Training sample selection: impact on screening automation in diagnostic test accuracy reviews

A.J. van Altena[1], R. Spijker[2,3], M.M.G. Leeflang[1], and S.D. Olabarriaga[1]

[1]Amsterdam UMC, University of Amsterdam, Department of Epidemiology and Data Science, Amsterdam Public Health, Amsterdam, The Netherlands
[2]Amsterdam UMC, University of Amsterdam, Medical Library, Amsterdam Public Health, Amsterdam, The Netherlands
[3]Cochrane Netherlands, Julius Center for Health Sciences and Primary Care, University Medical Center Utrecht, Utrecht University, The Netherlands
{a.j.vanaltena, r.spijker, m.m.g.leeflang, s.d.olabarriaga}@amsterdamumc.nl

16th July 2021

A Supplemental Materials

A.1 Glossary

- **prediction model** or **model**: a construct that predicts the chance that a document is included in a review. A model needs to be trained on labelled data in order to build its prediction algorithm. For systematic reviews these data are documents from the reviews, labelled 'inclusion' or 'exclusion';
- **target review**: the review for which a model is built;
- **test set**: the documents from the target review;
- **training set**: the documents from the remaining 49 reviews.

A.2 Parameter Selection

Each classifier method has parameters that need to be set before training on a dataset. To build the model with the highest performance the optimal value of each parameter needs to be found. For the Random Forest classifier we tested values for: `bootstrap`, `max_depth`, `max_features`, `min_samples_leaf`, `min_samples_split`, and `n_estimators` [4]. Because the number of possible parameter value combinations quickly increases a random search is done first. In the random search a random set of parameter values is tested to get a general sense of the optimal parameter settings.

For the random search we chose parameter values as follows. Parameters with numeric values were mostly chosen with equal steps between two extremes.

For example, the `max_depth` test range was $[10, 20, 30, \ldots, 90, 100, 110]$. For parameters without clear boundaries online resources such as [3] were used to determine a suitable range. Parameters for which a choice had to be made (boolean or from a list of options) included most, if not all, options. For example, the `bootstrap` range contained the complete set of options: `True`, `False`. If options were dropped it was because they were similar to another option in the set. For example, the `max_features` may contain: `auto`, `sqrt`, and `log2`. We chose to forgo the `auto` option because it is equal to `sqrt`. Because this resulted in a search grid with less options the search time was reduced.

The random search was performed on ten of the fifty systematic reviews in the dataset. The combination of parameters that yielded the best result was kept for each review tested. Using these results the values that would be tested in the full grid search was determined by one of the researchers (AA):

1. if one value gave the best result for all tested reviews it was chosen as a definitive value;
2. if a value had a small range of values the range was used in the full search;
3. if a value had a large range, the extremes of the range were kept but most of the intermediate values were removed.

These choices were made to size the full grid in such a way that training the models on the complete dataset could be run in an acceptable amount of time.

Final prediction models were trained using a full grid search for each systematic review. The difference with random grid searches is that the complete set of parameter value combinations is tested. The model with the best performance is returned from the full grid search. The parameters used for both the random and full searches are described in the code found in [1].

A.3 Cosine Similarity Analysis

We calculated the cosine similarity between each review and the remaining 49 reviews. The similarity scores were then sorted from highest to lowest and plotted in Figure 1. Each line in Figure 1 depicts the cosine similarity score of a target review to the remaining 49 reviews. Overall, after ordering, the similarity starts high and drops down rapidly in the first couple of reviews. Reviews tend to be most similar to less than ten other reviews. For this reason, a maximum of ten reviews was chosen for construction of the training sets for the SIMILAR approach.

A.4 Performance Trends

To get insight in the overall trends in the performance results we plotted the WSS@95 of the SIMILAR and ALL approaches. The results for both approaches were split into the fifty reviews and sorted from lowest to highest median WSS@95, the plot is shown in Figure 2.

On visual inspection of the plot a couple of observations were made. There is a relatively large spread in the performance of the SIMILAR approach. Often

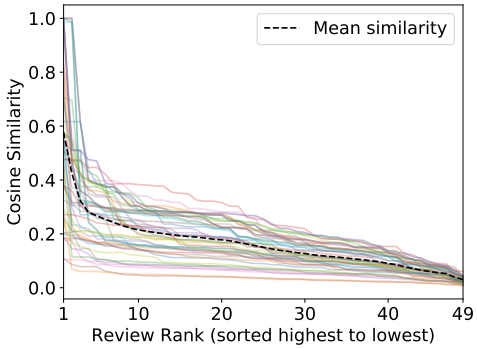

Figure 1: Cosine similarities between all pairs of reviews in our dataset (2,450 pairs). Each line in the figure represents a review and its similarity against the other 49 reviews, ranked from most to less similar.

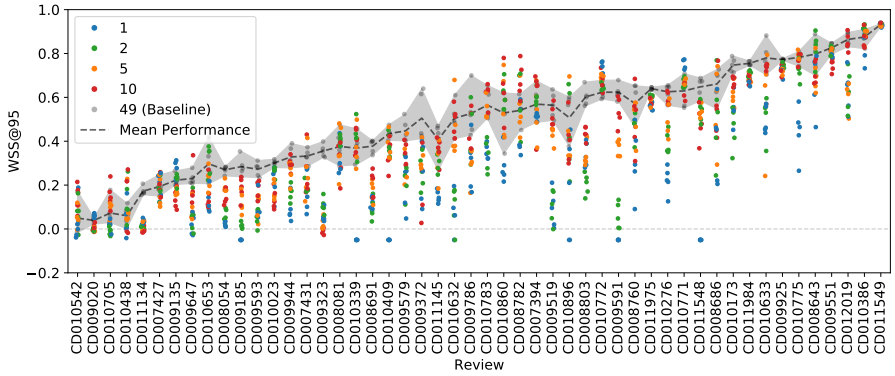

Figure 2: WSS@95 performance for SIMILAR models and ALL models for individual reviews (total 1,250 models). Each colour represents a training set size used for the SIMILAR approach. Note that, to adjust for variance, five models were trained for each training set size. Therefore, each review has five points per colour. The grey area represents the performance range of the ALL approach. The reviews are ordered by increasing median performance in the ALL approach.

the smaller training sets ($n \in \{1, 2\}$) have a lower performance compared to the larger sets ($n \in \{5, 10\}$). However, there is no clear trend in performance gain or loss.

Some of the reviews perform better on a smaller training set (e.g. CD010772), others have approximately the same performance (e.g. CD011549), and the remaining reviews clearly perform better on more data (e.g. CD009323). A clear trend is found in the overall performance of reviews when put next to each other. For some reviews, performance is always high (e.g. CD0111549), or low (e.g. CD009020), irrespectively of training set size. In their work Cohen et al. [2] noted that one of their reviews had a divergent prediction performance (i.e., lower than all others), which they figured was likely due to the low number of inclusions.

To test this observation we investigated the characteristics of the reviews. For each review we collected the following:

- percentage of empty abstracts in the review as a whole;
- percentage of empty abstracts only in the included documents;
- percentage of included documents;
- average number of words in the abstracts and titles;
- number of documents;
- whether the review was an update;
- and, the review's publication year.

A Pearson correlation was calculated between each of the metadata columns and the performance of the ALL approach.

The correlation results are shown in Table 1. No strong correlations were found. The percentage of empty abstracts in the included documents was moderately negatively correlated with performance (-0.28). As was the percentage of included documents (-0.24). The number of words in the abstracts was moderately positively correlated with performance (0.22).

Table 1: Pearson correlations between review metadata and WSS@95 performance.

Metadata type	Performance
% empty abstracts	0.08
% empty abstracts in inclusions	-0.28
% inclusions	-0.24
# words in abstracts	0.22
# words in titles	0.05
# of documents	0.05
is update	-0.04
publication year	0.11

Base on these results we hypothesise that the differences in performance are likely due to a combination of review metadata. Or perhaps there are some

unobserved influences such as the availability of reviews in the dataset written by the same authors. These reviews will often have similar research topics and search strategies and are therefore more valuable in the training set. However, an in-depth analysis of these effects is outside of the scope of this paper and remains for further research.

A.5 Computational Effort

To illustrate the difference in computational effort we tracked the training time for two training set sizes: 1 and 49. The results are plotted in Figure 3. The training set with one review is significantly ($p < 0.001$) faster. The results show that computational effort is much lower for smaller training sets.

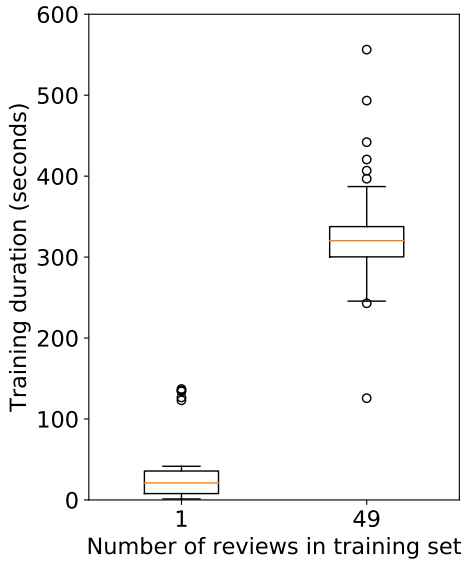

Figure 3: Training time, measured in seconds, for training set sizes 1 and 49.