Statistical Learning Techniques for the Partial Automation of Article Screening in Meta Analyses Pertaining to Psychology

Nosten, Tobias

Universiteit Leiden

Faculteit der Sociale Wetenschappen

Statistical Learning Techniques for the Partial Automation of Article Screening in Meta-Analyses Pertaining to Psychology

Tobias Nosten

Master's Thesis Psychology,

Methodology and Statistics Unit, Institute of Psychology

Faculty of Social and Behavioral Sciences, Leiden University

Date: March 8th 2021

Student number: s1518771

Supervisor: Dr. Elise Dusseldorp (internal), Dr. Clara Calero-Medina (external)

Abstract

The article screening process for meta-analyses is time-intensive and laborious. Statistical learning and natural language processing techniques can be used to partially automate this process. In this study, the performance of four models were compared using a range of evaluation metrics. The first model was built using Latent Dirichlet Allocation (LDA) to extract the topics from the articles to be used as input for a random forest. The second model was built using LDA topics as input for an Extreme Gradient Boosted (XGBoost) tree. The third and fourth models added to the first two by also incorporating a bibliometric feature as input to the respective classifiers. To compare these models, the article catalogues from two meta-analyses pertaining to the field of psychology were gathered and processed. Thus, two real life data-sets were used for the analysis. All four models were built using the full body of text from the article as input for the LDA. These four models were pitted against a benchmark model which represented the more conventional approach to automated article screening. In both datasets, all four proposed models outperformed the benchmark model across all the performance metrics. In the first dataset, the model using LDA topics and bibliometric features as input to the XGBoost was the highest performing model. In the second dataset, the model using LDA topics and bibliometric features as input to the random forest was the highest performing model. The results of this study support the growing evidence that the partial automation of article screening for meta-analyses is indeed possible with a high level of efficiency.

Table of Contents

Contents

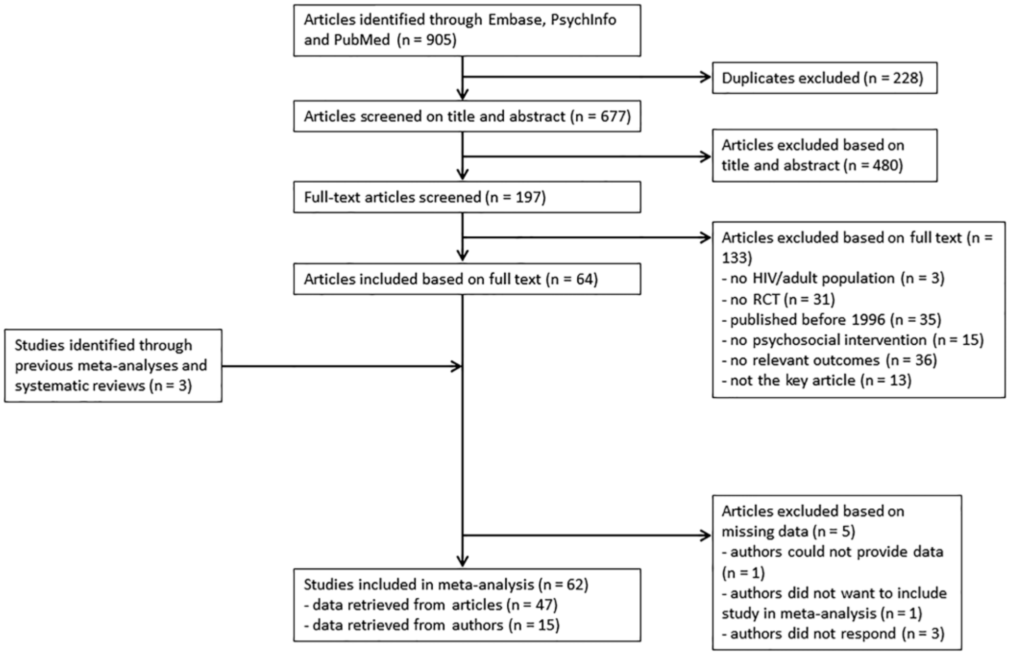
1 Introduction

Systematic reviews and meta-analyses form an important component in the landscape of academic publications; they are used to synthesize the results found in a set of individual articles pertaining to a specific topic. With the influx of academic articles increasing dramatically year by year (Larsen & von Ins, 2010), it is becoming increasingly challenging for researchers to have a comprehensive overview of the literature at a given moment. Thus, research synthesis is an increasingly important endeavor. The automation of systematic reviews is a growing, multidisciplinary field of study which offers some solutions to this issue. With current machine-learning technologies, it is not yet feasible to fully automate the process of writing a systematic review from start to finish (Tsafnat et al., 2013). Certain stages in the process, like designing the research question, require some creative input from authors (Tsafnat et al., 2014). In other words, rather than relying on machine-learning based tools for every step of the process, having humans in-the-loop presents some advantages, and thus partial automation of systematic reviews may be a more tenable solution than full automation.

Several stages in the process of creating systematic reviews have been targeted for partial automation, namely finding previous literature, searching for and de-duplicating articles, screening abstracts, and obtaining full-texts (Jaspers et al., 2018). The greatest gains in efficiency are likely to be found in the identification of relevant studies to be used for the systematic review (Thomas et al., 2011). Illustrated in figure 1, this process of filtering out irrelevant papers can be especially time consuming, particularly during the full-text screening stage when the reviewer has to read through an entire article to determine whether it should be included or not. This is exacerbated in large-scale systematic reviews, when reviewers have to read through hundreds, and potentially thousands of articles. A common strategy to alleviate this load is by acquiring the help of a second reviewer. However, this is not a flawless solution since second reviewers need to be trained, and need to make judgements as similar to the primary reviewer as possible. A practical use-case of automating the article selection process would be to replace the second reviewer, so as to make more efficient use of the resources available to researchers (Olofsson et al., 2017).

Figure 1

Example of a Workflow for Article Screening in Meta-Analyses


Articles identified through Embase, PsychInfo and PubMed (n = 905)
Duplicates excluded (n = 228)
Articles screened on title and abstract (n = 677)
Articles excluded based on title and abstract (n = 480)
Full-text articles screened (n = 197)
Articles excluded based on full text (n = 133)
- no HIV/adult population (n = 3)
- no RCT (n = 31)
- published before 1996 (n = 35)
- no psychosocial intervention (n = 15)
- no relevant outcomes (n = 36)
- not the key article (n = 13)
Articles included based on full text (n = 64)
Studies identified through previous meta-analyses and systematic reviews (n = 3)
Articles excluded based on missing data (n = 5)
- authors could not provide data (n = 1)
- authors did not want to include study in meta-analysis (n = 1)
- authors did not respond (n = 3)
Studies included in meta-analysis (n = 62)
- data retrieved from articles (n = 47)
- data retrieved from authors (n = 15)

Note. A Flow chart of study inclusion and exclusion, taken with permission from (Van Luenen et al., 2018)

The task of reading and classifying text data into categories (0 = not included, 1 = included) is one that machine-learning techniques may be especially suited for. However, the text data must be preprocessed and formatted before the machine-learning techniques can be applied. This preprocessing is done using techniques from the field of natural language processing (NLP). In the context of the current study, the goal of these techniques is to select features of the text in order to quantify and thus characterize each article. These features can then be used as the input for the machine-learning techniques. Perhaps the most popular approach to the quantification of articles is the uni-grams approach, which is based on the frequency of individual terms in a document (O'Connor et al., 2018). Jaspers et al. (2018) compared the uni-grams approach with topic modeling (elaborated in section 2.4.1), and found that topic modeling offers similar dimensionality reduction as uni-gram based models, while also offering the advantage of insight on inter-word relations. Furthermore, Mo et al., (2015) found topic-based feature representations of documents to outperform the uni-grams representation in a direct comparison in an article screening task.

Once the articles are transformed into a manageable format, whether using the uni-grams or the topic modeling approach, classifiers can be constructed to determine whether or not to include a given paper for further analyses (O'Connor et al., 2018). This represents a semi-supervised paradigm of machine-learning, since the reviewer has to inform the model in some ways. The classification problem in automating meta-analyses is particularly prone to suffer from class imbalance, as there are much less articles included in the final meta-analysis than there are excluded. In a comparison of a wide range of classifiers used to screen articles, Jaspers et al. (2018) found that after correcting for class imbalance using the Random OverSampling Examples (ROSE) technique, random forest generally outperformed other classifiers in terms of sensitivity, specificity, and precision of the models. Support Vector Machines (SVM) have also been shown to perform well on tasks of this nature (Ouzzani et al., 2016).

The bibliometric properties of an article may serve as an enhancement to the text features found using NLP techniques, when used as the input variables to a classifier. Given the assumption that researchers cite other similar papers in their work, one can infer closeness of publications by looking at the pattern of citations. A novel approach to clustering scientific documents is to enhance NLP techniques using the citation contexts, which are the texts surrounding the reference markers used to refer to other scientific works (Aljaber et al., 2008). Khabsa et al. (2016) used citation features in addition to text features as an input to a random forest classifier, and found that this outperformed the random forest constructed using only text features. Subelj et al. (2020) introduced the idea of intermediacy of publications, in which citation networks can be used to examine how key terms in specific documents are linked to older or newer publications. In aggregate, there is compelling evidence that the bibliometric characteristics of scientific articles can be of use for automated article screening.

Past literature on automated article screening have mostly focused on training the classifier using the information contained in the title and abstract of an article (Marshall & Wallace, 2019). Some listed reasons for preferring abstract-based screening include the lack of accessibility, and higher computational cost associated with full text screening (Lin, 2009)**.** Full texts also differ to abstract texts in that abstracts are more concise and are richer in keywords, while full texts contain longer sentences and more occurrences of parenthesized text, which can be consequential for information extraction tools (Cohen et al., 2010). Jaspers et al. (2018) applied the same framework as they used with title and abstract data onto full-text data. A comparison was made between 90 full texts with the corresponding abstracts. Again, they found that random forests and other similar tree ensemble methods coupled with ROSE sampling were particularly effective and outperformed other classifiers. The authors observed that the amount of correctly identified relevant papers was somewhat smaller in full text screening than with abstract screening, but the amount of correctly identified irrelevant articles was higher, thus they could not declare a clear winner between the two approaches. As such, the full-text framework remains an open-ended, and potentially fruitful angle in which to approach the issue of automated article selection.

As of the writing of the current study, there are several web-apps for abstract screening available for consumer use. Many of these implement title and abstract-based corpuses, and use variants of the uni-grams approach for feature extraction (Marshall & Wallace, 2019). One such example is Rayyan (Ouzzani et al., 2016), a free web-app created by the Qatar Computing Research Institute. Rayyan employs the uni-grams approach on the titles and abstracts of a set of articles, and uses the results as the input variables for an SVM. Additionally, Rayyan includes pairs of words (bi-grams), and previously calculated Medical Subject Heading (MeSH) terms as features in this SVM. The advantage of using this software is that it is relatively convenient, user-friendly, and has many complimentary features such as collaborative reviewing options which teams of reviewers may find appealing. For the intents and purposes of the current study, Rayyan is considered state of the art in automated abstract screening (Olofsson et al., 2017).

Despite the competitiveness of using topic models as input for a random forest, it is a tenable assumption that models using novel classifiers may supersede the random forest in this semi-automated article screening task. One such candidate is the Extreme Gradient Boosting (XGBoost) classifier, which in recent years has become a premier classification method in domains like image classification (Samat et al., 2020) and bioinformatics (Deng et al., 2020). XGBoost presents a novel and powerful tree-based classifier which, has not yet been implemented in an automated article screening task. Furthermore, while bibliometric features have proven to be a promising enhancement to text features as input variables for a random forest in this context, a variety of methodologies exist for selecting citation-based features. The Leiden clustering algorithm (Traag et al., 2019) developed at the Center of Science and Technology Studies (CWTS), is a novel method which represents the state of the art in extracting citation-based features. This method has not yet been applied in an article-screening context, and as such, its usefulness in this context will be explored further in this study. Lastly, the current study focuses on full text article screening, which has not yet been extensively studied.

To summarize, the main goal of this study is to explore whether the semi-automation of article screening can be done more effectively by exploring three novel avenues: using an XGBoost classifier compared with random forest classifier, by introducing the bibliometric context by way of the Leiden clustering algorithm, and by using full text data rather than abstract data. Thus, the following research questions will be used to contextualize the study from here on.

1) Will the random forest (i.e., Model A) perform worse than the XGBoost (Model B)
2) Can the performance of Models A and B be improved by including the bibliometric cluster IDs (Models C & D)
3) How does the performance of the models developed in 1) and 2) compare to the benchmark, an SVM model based on unigrams from titles and abstracts

In order to answer these research questions, the current study uses sets of publications from two published meta-analyses and train classifiers in an attempt to get as close as possible to the published findings. In this sense, these two datasets are considered as 'golden standards', in that the correct answers to the article screening task are known, at least to the level of accuracy of a human reviewer. Thus, the models can be evaluated with reference to a practical standard.

2 Methods

2.1 Datasets

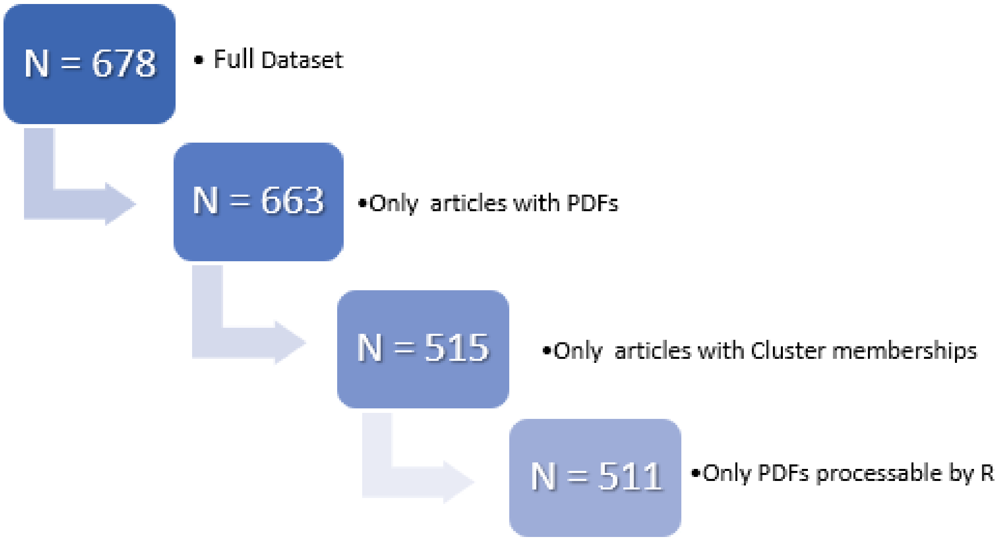
Two datasets are both coming from completed meta-analyses performed by researchers at Leiden University. Each dataset (referred to as dataset 1 and 2) is composed of a selection of articles, which was gathered by the authors of the respective meta-analyses through the use of key word searches in scholarly databases. These selections of articles were shared in the form of a Microsoft Endnote library. Endnote internally assigns each article a unique identifier, which was extracted and maintained throughout the analysis. In the published versions of both meta-analyses, further articles were obtained and were added to the pool of articles contained in the endnote libraries. However, only the initial contents of these endnote files are considered in the current study.

Dataset 1

The first meta-analysis by Van Luenen et al. (2018) examined the benefits of psychosocial interventions for people living with HIV. The initial set of articles (N = 678) was obtained by using a keyword search in Embase, PsychInfo, and PubMed (Appendix 1), and de-duplicating articles. The PDF files associated to these articles were retrieved using the built-in Endnote PDF finding feature and by manually searching on the Web of Science and Google Scholar. Using the 'tm' package in R (Feinerer & Hornik, 2019), these PDFs were transformed into raw text. In the end, there was a noteworthy amount of missing data in dataset 1, with 15 articles being removed because the PDFs were not found. Due to the scope of the algorithm outlined in section 2.4.2, a further 148 articles were not assigned a cluster membership ID by the Leiden algorithm, and thus excluded from the analysis. An additional 4 articles were removed because they were not processable in R. Thus, out of an initial 678 articles, 511 are included in the analysis for dataset 1. The flow diagram of this process is summarized in figure 2 below.

Figure 2

Flow Diagram of Dataset 1

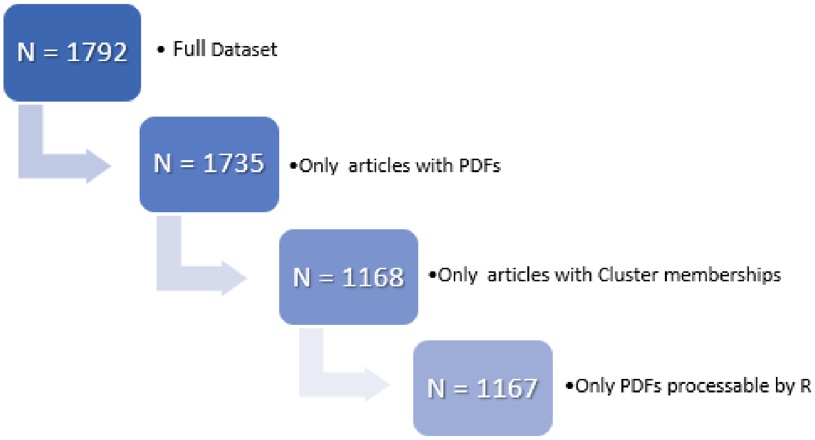

N = 678
• Full Dataset
N = 663
•Only articles with PDFs
N = 515
•Only articles with Cluster memberships
N = 511
•Only PDFs processable by R

Dataset 2

The second meta-analysis examined anxiety disorders in later stages of life. The corresponding data extracted was composed of the entire set of articles (N = 1796) obtained after using a keyword search in Psychinfo, Web of Science, Cochrane Library & Pubmed (Appendix 2), and de-duplicating articles. This set of articles was in the form of an Endnote Library. A total of 57 articles of this set were removed because no PDFs could be found. Due to the scope of the algorithm outlined in section 2.4.2, an additional 569 articles did not have an assigned CWTS cluster ID and were thus excluded. Lastly, one article was removed because it was not processable in R. Thus, out of an initial 1796 articles, 1167 are included in the final analysis of dataset 2. The flow diagram of this process is summarized in figure 3 below.

Figure 3

Flow Diagram of Dataset 2



Imbalanced cases

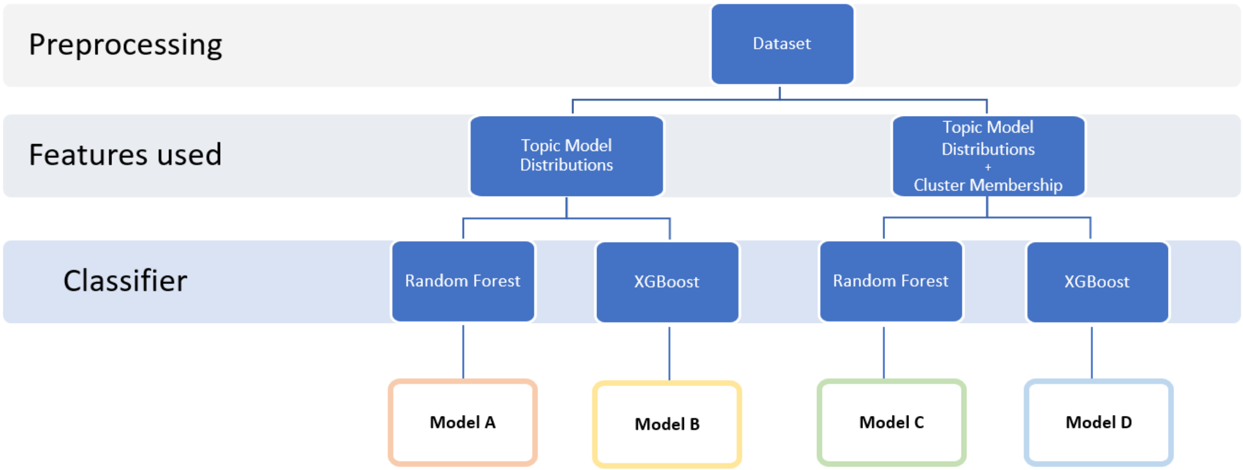
Class imbalance was present in both of the datasets. In dataset 1, of the 511 articles considered in the analysis, only 36 were included in the final meta-analysis, and thus the minority class was only 7% of the total data. Dataset 2 was even more imbalanced, with 34 of the 1167 articles (3%) being selected for the final meta-analysis. Several approaches were put into place in order to address the substantially imbalanced cases in both datasets. The first was up-sampling, which entails randomly sampling the minority class of the training set with replacement to be the same size as the majority class. For example, in training dataset 1 there were 329 articles marked “excluded” (0) and 28 articles marked “included” (1). The up-sampled training set has 329 excluded and 329 included articles. This remedy was performed using the *caret* package in R (Kuhn, 2008). The second remedy was to determine alternative cutoff points for the predicted probabilities by using a receiver operating characteristic curve (ROC) curve, which calculates the sensitivity and specificity across a continuum of cutoff values (Kuhn & Johnson, 2015). These alternate cutoff points were calculated using the *pROC* package in R (Robin et al., 2011).

2.3 Design

As outlined in research questions 1 and 2, ta primary endeavor in this study was to compare the performance of 4 models that will be applied to both datasets. These models differed in terms of the features they were built on, and further by the classifier used, as illustrated in figure 4.

Figure 4

The Elements of Each of the Proposed Models



A 70/30 training-test split was used, in that the classifiers were trained on 70% of randomly selected articles, and validated on the remaining 30% of articles. This training-test split was done to improve the generalizability of the results. The pseudo-Rayyan method alluded to in research question 3 can be considered the inspiration for the 5th model, the performance of which was considered a benchmark in this study. An emulation of the Rayyan method was used because the software does not offer the resulting confusion matrix or other evaluation metrics produced by its SVM classifier. Rather, Rayyan transforms the confusion matrix into a 5-star rating system (Ouzzani et al., 2016), which cannot be directly compared to models A-D explored in the current study. Only a simplification of the Rayyan method was possible because of a lack of availability of MeSH terms and limited computational resources. Each of these models (A, B, C, D, and pseudo-Rayyan) was compared and evaluated based on pre-specified evaluation criteria, expanded upon further in section 2.6.

2.4 Feature extraction

In models A and B, the scores based on the probabilities that a topic occurs in a given article are used as predictor variables for the classifiers. In models C and D, the bibliometric cluster memberships are also considered as features of interest. These bibliometric features are based on a more indirect, social perspective.

2.4.1 Topic Modeling Scores

As previously mentioned, several recent studies have used NLP techniques to select features of the text in order to quantify and characterize each article (O'Connor et al., 2018). A topic modeling approach was chosen in the current study over the uni-grams approach. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is one of the principal approaches to topic modelling, in which an intra-document statistical structure is captured by determining the latent semantic contexts within each document. Each latent semantic context z (henceforth referred to as a topic), is composed of a mixture of w terms, also referred to as words. A word is defined as an item from a vocabulary indexed by $\{1,\ldots,V\}$. Thus, the v^{th} word in the vocabulary is represented by vector $\mathbf{w}$, of length V with V representing the size of the vocabulary.

The LDA model estimates each topic *z* as a mixture of words. The Beta matrix represents the i^{th} topic containing the j^{th} word, defined as $\beta_{ij} = p\ (w_j = 1 \mid z_i = 1)$ where Beta is a *k x V* matrix. The dimensionality *k* of the Dirichlet distribution (which equals the dimensionality of the topic variable *z*) is assumed to be known and fixed. The number of *k* topics is the most important parameter to define in advance (Blei et al., 2003); if *k* is too small, the collection will be limited to very few semantic contexts, whereas if *k* is too large, the risk becomes that the collection is divided into too many contexts and loses interpretability. In addition, increasing the value of *k* will lead to higher computational cost.

Each document **w** is composed of a mixture of topics[1]. A document is defined as a vector containing a set of words denoted by $\mathbf{w} = (w_1, w_2, \ldots, w_N)$, where w_n is the *n*th word in the sequence. *N* refers to the number of words in each document. A corpus is a collection of *M* documents, denoted as $D = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$. The document-topic probabilities are given via the gamma matrix, which provides the proportion of words from a given document **w** that are generated from topic *z*. With the parameter α defined as a *k*-vector of which the components $\alpha_i > 0$, and the parameter β as previously defined, the joint distribution of a topic mixture θ, a set of *k* topics **z**, and a set of *k* vectors **w** is given by the following

$$p(\theta, \mathbf{z}, \mathbf{w} | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^{N} p(z_n | \theta)\, p(w_n | z_n, \beta).$$

Consider a two-topic model as an example, where Document w_1 is 80% Topic z_1 and 20% Topic z_2, while Document w_2 is 40% Topic z_1 and 60% Topic z_2. Topic 1 in this example is an abstraction related to psychology while topic 2 is an abstraction related to geology. The most common words in the psychology topic might be “depression”, “patients” and “anxiety”, whereas the most common words in the geology topic, might be “sediment”, “material” and “earth”. The model is estimating that, for example, only about 20% of words in Document 1 were generated by topic 2. Further, it computes the probability that a term belongs to a given topic, for instance the term “anxiety” has a high probability of being generated from z_1 and a low probability of being generated from z_2. However, it is important to note that words can be shared between topics, for instance the term “sample” is used by psychologists and geologists, and could have a high probability of being generated from either z_1 or z_2.

[1] Following the original notation by Blei et al., (2003), a distinction is made between the italicized w, representing a word, and boldface w, representing a document.

The “topicmodels” package in R (Grün & Hornik, 2011) is used to create to LDA topic model distributions in the current study. The only parameter to choose in this implementation is the value of *k*. A value of *k* = 20 topics is considered a standard value for this parameter (Blei et al., 2003), and will be implemented in this study.

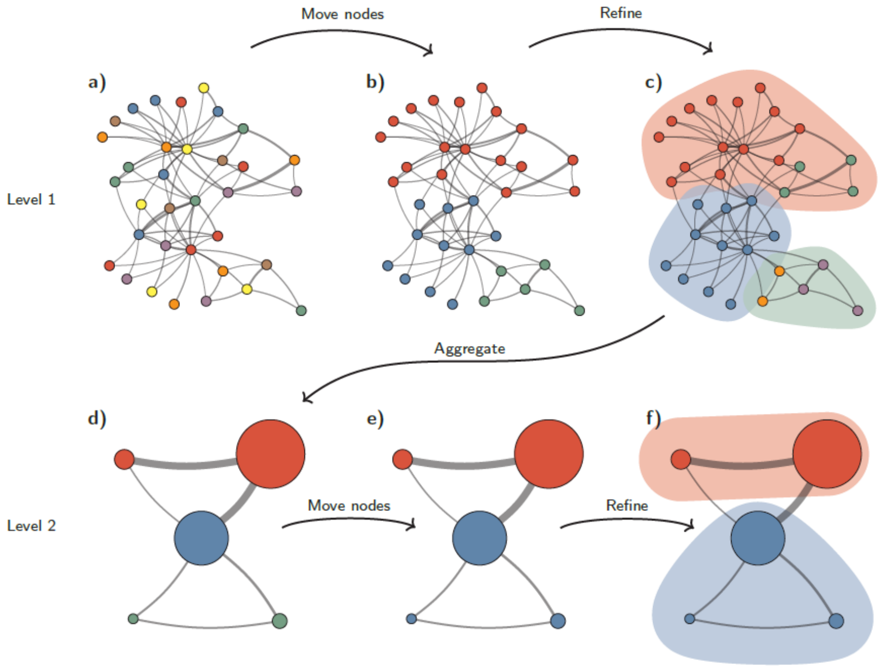
2.4.2 Bibliometric clustering

The CWTS cluster memberships used in models C and D can be considered a bibliometric feature. Bibliometrics in this case is characterized as taking a meta-scientific perspective, considering the nuances of how scientific information in academia is communicated, disseminated and consumed. Of particular interest is the citation of articles, whereby the authors of a given article cite another article, thereby creating a bibliometric link between the two. The Leiden clustering method (Traag et al., 2019) assigns cluster values to publications on the basis of citation relations, regardless of the direction of the citation; rather than placing importance on whether an article is the citer or being cited, the importance lies within the link. As illustrated in figure 5, the Leiden algorithm was constructed to detect communities, defined as dense groups of nodes in a complex cluster. The input of the algorithm is the citation relations, and the output is one variable denoting the cluster group membership. Research areas are then defined at different levels of granularity, and are organized hierarchically. In order of increasing granularity, all articles available in Web of Science are grouped into 23 broad disciplines at the first level, followed by 805 fields, and then 4013 subfields at the most granular level.

By assigning each of the articles in the dataset a cluster ID, we are identifying subgroups in the dataset which reflect the research area that the article belongs to. Examining the subgroup membership rather than individual article ID acts as a dimension reduction step in the current analysis. This algorithm was applied to all the articles made available by the Web of Science, with the date of publication after 1999. Thus, articles which are not within the scope the Web of Science or published before this date do not have a cluster membership ID. The only parameter to choose in determining the Leiden clustering ID is the granularity of the research area. The most granular level of 4013 subfields is used in constructing models C and D since this level offers the most detail and is recommended by the authors (Waltman & Van Eck, 2013).

Figure 5

A Visualization of the Leiden Algorithm



Note. The process underlying the Leiden algorithm, which clusters articles together based on bibliometric relations. Taken with permission from (Traag et al., 2019).

2.5 Statistical Analysis

The task of automating the article selection process for meta-analyses entails classifying the articles into two classes: to be included in the systematic review (1) and to be excluded (0). Throughout the remainder of the study, this will be viewed under the lens of document classification problem. Supervised learning techniques will be used since there are two golden standards available for this project. The input variables for the classifiers used in this study will be LDA topic distributions, and the cluster membership variables as produced by the Leiden algorithm. The classifiers that will be used are discussed below.

As alluded to in research question 1, tree-based classifiers were the method of choice in the current study. Simple classification trees are basic prediction tools that can be used when the outcome variable is categorical. The predictor variable space *{X_1, X_2, …, X_p }* is separated into a set of distinct rectangular regions *{R_1, R_2, …, R_p}* which correspond to the terminal node, or the leaf of a tree, each with a constant predicted value $\hat{y}_j$. The tree is grown by recursively splitting training observations with a simple prediction model applied at each partition. Typically, when building a classification tree, the Gini index is used to evaluate the quality of the split, such that out of all the splits of the candidate variables, the split which leads to the lowest Gini index is chosen. The Gini index is defined as

$$\mathrm{G} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

where $\hat{p}_{mk}$ is the proportion of training observations in the *m*th region that are from the *k*th class.

While individual classification trees are easily interpretable and applicable, they suffer from the bias-variance tradeoff: small, simple trees generally have low variance and high bias, whereas large, complex trees generally have low bias and high variance (James et al., 2013). This tradeoff can also be characterized as trying to achieve balance between the simplicity of a model necessary for generalization, and the complexity necessary for predictive power. Ensemble methods navigate this balance by generating many individual weak learners like trees, giving each learner a slightly different set of training data. Ensembles usually have an estimated prediction error smaller than or equal to the individual learner (James et al., 2013). In order to estimate predictive accuracy, ten-fold cross validation can be used on the training data to get a good estimate of the prediction error to be expected.

There are several general tuning parameters for tree ensembles. The first is depth, which refers to the number of splits and thus size of the tree. Large trees can lead to very unstable results when the predictor variables are weakly related to the response variable and are intercorrelated (Segal, 2004). Tree pruning refers to the process of applying a maximum threshold on the depth of the tree, in essence removing the least important splits. The second tuning parameters is the number of trees. In principle, increasing the number of trees would reduce the variance and thus ameliorate the accuracy of the model. Ensemble methods generally have good accuracy, however over-fitting remains an issue and scaling to large datasets has been a challenge in the past (James et al., 2013). While this increasing complexity may normally lead to overfitting of data, the ensemble methods outlined below have measures in place to deal with this.

Random forests

As previously mentioned, random forests (Breiman, 2001) are a powerful classification tool that have been shown to have success in the text classification of articles (Jaspers et al, 2018). The bootstrap aggregation (bagging) procedure lies at the core of random forests. A number of trees are built on bootstrapped training samples, and the predictions of each individual tree are averaged so as to reduce the inherently high variance of the individual decision trees. The trees are kept unpruned (tree depth is set to maximum depth) so that they retain low bias. Random forests have an additional tuning parameter, namely the number of predictor variables used for split selection. At each split considered, a random subset of *m* predictors is chosen as split candidates from the full set of *p* predictors, with $m \approx \sqrt{p}$ as a default. By using a random subset of predictors at each split, the individual trees are decorrelated, which reduces the variance when the trees are averaged. This choice of predictor subset size *m* is what distinguishes random forests from bagging. In regards to predicting the final class of the text data, the class receiving the majority of the votes from individual trees is considered to be the prediction from the forest. The "randomForest" R package (Liaw & Wiener, 2002) is used in the current study with the default value of $m \approx \sqrt{p}$, being maintained, and the number of trees set to *ntree* = 5000 for both models A and C.

XGBoost

Unlike random forests which use fully grown trees, boosting starts with shallow trees, which are weak learners. Each tree is grown sequentially, using information gathered from previously grown trees. In other words, boosting does not involve bootstrap sampling, rather each tree is grown using a modified sample of the original dataset. Boosting methods work by incrementally reducing the error for each tree, thus the algorithm learns more 'slowly' than other trees, which is conducive for less overfitting (Hastie et al., 2017). The shrinkage parameter is an additional tuning parameter for boosted trees. It is characterized as the rate at which boosted trees learn. Boosted trees tend to work best with weak learners and thus tree size is generally kept low as a default (James et al, 2013). Gradient Boosting (Friedman, 2001) implements the gradient descent algorithm in reducing the error, further optimizing conventional boosting methods.

Extreme Gradient Boosting (XGBoost) (Chen & Guestrin, 2016) is a relatively recent ensemble method gaining in popularity. It further pushes the advantages of gradient boosted models by also including optimized features, especially with regularization and parallelized computing. In this context, regularization refers to the inclusion of a regularization term which helps to smooth the final learned weights to avoid overfitting. The parallelized computing refers to the simultaneous use of all the cores in the user's computer, which offers a significant boost in processing speed. The total number of trees is a more critical parameter in boosted ensembles than in random forests, however XGBoost offers built-in functions in the R package to perform cross-validation at each boosting iteration and thus returns the optimal number of trees required.

The *xgboost* R package (Chen & Guestrin, 2020) was used to create models B and D. Aside from the number of trees, the remaining parameters to be specified were the *eta*, *gamma*, *max depth* parameters. The *eta* parameter is analogous to learning rates in gradient boosted trees. Increasing this parameter makes the boosting more conservative. The *gamma* parameter specifies the minimum loss reduction needed to make a split. Increasing *gamma* leads to a more conservative model. Lastly, the *max depth* parameter refers to the maximum depth of the trees, the increase of which leads to a more conservative model. Each of these parameters was tuned by way of a grid search.

2.6 Evaluation Metrics

The correct class membership as identified by the authors of the meta-analyses served as the absolute benchmark for the performance of the classifiers. However, to achieve a correct classification rate of 100% is unrealistic. As such, the performance of a simplification of Rayyan on the same datasets served as a secondary benchmark. Each model was evaluated based on classification sensitivity (true positive rate), and specificity (true negative rate) for all possible cutoff values. Using these, the ROC curve, and consequently the area under the curve (AUC), were assessed to evaluate the model. Using the pROC package, the statistical difference in the AUC of the nested models (A&C, B&D) was calculated to address the second research question.

Considering the utility of the current project, it is assumed that reviewers engaged in an article screening task for a meta-analysis could afford more false positives than false negatives. From a practical perspective, it is relatively inconsequential to include an article that should have been excluded (i.e., a false positive), because this mistake can be corrected at a later stage of the reviewing process, such as during full-text screening. On the other hand, to exclude an article that should have been included (i.e., a false negative) is very consequential to the project, since the number of included articles is generally much more scarce than the number of excluded articles; this would be akin to throwing away a rare gem after mistaking it for a common pebble.

As such, sensitivity was considered more valuable than specificity. Thus, a final specialized metric was used to evaluate the models, calculated as follows:

$$\text{2SenSpec} = (\textit{2 x sensitivity + specificity}),$$

where the sensitivity for each model is weighted twice as heavily than the specificity. For each model, the cutoff value of predicted probabilities which lead to the highest 2SenSpec value, was chosen as the most optimal cutoff value.

3 Results

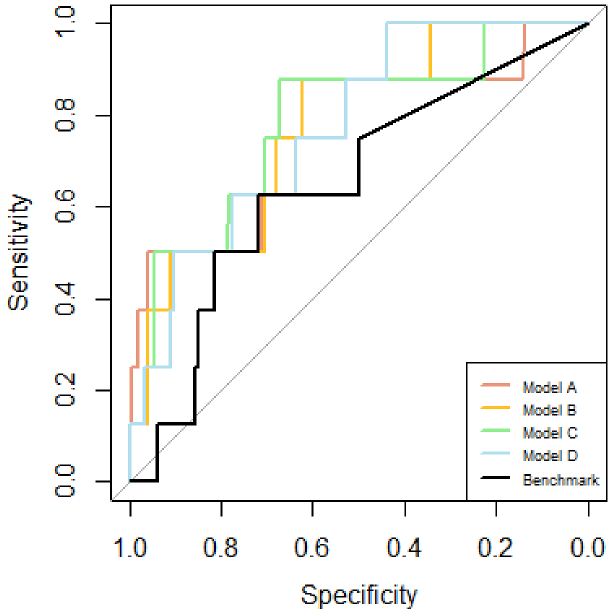
3.1 Dataset 1

Overall, using the optimal cutoff value for predicted probabilities based on the 2SenSpec metric resulted in classifiers with moderate to perfect sensitivity (ranging between 75% -100%), but with relatively low specificity (ranging between 43.8 – 67%). The tradeoff between sensitivity and specificity is illustrated in the ROCs in figure 6.

The evaluation metrics for each of the models for dataset 1 is shown in Table 1, with individual confusion matrices of the classifiers shown in Appendix A. Between the models that do not take into account bibliometric information, the use of XGBoost (Model B) instead of random forest (Model A) did not improve the overall predictive performance. However, between the models that do take into account bibliometric information, the use of XGBoost (Model D) instead of random forest (Model C) did lead to an improvement in overall prediction performance, particularly in regards to the sensitivity. The inclusion of bibliometric cluster memberships in both models C and D did not lead to statistically significant increases in AUC compared to models A and B, respectively. All of the models built with full-text corpuses (A, B, C, and D) outperformed the title and abstract-based benchmark model.

Figure 6

ROC Curves for Dataset 1



Note. The ROC curves representing the predictive test set performance of each model for dataset 1.

Table 1

Test Set Performance of Each Model on Dataset 1.

Random Forest	**x**		**x**		
XGBoost		**x**		**x**	
Cluster IDs			**x**	**x**	
	Model A	Model B	Model C	Model D	Benchmark
AUC	0.770	**0.772**	0.770	0.770	0.647
	(0.57, 0.98)	(0.61, 0.94)	(0.60, 0.97)	(0.62, 0.92)	(0.45, 0.85)
2SenSpec	2.421	2.373	2.421	**2.438**	2.000
Sensitivity	0.875	0.875	0.875	**1.000**	0.750
Specificity	**0.671**	0.623	**0.671**	0.438	0.500

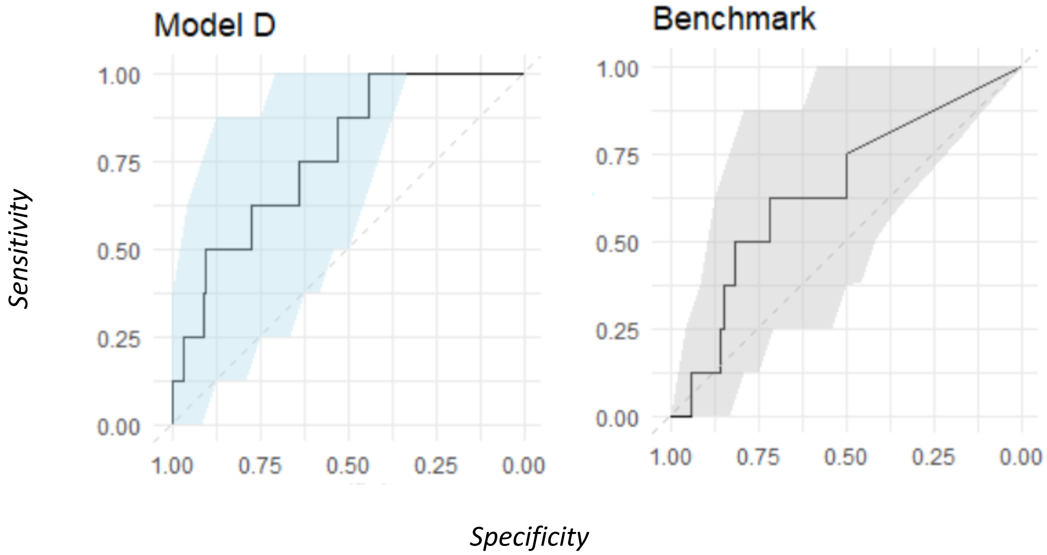
Note. The 95% confidence intervals of the AUC are reported within parentheses. The best performing model per metric is displayed in boldface.

With a 2SenSpec value of 2.438 and a sensitivity value of 100%, the model which took into account bibliometric information and implemented XGBoost outperformed the other 3 proposed models, albeit with a low specificity score of 43.8%. Given the criteria outlined in section 2.6, this model was thus considered the model of choice for dataset 1. As can be seen in the resulting confusion matrix

(Appendix 1), this model correctly filtered out 64 irrelevant articles out of the 154 articles in the test dataset without making a single false negative error. That being said, there was a noteworthy degree of uncertainty in the AUC, as reflected by the large confidence interval illustrated in figure 7 below.

Figure 7

ROC Curves with 95% Confidence Intervals of Best Model and Benchmark for Dataset 1

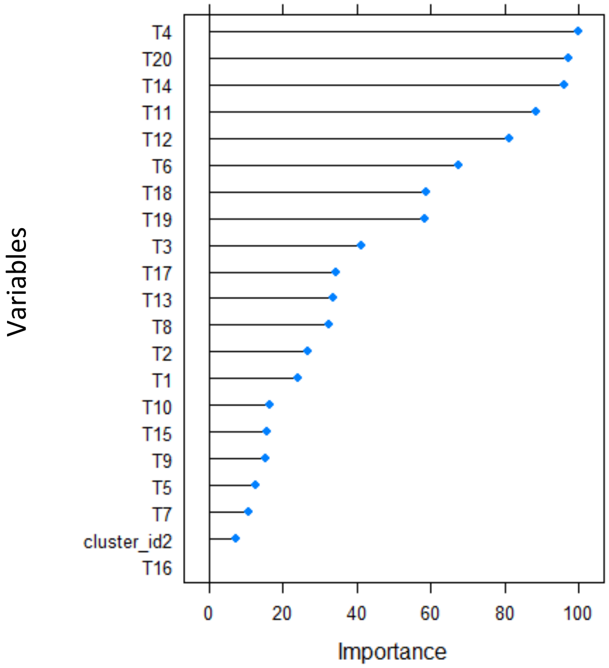


Note. The black lines in these plots show the same ROC curves reported in figure 6 for Model D and the benchmark. The 95% confidence intervals of the sensitivities at a given specificity point are illustrated with the transparent shapes.

Figure 8 gives an overview of which features were the most important for the XGBoost in Model D. The importance measure is based on the number of times a variable is used for splitting in an individual tree, and is weighted by the improvement of the model by each split. Interestingly, the bibliometric feature, denoted as 'cluster_id2', has a low importance measure. The bar plots in Figure 9 illustrate the words which best describe the three most important topics from figure 8. Upon visual inspection, these topics seem to correctly encapsulate the theme of the meta-analysis as described in section 2.1; some of the recurring terms include "HIV", "intervention", and seem to suggest psychological care. For reference, the remaining topics are visualized in Appendix 3.

Figure 8

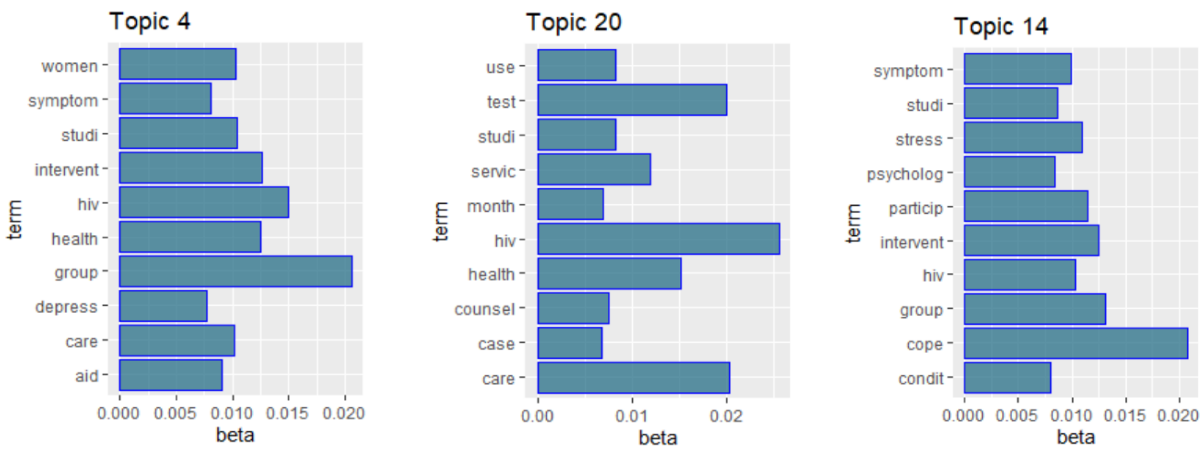
The Variable Importance Plot of Model D



Note. A plot of the variables in the best performing model for dataset 1, with higher values indicating more important features. Variables T1 through T20 denote topics 1 through 20.

Figure 9

Most Descriptive Terms in Each of the Most Important Topics for Dataset 1



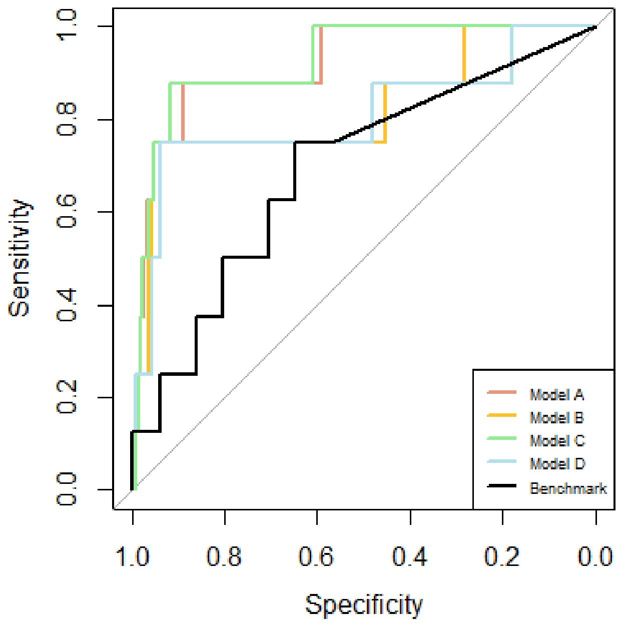
Note. The top ten most representative terms within each topic. The beta value represents the beta matrix of the LDA model, denoting the probability that a term will occur given the topic.

3.2 Dataset 2

Overall, using the optimal cutoff value for predicted probabilities based on the 2SenSpec metric resulted in classifiers with moderate sensitivity (69 – 85%), and with poor to very high specificity (ranging between 64.8 – 95.4%) overall. The tradeoff between sensitivity and specificity in dataset 2 is illustrated in the ROCs of figure 10.

Figure 10

ROC Curves for Dataset 2



Note. The ROC curves representing the predictive test set performance of each model for dataset 2.

The evaluation metrics for each of the models for dataset 2 is shown in Table 2, with individual confusion matrices of the classifiers shown in Appendix B. The use of XGBoost instead of random forest did not improve the overall predictive performance, whether taking into account bibliometric information or not. Rather, the random forest models (Models A and C) outperformed the XGBoost models (Models B and D) with regard to all metrics save for specificity. The inclusion of bibliometric cluster memberships in both models C and D did not lead to statistically significant increases in AUC compared to models A and B, respectively. As with dataset 1, all of the models built with full-text corpuses (A, B, C, and D) outperformed the title and abstract-based benchmark model.

Table 2

Test Set Performance of Each Model on Dataset 2.

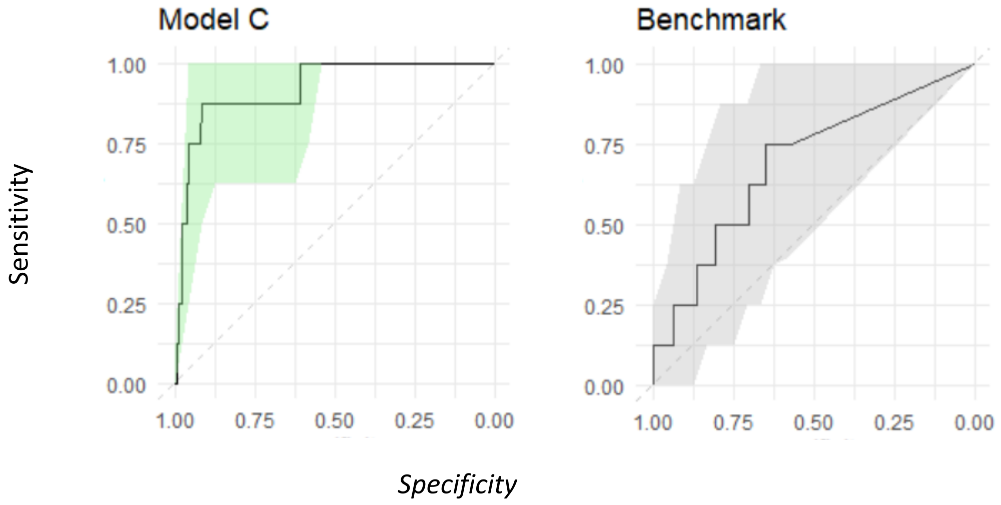
Random Forest	**x**		**x**		
XGBoost		**x**		**x**	
Cluster IDs			**x**	**x**	
	Model A	Model B	**Model C**	Model D	Benchmark
AUC	0.916	0.820	**0.922**	0.806	0.690
	(0.82, 1.00)	(0.62, 1.00)	(0.83, 1.00)	(0.59, 1.00)	(0.5, 0.88)
2SenSpec	2.638	2.454	**2.666**	2.437	2.148
Sensitivity	**0.875**	0.750	**0.875**	0.750	0.750
Specificity	0.888	**0.954**	0.916	0.937	0.648

Note. The 95% confidence intervals of the AUC are reported within parentheses. The best performing model per metric is displayed in boldface.

With a 2SenSpec value of 2.666 and a high value of sensitivity (87.5%) and specificity (91.6%), the model which took into account bibliometric information and implemented a random forest (Model C) outperformed the other 3 models. Given the criteria outlined in section 2.6, model C was thus considered the model of choice for dataset 2. Using the resulting confusion matrix (Appendix 2), we can see that this model correctly filtered out 318 irrelevant articles out of 355 articles in the test dataset, while committing 1 false negative error. As with dataset 1, there was a noteworthy degree of uncertainty in the AUC, as reflected by the large confidence intervals illustrated in figure 11 below.

Figure 11

ROC Curves with 95% Confidence Intervals of Best Model and Benchmark for Dataset 2

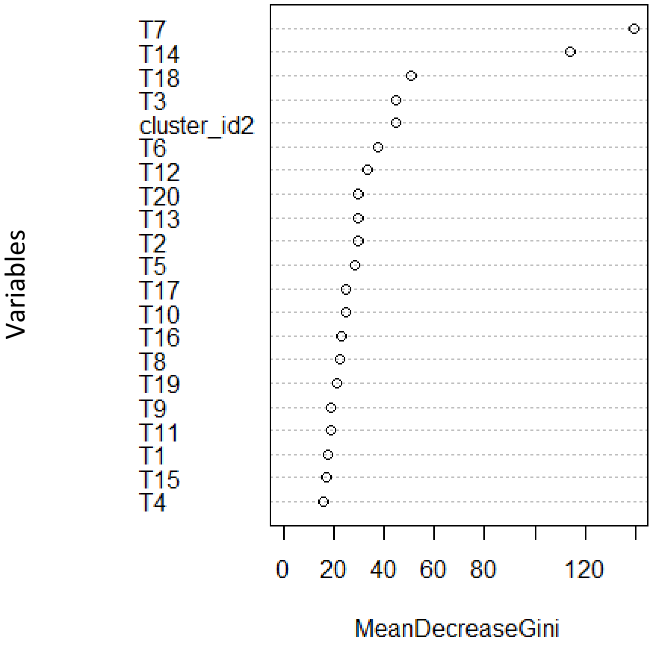


Note. The black lines in these plots show the same ROC curves reported in figure 8 for Model C and the benchmark. The 95% confidence intervals of the sensitivities at a given specificity point are illustrated with the transparent shapes.

The variable importance plot below (Figure 12) gives an overview of which features were the most important for the random forest in Model C. The importance measure is based on the total amount that the Gini index is decreased by splits over a given predictor variable, and averaged over all the trees. The bibliometric feature, denoted as 'cluster_id2', is the fifth most important variable, though considerably less important than the top two. The bar plots in Figure 13 illustrate the words which best describe the two most important topics from figure 12. Upon visual inspection, these topics seem to correctly encapsulate the theme of the meta-analysis as described in section 2.1; some of the recurring terms include "age", "psychiatry", and seem to suggest psychological care. For reference, the remaining topics are visualized in Appendix 4.

Figure 12

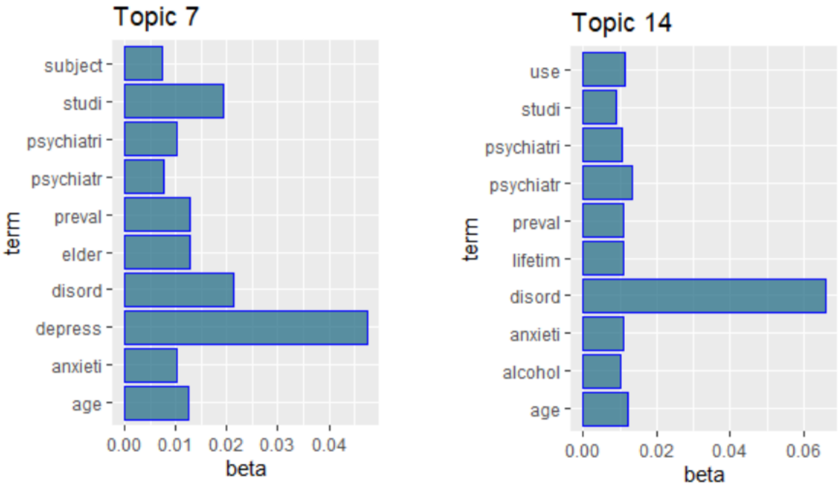
The Variable Importance Plot of Model C



Note. A plot of the variables in the best performing model for dataset 2, with higher values indicating more important features. Variables T1 through T20 denote topics 1 through 20

Figure 13

Most Descriptive Terms in Each of the Most Important Topics for Dataset 2



Note. The top ten most representative terms within each topic. The beta value represents the beta matrix of the LDA model, denoting the probability that a term will occur given the topic.

4 Discussion

In this study, we investigated the partial automation of meta-analyses under the context of three research questions. Before discussing our results, we first summarize the main findings per question.

The first research question examined whether the XGBoost classifier would outperform the random forest classifier. We observed mixed results in the comparison of the two classifiers. Without incorporating bibliometric information, the random forest had a higher 2SenSpec value than the XGBoost for both datasets. This continued to be the case after including bibliometric features in dataset 2. However, with the bibliometric features in consideration, XGBoost (model D) outperformed random forest (model C) for dataset 1. This was due to a major boost in the sensitivity of the model, at the expense of the other metrics.

The second research question examined whether the inclusion of bibliometric cluster memberships would ameliorate the performance of models A and B, especially regarding the 2SenSpec metric. In dataset 1, the inclusion of the bibliometric feature appeared to greatly enhance the performance of the XGBoost, while not affecting the performance of the random forest. However, the variable importance plot for dataset 1 (figure 8) revealed that the bibliometric feature was a relatively unimportant variable. The inverse was observed in dataset 2, where the inclusion of the bibliometric feature improved the performance of the random forest while not improving performance of the XGBoost. In dataset 2, the variable importance plot (figure 12) revealed that the bibliometric feature was the 5th most important variable, though not nearly as important as the top two variables. That being said, the best performing models for both datasets did incorporate bibliometric features. Thus, we can say that the bibliometric cluster membership variable as created by the Leiden algorithm does play a role in the improvement of these models, though the extent is not immediately clear.

The third research question compared the four proposed models (A, B, C, and D) to the benchmark, which employed an SVM constructed using the uni-gram features from titles and abstracts. In both datasets, all four proposed models outperformed the benchmark model.

The remainder of this section serves as a general discussion of the results, as well as recommendations for future research.

Regarding research question 1, the observations for dataset 2 corroborate the results by Jaspers et al. (2018), which found random forests to be among the most competitive options for automated article screening. Overall, the random forests achieved a good balance between sensitivity and specificity. Interestingly, both XGBoost models in dataset 2 had higher values for specificity than for sensitivity, thus leading to relatively poor 2SenSpec values. It is worth noting that even when calibrating the probability threshold to obtain higher sensitivity than specificity, the sudden drop is sensitivity was so dramatic that the adjustment would not make sense. These dramatic jumps in sensitivity are most probably due to the scarcity of included articles, the minority class. This is also illustrated in the step-like appearance of the ROC curves of the models. A further likely consequence of the scarcity of included articles in the test set was the large confidence intervals, pictured in figures 7 and 11.

To this end, the training-test split selected for this study could be considered a limiting factor. Exploring new training-test splits could mitigate the scarcity of the included articles in the test set. Jaspers et al. (2018) found that for the task at hand, using 50% of the data as the training set was optimal, though in some cases even smaller training sets of 20% were conducive to good results. Furthermore, albeit standard practice, the practical implications of having a training set composed of 70% of the data means that a reviewer would have to manually review the majority of the articles before any attempt at partial automation can take place. In order to minimize the negative effects of the severe class imbalance, and maximize the amount of work that can be partially automated, the exploration of varying training-test splits or cross validation in this context warrants further research.

The bias-variance tradeoff, introduced in section 2.5, is relevant in examining the differences in performance between the two classifiers. By design, the random forest starts off with low bias and high variance, and works to reduce error by lowering the variance. On the other hand, XGBoost starts off with high bias and low variance, and works to reduce error by lowering the bias. Both datasets had high class imbalance and were upsampled, which introduces more bias into these datasets. One possible explanation for these results is that the upsampling procedure placed too strong an artificial bias on the training set, and XGBoost was underfitting. Kuhn and Johnson (2015) demonstrate that different imbalance correction procedures differ in their effectiveness, and list cost sensitive classification or ROSE sampling (Menardi & Torelli, 2014) as alternative procedures. Another method of interest is the in-built class weight parameter for XGBoost, which may be more optimal for the performance of the algorithm, given the high level of bias in the datasets in this task. A consideration when evaluating the performance of XGBoost is that the classifier was built using ROC as a training metric. According to Saito and Rehmsmeier (2015), training classifiers with ROC is less robust to imbalanced classes than the precision recall (PR) curve. Methodologies based on the PR curve may be interesting for future research in this field which inherently deals with imbalanced datasets. Regardless, considering its success in dataset 1, more explorations of the XGBoost classifier could be of interest for future research in this field.

With regard to the results for research question 2, the inclusion of bibliometric features as input for the classifiers generally had a positive effect on predictive performance. These observations are in line with the findings by Khabsa et al. (2016), which saw an improvement to the performance of a random forest model after the addition of co-citations, a different type of bibliometric feature. A more direct comparison of the influence of the bibliometric cluster memberships with that of co-citations warrants future investigation. The more detailed underpinnings of how these bibliometric features interact with the text-based features also warrants further investigation.

An important limitation in the current study was the cost of implementing the bibliometric features, namely the lowered sample size in both datasets; dataset 1 and 2 were reduced by 23% and 33% respectively, due to the inclusion of bibliometric features. Particularly when there is already severe class imbalance in these datasets, reducing the sample sizes risked exacerbating the class imbalance, as well as reducing the statistical power (Kang, 2013). Future studies may consider applying the Leiden algorithm to other databases such as Scopus, or calculating pseudo cluster membership IDs for the missing articles by examining the journals that cite them the most frequently.

With regard to research question 3, it was clear that the benchmark model performed relatively poorly for both datasets. The design of the current study does not allow for us to isolate the exact mechanisms underlying this relatively poor performance, since the benchmark model is fundamentally different from the other models presented. In particular, the benchmark differs regarding the input data (the unigrams of abstracts and titles), and the classifier (SVM). Another important consideration is that a simplified approximation was implemented in the current study, with the bigrams and MeSH terms not being incorporated in the benchmark. Thus, the real Rayyan system may have achieved better results. As such, the implementation of the benchmark can be considered a primary limitation of the current study. Two avenues exist for a more direct comparison with Rayyan, the first being direct collaboration with the creators of the software, and the second being converting the results of the current study into the 5-star scaling system used by Rayyan. In either scenario, being able to directly compare the performance with Rayyan would lead to a more accurate evaluation of the proposed models.

From a methodological standpoint, it is worth noting that the design of the study could be considered a limiting factor as well. The use of real data sets entails potential issues, like the possibility that the authors missed articles that should have been included. In a simulation design, this would be mitigated since the absolute truth would be known. Furthermore, due to the design of the current study, it is difficult to determine the substantive contribution of the full text or the LDA components towards the performance of the proposed models. Moreover, since the proposed models were not all nested, testing the significant difference between them became problematic. As such, the comparison of the performance metrics should be done so with caution. Considering the results and limitations of the current study, a logical follow up study should consider a factorial design which directly compares each of the components explored in the current study. In practice, this would imply simulating and/or collecting both abstract and full-text data for all articles in a set, and using both uni-grams and LDA for feature extraction, before using these as input in each classifier (random forest, XGBoost, SVM). This would mitigate the effect of missing data associated with the collection of full-text data performed in the current study as well as allow for a more direct comparison with benchmark models like Rayyan. Most importantly, this would be conducive to collecting stronger evidence by which to compare the models.

Despite the limitations, our findings do shed some light on the open discussion in the automated article screening literature concerning the utility of full text data compared to abstract data (Cohen et al., 2010; Lin, 2009; Jaspers et al., 2018). More specifically, the current study shows that basing the corpus on full-text data can lead to strong performance in an automated article screening task. To a similar extent, the results from the current study show that showcase the effectiveness of using LDA to quantify the corpus.

4.3 Conclusion

The current study sought to examine the possibility of partially automating the article screening process for meta-analyses using statistical learning and NLP techniques. Despite noteworthy missing data and class imbalance, all the proposed models discussed here would likely lead to some increase in efficiency for reviewers of a meta-analysis; on a practical level, these models were indeed able to sift out irrelevant articles, though choosing definitive winners and losers between models remains to be seen. These findings provide support for some novel avenues which as of yet have not been extensively explored and demonstrate that it is possible to at least partially automate the article screening process. The current study proves a framework, which with some development and exploration may well push the boundaries of the young field of automated article screening.

References

Aljaber, B., Stokes, N., Bailey, J., & Pei, J. (2010). Document clustering of scientific texts using citation contexts. *Information Retrieval*, *13*(2), 101–131. https://doi.org/10.1007/s10791-009-9108-x

Cohen, K., Johnson, H., Verspoor, K., Roeder, C., & Hunter, L. (2010). The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinformatics*, *11*(1). https://doi.org/10.1186/1471-2105-11-492

Blei, D., Ng A., & Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 993–1022.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, T., Guestrin, C. (2020). xgboost: Extreme Gradient Boosting. R package version 1.1.1.1, https://cran.r-project.org/web/packages/xgboost/index.html

Deng, A., Zhang, H., Wang, W., Zhang, J., Fan, D., Chen, P., & Wang, B. (2020). Developing Computational Model to Predict Protein-Protein Interaction Sites Based on the XGBoost Algorithm. *International Journal of Molecular Sciences*, *21*(7), 2274. https://doi:10.3390/ijms21072274

Feinerer, I., Hornik, K. (2019). *tm: Text Mining Package*. R package version 0.7-7, https://CRAN.R-project.org/package=tm.

Friedman, F. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics, 29*(5), 1189-1232.

Grün, B., Hornik, K. (2011). topicmodels: An R Package for Fitting Topic Models. *Journal of Statistical Software*, *40*(13), 1–30. doi: 10.18637/jss.v040.i13

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Elements of Statistical Learning (Springer series in statistics). New York: Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning (Vol. 112). New York, NY: Springer.

Jaspers, S., De Troyer, E., & Aerts, M. (2018). Machine learning techniques for the automation of literature reviews and systematic reviews in EFSA. *EFSA Supporting Publications*, *15*(6). https://doi.org/10.2903/sp.efsa.2018.EN-1427

Kang, H. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, *64*(5), 402–406. https://doi.org/10.4097/kjae.2013.64.5.402

Khabsa, M., Elmagarmid, A., Ilyas, I., Hammady, H. & Ouzzani, M. (2016). Learning to identify relevant studies for systematic reviews using random forest and external information. *Machine Learning*, 102, 465–482.

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, *28*(5), 1 - 26. http://dx.doi.org/10.18637/jss.v028.i05

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. New York: Springer New York.

Larsen, P. O., & von Ins, M. (2010). The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics*, *84*(3), 575–603. https://doi.org/10.1007/s11192-010-0202-z

Liaw, A., Wiener, M. (2002). Classification and Regression by randomForest. *R News*, *2*(3), 18—22

Lin, J. (2009). Is searching full text more effective than searching abstracts? *BMC Bioinformatics*, *10*, 46. https://doi.org/10.1186/1471-2105-10-46

Luenen, S. van, Garnefski, N., Spinhoven, P., Spaan, P., Dusseldorp, E.M.L., & Kraaij, V. (2018). The Benefits of Psychosocial Interventions for Mental Health in People Living with HIV: A Systematic Review and Meta-analysis. *AIDS and Behavior,* AIDS and Behavior.

Marshall, I. J., & Wallace, B. C. (2019). Toward systematic review automation: A practical guide to using machine learning tools in research synthesis. *Systematic Reviews*, *8*(1). https://doi.org/10.1186/s13643-019-1074-9

Menardi, G. and Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28:92–122.

Mo, Y., Kontonatsios, G., & Ananiadou, S. (2015). Supporting systematic reviews using LDA-based document representations. *Systematic Reviews*, *4*(1), 172. https://doi.org/10.1186/s13643-015-0117-0

O'Connor, A. M., Tsafnat, G., Gilbert, S. B., Thayer, K. A., & Wolfe, M. S. (2018). Moving toward the automation of the systematic review process: A summary of discussions at the second meeting of International Collaboration for the Automation of Systematic Reviews (ICASR). *Systematic Reviews*, *7*(1). https://doi.org/10.1186/s13643-017-0667-4

Olofsson, H., Brolund, A., Hellberg, C., Silverstein, R., Stenström, K., Österberg, M., & Dagerhamn, J. (2017). Can abstract screening workload be reduced using text mining? User experiences of the tool Rayyan. *Research Synthesis Methods*, *8*(3), 275–280. https://doi.org/10.1002/jrsm.1237

Ouzzani, M., Hammady, H., Fedorowicz, Z., & Elmagarmid, A. (2016). Rayyan—A web and mobile app for systematic reviews. *Systematic Reviews*, *5*(1). https://doi.org/10.1186/s13643-016-0384-4

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J., Müller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, *12*, 77

Saito, T., & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, *10*(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Samat, A., Li, E., Wang, W., Liu, S., Lin, C., & Abuduwaili, J. (2020). Meta-XGBoost for Hyperspectral Image Classification Using Extended MSER-Guided Morphological Profiles. *Remote Sensing*, *12*(12), 1973. doi:10.3390/rs12121973

Segal, M. R. (2004). Machine learning benchmarks and random forest regression. Center for Bioinformatics & Molecular Biostatistics

Šubelj, L., Waltman, L., Traag, V., & van Eck, N. J. (2020). Intermediacy of publications. *Royal Society Open Science*, *7*(1), 190207. https://doi.org/10.1098/rsos.190207

Thomas, J., McNaught, J., & Ananiadou, S. (2011). Applications of text mining within systematic reviews. *Research Synthesis Methods*, *2*(1), 1–14. https://doi.org/10.1002/jrsm.27

Traag, V. A, Waltman, L, & Van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports, 9*(1), 5233-12.

Tsafnat, G., Dunn, A., Glasziou, P., & Coiera, E. (2013). The automation of systematic reviews. *BMJ*, *346*(jan10 1), f139–f139. https://doi.org/10.1136/bmj.f139

Tsafnat, G., Glasziou, P., Choong, M., Dunn, A., Galgani, F., & Coiera, E. (2014). Systematic review automation technologies. *Systematic Reviews,3*(1), 74

Waltman, L., & van Eck, N. J. (2013). A systematic empirical comparison of different approaches for normalizing citation impact indicators. *Journal of Informetrics*, *7*(4), 833–849. https://doi.org/10.1016/j.joi.2013.08.002

Appendix A

Figure 14

Confusion Matrices for all the Proposed Models in Dataset 1

	Truth	
Model A Predictions	Exclude	Included
Exclude	**98**	**1**
Include	**48**	**7**

	Truth	
Model B Predictions	Exclude	Included
Exclude	**91**	**1**
Include	**55**	**7**

	Truth	
Model C Predictions	Exclude	Included
Exclude	**98**	**1**
Include	**48**	**7**

	Truth	
Model D Predictions	Exclude	Included
Exclude	**64**	**0**
Include	**82**	**8**

Appendix 2

Figure 15

Confusion Matrices for all the Proposed Models in Dataset 2

	Truth	
Model A Predictions	Exclude	Included
Exclude	**308**	**1**
Include	**39**	**7**

	Truth	
Model B Predictions	Exclude	Included
Exclude	**331**	**2**
Include	**16**	**6**

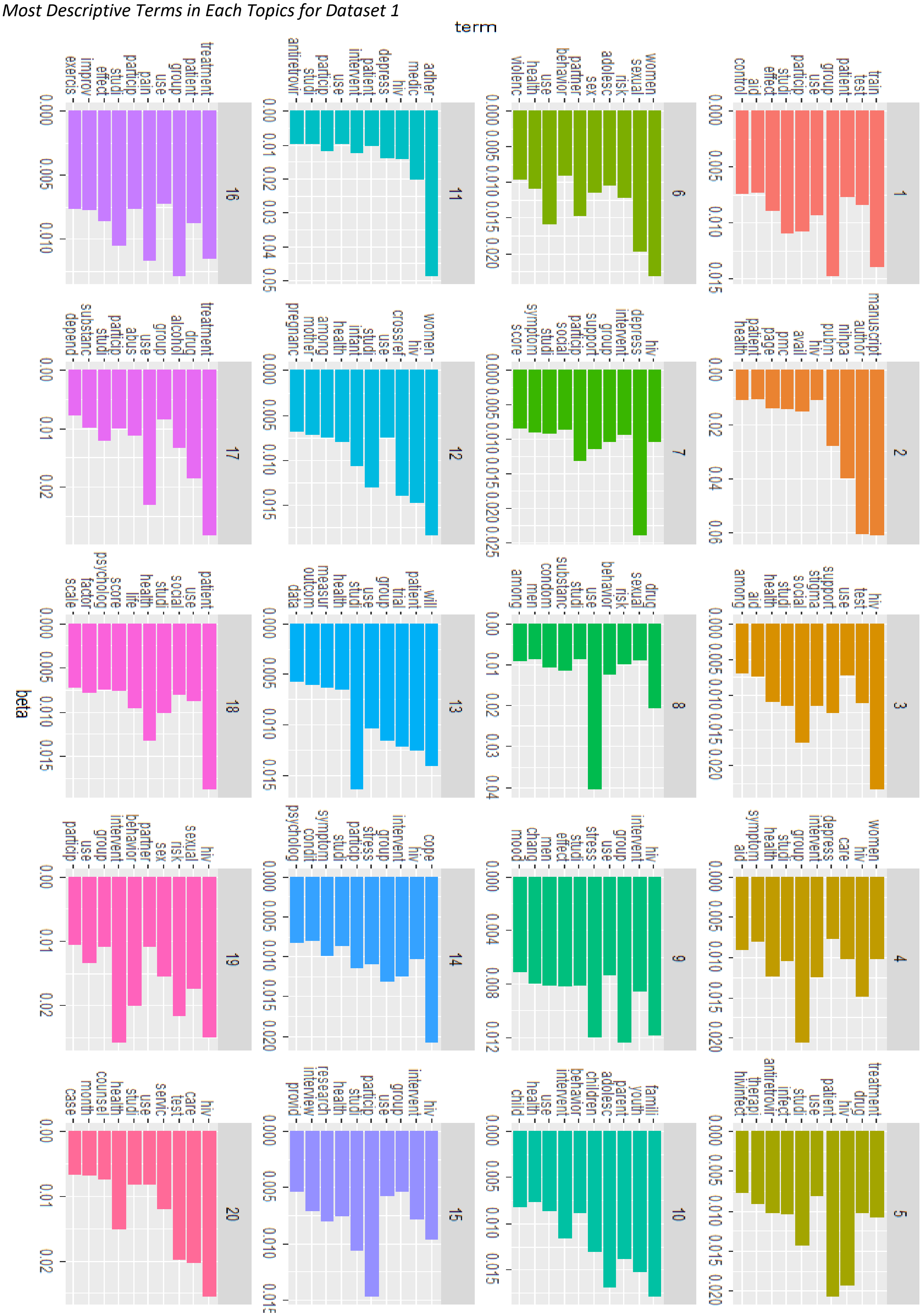
	Truth	
Model C Predictions	Exclude	Included
Exclude	**318**	**1**
Include	**29**	**7**

	Truth	
Model D Predictions	Exclude	Included
Exclude	**325**	**2**
Include	**22**	**6**

Appendix 3

Figure 16

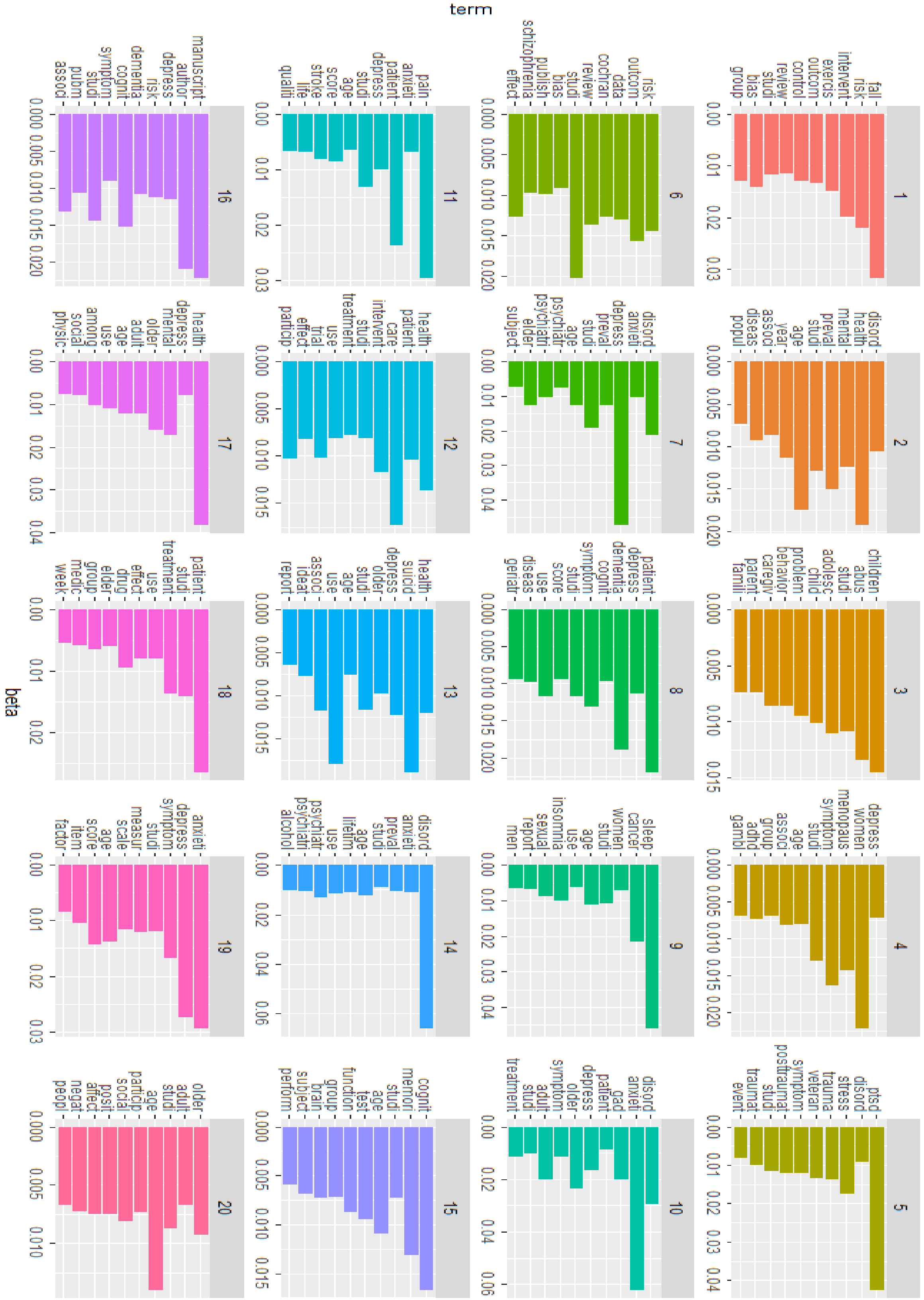
Most Descriptive Terms in Each Topics for Dataset 1



Appendix 4

Figure 17

Most Descriptive Terms in Each Topics for Dataset 2


term
beta
fall
risk
intervent
exercis
outcom
control
review
studi
bias
group
disord
health
mental
preval
studi
age
year
assoc
diseas
popul
children
abus
studi
adolesc
child
problem
behavior
caregiv
parent
famili
depress
women
menopaus
symptom
studi
age
assoc
group
adhd
gambl
ptsd
disord
stress
trauma
veteran
symptom
posttraumat
studi
traumat
event
risk
outcom
data
cochran
review
studi
bias
publish
schizophrenia
effect
disord
anxieti
depress
preval
studi
age
psychiatr
elder
subject
patient
depress
dementia
cognit
symptom
studi
score
use
diseas
geriatr
sleep
cancer
women
studi
age
use
insomnia
sexual
report
men
disord
anxieti
gad
patient
depress
older
symptom
adult
studi
treatment
pain
anxieti
patient
depress
studi
age
score
stroke
life
qualiti
health
patient
care
intervent
studi
treatment
use
trial
effect
particip
health
suicid
depress
older
studi
age
use
assoc
ideat
report
disord
anxieti
preval
studi
age
lifetim
use
psychiatr
psychiatri
alcohol
cognit
memori
studi
age
test
function
group
brain
subject
perform
manuscript
author
depress
risk
dementia
cognit
symptom
studi
pubm
assoc
health
depress
mental
older
adult
age
use
among
social
physic
patient
studi
treatment
use
effect
drug
elder
group
medic
week
anxieti
depress
symptom
studi
measur
scale
age
score
item
factor
older
adult
studi
age
particip
social
posit
affect
negat
peopl