T. FAST2: An intelligent assistant for finding relevant papers. Expert Systems with Applications. 2019;120:57–71.

12. Yu Z, Kraft NA, Menzies T. Finding better active learners for faster literature reviews. Empirical Software Engineering. Springer Science and Business Media LLC; 2018;23:3161–86.

13. Miwa M, Thomas J, O'Mara-Eves A, Ananiadou S. Reducing systematic review workload through certainty-based screening. Journal of Biomedical Informatics. 2014;51:242–53.

14. Cormack GV, Grossman MR. Evaluation of machine-learning protocols for technology-assisted review in electronic discovery. Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. Gold Coast, Queensland, Australia: Association for Computing Machinery; 2014. pp. 153–62.

15. Cormack GV, Grossman MR. Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review. 2015 [cited 2020 Apr 29]; Available from: http://arxiv.org/abs/1504.06868

16. Settles B. Active Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. 2012;6:1–114.

17. Aggarwal CC, Zhai C. A Survey of Text Classification Algorithms. In: Aggarwal CC, Zhai C, editors. Mining Text Data. Boston, MA: Springer US; 2012. pp. 163–222.

18. Marshall IJ, Johnson BT, Wang Z, Rajasekaran S, Wallace BC. Semi-Automated evidence synthesis in health psychology: Current methods and future prospects. Health Psychology Review. Routledge; 2020;14:145–58.

19. van de Schoot R, de Bruin J, Schram R, Zahedi P, Kramer B, Ferdinands G, et al. ASReview: Active learning for systematic reviews. Zenodo; 2020;

20. Tong S, Koller D. Support vector machine active learning with applications to text classification. Journal of machine learning research. 2001;2:45–66.

21. Kremer J, Steenstrup Pedersen K, Igel C. Active learning with support vector machines. WIREs Data Mining and Knowledge Discovery. 2014;4:313–26.

22. Wallace BC, Small K, Brodley CE, Lau J, Trikalinos TA. Deploying an interactive machine learning system in an evidence-based practice center: Abstrackr. Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium. Miami, Florida, USA: Association for Computing Machinery; 2012. pp. 819–24.

23. Cheng SH, Augustin C, Bethel A, Gill D, Anzaroot S, Brun J, et al. Using machine learning to advance synthesis and use of conservation and environmental evidence. Conservation Biology. 2018;32:762–4.

24. Ouzzani M, Hammady H, Fedorowicz Z, Elmagarmid A. Rayyan—a web and mobile app for systematic reviews. Systematic Reviews. 2016;5:210.

25. Przybyła P, Brockmeier AJ, Kontonatsios G, Pogam M-AL, McNaught J, Erik von Elm, et al. Prioritising references for systematic reviews with RobotAnalyst: A user study. Research Synthesis Methods. 2018;9:470–88.

26. Kilicoglu H, Demner-Fushman D, Rindflesch TC, Wilczynski NL, Haynes RB. Towards Automatic Recognition of Scientifically Rigorous Clinical Research Evidence. Journal of the American Medical Informatics Association. 2009;16:25–31.

27. Aphinyanaphongs Y. Text Categorization Models for High-Quality Article Retrieval in Internal Medicine. Journal of the American Medical Informatics Association. 2004;12:207–16.

28. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011;12:2825–30.

29. Zhang H. The Optimality of Naive Bayes. 2004.

30. Breiman L. Random Forests. Machine Learning. 2001;45:5–32.

31. Ramos J, others. Using tf-idf to determine word relevance in document queries. Proceedings of the first instructional conference on machine learning. Piscataway, NJ; 2003. pp. 133–42.

32. Le QV, Mikolov T. Distributed Representations of Sentences and Documents. 2014 [cited 2020 Feb 4]; Available from: http://arxiv.org/abs/1405.4053

33. Fu JH, Lee SL. Certainty-Enhanced Active Learning for Improving Imbalanced Data Classification. 2011 IEEE 11th International Conference on Data Mining Workshops. Vancouver, BC, Canada: IEEE; 2011. pp. 405–12.

34. Appenzeller-Herzog C. Data from Comparative effectiveness of common therapies for Wilson disease: A systematic review and meta-analysis of controlled studies [Internet]. Zenodo; 2020. Available from: https://doi.org/10.5281/zenodo.3625931

35. Appenzeller-Herzog C, Mathes T, Heeres MLS, Weiss KH, Houwen RHJ, Ewald H. Comparative effectiveness of common therapies for Wilson disease: A systematic review and meta-analysis of controlled studies. Liver International. 2019;39:2136–52.

36. Cohen AM, Hersh WR, Peterson K, Yen P-Y. Reducing Workload in Systematic Review Preparation Using Automated Citation Classification. J Am Med Inform Assoc. 2006;13:206–19.

37. Kwok KTT, Nieuwenhuijse DF, Phan MVT, Koopmans MPG. Virus Metagenomics in Farm Animals: A Systematic Review. Viruses. Multidisciplinary Digital Publishing Institute; 2020;12:107.

38. Hall T, Beecham S, Bowes D, Gray D, Counsell S. A Systematic Literature Review on Fault Prediction Performance in Software Engineering. IEEE Transactions on Software Engineering. 2012;38:1276–304.

39. Nagtegaal R, Tummers L, Noordegraaf M, Bekkers V. Nudging healthcare professionals towards evidence-based medicine: A systematic scoping review [Internet]. Harvard Dataverse; 2019. Available from: https://doi.org/10.7910/DVN/WMGPGZ

40. Nagtegaal R, Tummers L, Noordegraaf M, Bekkers V. Nudging healthcare professionals towards evidence-based medicine: A systematic scoping review. Journal of Behavioral Public Administration. 2019;2.

41. van de Schoot R, Sijbrandij M, Winter SD, Depaoli S, Vermunt JK. The GRoLTS-Checklist: Guidelines for reporting on latent trajectory studies. Structural Equation Modeling: A Multidisciplinary Journal. Routledge; 2017;24:451–67.

42. R Core Team. R: A language and environment for statistical computing [Internet]. Vienna, Austria: R Foundation for Statistical Computing; 2019. Available from: https://www.R-project.org/

43. Řehůřek R, Sojka P. Software framework for topic modelling with large corpora. Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks. Valletta, Malta: ELRA; 2010. pp. 45–50.

44. Reimers N, Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 2019 [cited 2020 Feb 10]; Available from: http://arxiv.org/abs/1908.10084

45. Lewis DD, Catlett J. Heterogeneous Uncertainty Sampling for Supervised Learning. In: Cohen WW, Hirsh H, editors. Machine Learning Proceedings 1994. San Francisco (CA): Morgan Kaufmann; 1994. pp. 148–56.