Manuscript drafts

Gerbrich Ferdinands

1/14/2020

Introduction

Methods A convenience sample of 5 existing systematic reviews on varying topics was collected.

Results

Discussion

Introduction

Systematic Reviews (SR's) are booming - but they are a lot of work Various machien learning tools have been proposed to reduce workload in abstract screening.

- objectives - to demonstrate effectiveness of ml algorithm in reducing abstract classification for systematic reviews
- justification -
- background
- guidance to reader
- summary/conclusion

A SR can be divided into phases. Everything starts with a **systematic search**, leading to then citation screening is performed, then full-text screening [16]

What must be the objective of our tool?: It is the tedious task citation screening part where loads of time can be saved.

models are designed in a 'realistic' way (you have some inclusions)

Selecting papers is a two-step process: abstract & fulltext screening

Binary classification problem. predict whether a paper in the pool is an inclusion or exclusion, based on labeled instances from $\mathcal{L}$ and use training data to understand how input variables are related to class.

active learning for systematic reviews

corpus = all the text:

Active learning = increasing classification performance with every query. The query strategy determines the way unlabeled papers are queried to the researcher.

[6]

- pool unlabeled abstracts $\mathcal{U}$
- labeled data set $\mathcal{L}$,
- instance x, label y
- utility measure $\phi_A(\cdot)$
- x_A^* best query instance according to $\phi_A(\cdot)$

`while`

RQ1 - what are good classifiers RQ2 - what are good optimization strategies

Background

[14] literature review

[21], [22] simulated 32 svm classifiers, on software engineering. A popular classifier is SVM. succes with HUTM (fastread), uncertainty, mix of weighting and agressive undersampling, In terms of Yu et al, we adopt .CT.

SVM - tf-idf on medical data, uncertainty sampling, agressive undersampling. [20]

abstrackr

SVM + Weighting + uncertainty (bow) produced good methods [11] Also include social sciences data besides medical data.

[4] perceptron-based classifier (neural network)

SVM on legal documents (no balancing, certainty) [5] in limitations section mentions that LR yields about same results, nb inferior results.

[8] - SVM, naive bayes, boosting and combinations. future work should optimize parameters. “Regarding the base classifiers used in identifying method- ologically rigorous studies, boosting consistently strikes the best balance between precision and recall, whereas naive Bayes in general performs well on recall (demonstrating a tradeoff between recall and precision), as does polynomial SVM on precision. The AUC results are mixed, although boosting has a slight edge overall. These results demonstrate that different classifiers can be used to satisfy different information needs (SVM for specificity, naive Bayes for sensitivity, and boosting for balance between the two, for example).”

Our extensions is that we try different classifiers, on more datasets.

Methods

Goal: evaluate performance of different models of the ASReview tool. The screening process is simulated using ASReview, seeing if the original inclusions replicate. What would happen if the citation screening would have been performed using asreview?

Datasets

The algorithm will be tested on five systematic reviews from various research areas. test datasets serve as systematic search results, then perform active learning to detect inclusions. Statistics on the SRs can be found in Table 1. All datasets accompanying the systematic reviews are openly published. The datasets contain information on all citations obtained in the search strategy and which citations were included in the systematic review.

ace Cohen et al. collected systematic review datasets from the medical sciences [4]. All systematic reviews in this database are on drug efficacy. The ace dataset used in the current study comes from a systematic review on the efficacy of Angiotensin-converting enzyme (ACE) inhibitors.

a machine learning-based citation classification tool to reduce workload in systematic reviews of drug class efficacy. Using a perceptron classifier, WSS@95% = 56.61 in [4]. (5x2 crossvalidation). Can we beat this? The data

software A SR on fault prediction in software engineering by [7]. The dataset is retrieved from [21], who collected datasets on literature reviews from the software engineering field.

nudging A review from behavioural public administration area on nudging healthcare professionals. A SR from the public review [13] The data [12]

ptsd A literature review [**Vandeschoot2017**] on studies that applied latent trajectory analyses on post-traumatic stress after exposure to trauma. Psychology. The corresponding dataset can be found on the Open Science Framework [. . .]

Wilson The review [2] The dataset [1]

For every SR, the raw datafiles were preprocessed into a test dataset.

These test datasets contain authors, title, abstract and annotation of whether the entry was included in the final review or not (0/1).

Entries with missing abstracts were removed. Duplicate entries were removed. Preprocessing scripts can be found on the GitHub[1]

Table 1: Statistics on datasets from original systematic reviews.

[1] https://github.com/GerbrichFerdinands/asreview-thesis

Dataset	Original study			Test collection		
	Candidate studies	Final inclusions	Inclusion rate (%)	Candidate studies	Final inclusions	Inclusion rate (%)
ace	2544	41	1.61	2235	41	1.83
nudging	2006	100	4.99	0	0	NaN
ptsd	6185	34	0.55	5031	38	0.76
software	8911	104	1.17	8896	104	1.17
wilson	3453	26	0.75	2334	24	1.03

The inclusion rate is ... data is imbalanced. what is the philosophy False negatives must be avoided ... The cost of a false negative outweighs the cost of a false positive. Note that we assume the oracle/original user to hold the truth. This is of course not always the case.

There are two classes in the data: exlusions and inclusions. The inclusions are clearly the minority class.

Models

Five different active learning models were build. Every model m was used to perform an automated systematic review on SR dataset d. The models all apply a different classifier c with its own set of (hyper)parameters h.

Classifiers The classifier predicts the class of all papers, given the labeled data set $\mathcal{L}$.

Given the labels and some features from all abstracts in the pool, the classifier To predict whether a paper should be an exclusion or an inclusion, different classifiers

- Naive Bayes (B)
- Random Forests (R)
- Support Vecor Machine (S)
- Logistic Regression (L)
- Dense Neural Network (N)

Besides c, components of m are fea

To summarize, every model m consists of the following key components: classifier, feature extraction strategy, query strategy, balance strategy.

For example M_1 BTMD (naive bayes, tfidf, certainty sampling, double balance) STMD

Word representation To be able to predict whether a paper needs to be included or excluded (e.g. to predict class), the classifier needs some features from the papers.
As features we use the title and abstract from every paper - this is what is prescribed by prisma, (check!) The classifier cannot predict the paper class from the raw titles and abstracts as they are. Therefore, the content of the texts needs to be transformed into numerical representations called feature vectors.

A classical example is a 'bag of words' representation. For each each text, 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 [15]. The bag-of-words method is simplistic and will highly value often occuring but otherwise meaningless words such as "and". Term-frequency Inverse Document Frequency (TF-IDF) [17] circumvents this problem by adjusting a term frequency in a text with the inverse docuement frequency, the frequency of a given word in the entire corpus.

(T) (D)

The next abstract to be queried The model has various ways of deciding which instance should be queried next.

Rebalancing the training set To account for class imbalance in the data, the model can apply several strategies to rebalance the training set.

A reweighting strategy is applied where inclusions (the minority class) are weighted more heavily than the exclusions.

Starting point The initial training set consists of 5 inclusions and 5 exclusions, randomly sampled from the dataset. We use 5 inclusions and exclusions as we assume the researcher has some prior knowledge on this. The researcher has some prior knowledge about the pool, some papers ought to be included in the SR.

```
c = naivebayes
f = tfidf
q = max
n_prior_included = 5
n_prior_excluded = 5
```

Retraining We can choose to retrain the model after labeling n instances. - n_instances=10 (number of papers each query)

Simulations

To evaluate performance of the five models described above, the model is used to simulate five existing systematic reviews.

Optimizing hyperparameters

For every model*data combination, 3 sets of hyperparameters are generated to ... parallel Every model has its own hyperparameters. For every model, the hyperparameters are optimized three times, arriving at three versions of the model:

We nog have 75 combinations. for every for every model (5), for every dataset (5) and for every set of optimized hyperparameters (3), a simulation study consisting trials is performed. From these $5 * 5 * 3 = 75$ simulation studies, performance of the different models is evaluated.

A simulation is of one model on one dataset. The simulation is repeated for 10 trials t.

Every simulation study consists of 10 trials, to account for the randomness of prior inclusions and exclusions. So every trial the prior inclusions and exclusions are randomly selected. Results are aggregated (?)

Assumptions

- decisions of the original SR are **ground truth** (benchmark) (oracle)
- binary classifications: relevant/irrelevant

The software

ASReview takes the following parameters/arguments:

	Configurations
Models	2-Layer Neural Network, Naive Bayes, Random Forest, Support Vector Machine, Logistic Regression
Query Strategies	Cluster Sampling, Maximum Sampling, Cluster * Maximum Sampling, Maximum * Uncertainty Sampling, Maximum * Random Sampling, Cluster * Uncertainty Sampling, Cluster * Random Sampling
Feature extraction strategies	Doc2Vec, TF-IDF, sbert, embeddingIdf

Use these inputs to predict relevance of papers.

Stage 1: hyperparameter optimization

Or, more specific:

Models	Feature extraction strategies
dense_nn	doc2vec
nb	tfidf
rf	tfidf
svm	doc2vec
lr	tfidf

Hyperparameters

Every model has its own set of hyperparameters:

Optimization

The hyperparameters are optimized on the 5 datasets in three different ways:

- 1 on 1: maximum performance

$$d = D$$

- 4 on 1: cross-validation

$$d \notin D$$

$$D = 1, 2, 3, 4$$

- 5 on 1: more data = more better?

$$d \in D$$

This results $(5 + 5 + 1) * 5$ sets of hyperparameters.

Outcomes

For each model, Several metrics are used to compare performance of different models over datasets,

Dataset	Naive Bayes	Random Forests	Support Vector Machine	Logistic Regression	Dense Neural Network
ptsd	?				
ace	?				
hall	?				
nagtegaal	?				
. . . .	?				

? How to compare outcomes of 3 different optimization strategies?

Evaluation

Results

Discussion

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). [9]. 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 [18].

```
    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 [19]

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. [23]

ASReview uses the `MultinomialNB` from the scikit-learn package [15], 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, [3] 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

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) [10]
- 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.

Performance metrics

The goal is twofold: we want to identify all relevant papers, as fast as we can. Tradeoff: identifying all relevant papers and reducing workload. A good metric to evaluate this is..

What is more important: recall or precision?

Recall more highly valued than precision.

What about class imbalance?

RRF Amount of relevant references found after having screened a certain percentage of the total number of abstracts.

Work saved over sampling (WSS) Indicates how much time can be saved, at a given level of recall. WSS is in terms of the percentage of abstracts that don't have to be screened by the researcher. Typically, WSS is measured at a recall of 0.95. Reasonable because..

$$\texttt{WSS} = \frac{TN + FN}{N} - (1 - recall)$$

Raoul

Utility?

F-measure

ROC/AUC Is performance related to some characteristic (n, inclusion rate, ...)

Cross-validation

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

Appendix B - combinations

Model	Query Strategy	Feature extraction strategy
dense_nn	cluster	doc2vec
dense_nn	max	doc2vec
dense_nn	max * cluster	doc2vec
dense_nn	max * uncertainty	doc2vec
dense_nn	max * random	doc2vec
dense_nn	cluster * uncertainty	doc2vec
dense_nn	cluster * random	doc2vec
dense_nn	cluster	tfidf
dense_nn	max	tfidf
dense_nn	max * cluster	tfidf
dense_nn	max * uncertainty	tfidf
dense_nn	max * random	tfidf
dense_nn	cluster * uncertainty	tfidf
dense_nn	cluster * random	tfidf
dense_nn	cluster	sbert
dense_nn	max	sbert
dense_nn	max * cluster	sbert
dense_nn	max * uncertainty	sbert
dense_nn	max * random	sbert
dense_nn	cluster * uncertainty	sbert
dense_nn	cluster * random	sbert
dense_nn	cluster	embeddingIdf
dense_nn	max	embeddingIdf
dense_nn	max * cluster	embeddingIdf
dense_nn	max * uncertainty	embeddingIdf
dense_nn	max * random	embeddingIdf
dense_nn	cluster * uncertainty	embeddingIdf
dense_nn	cluster * random	embeddingIdf
nb	cluster	tfidf
nb	max	tfidf
nb	max * cluster	tfidf
nb	max * uncertainty	tfidf
nb	max * random	tfidf
nb	cluster * uncertainty	tfidf
nb	cluster * random	tfidf
rf	cluster	doc2vec
rf	max	doc2vec
rf	max * cluster	doc2vec
rf	max * uncertainty	doc2vec
rf	max * random	doc2vec
rf	cluster * uncertainty	doc2vec

(continued)

Model	Query Strategy	Feature extraction strategy
rf	cluster * random	doc2vec
rf	cluster	tfidf
rf	max	tfidf
rf	max * cluster	tfidf
rf	max * uncertainty	tfidf
rf	max * random	tfidf
rf	cluster * uncertainty	tfidf
rf	cluster * random	tfidf
rf	cluster	sbert
rf	max	sbert
rf	max * cluster	sbert
rf	max * uncertainty	sbert
rf	max * random	sbert
rf	cluster * uncertainty	sbert
rf	cluster * random	sbert
rf	cluster	embeddingIdf
rf	max	embeddingIdf
rf	max * cluster	embeddingIdf
rf	max * uncertainty	embeddingIdf
rf	max * random	embeddingIdf
rf	cluster * uncertainty	embeddingIdf
rf	cluster * random	embeddingIdf
svm	cluster	doc2vec
svm	max	doc2vec
svm	max * cluster	doc2vec
svm	max * uncertainty	doc2vec
svm	max * random	doc2vec
svm	cluster * uncertainty	doc2vec
svm	cluster * random	doc2vec
svm	cluster	tfidf
svm	max	tfidf
svm	max * cluster	tfidf
svm	max * uncertainty	tfidf
svm	max * random	tfidf
svm	cluster * uncertainty	tfidf
svm	cluster * random	tfidf
svm	cluster	sbert
svm	max	sbert
svm	max * cluster	sbert
svm	max * uncertainty	sbert
svm	max * random	sbert
svm	cluster * uncertainty	sbert
svm	cluster * random	sbert
svm	cluster	embeddingIdf
svm	max	embeddingIdf
svm	max * cluster	embeddingIdf
svm	max * uncertainty	embeddingIdf

(continued)

Model	Query Strategy	Feature extraction strategy
svm	max * random	embeddingIdf
svm	cluster * uncertainty	embeddingIdf
svm	cluster * random	embeddingIdf
lr	cluster	doc2vec
lr	max	doc2vec
lr	max * cluster	doc2vec
lr	max * uncertainty	doc2vec
lr	max * random	doc2vec
lr	cluster * uncertainty	doc2vec
lr	cluster * random	doc2vec
lr	cluster	tfidf
lr	max	tfidf
lr	max * cluster	tfidf
lr	max * uncertainty	tfidf
lr	max * random	tfidf
lr	cluster * uncertainty	tfidf
lr	cluster * random	tfidf
lr	cluster	sbert
lr	max	sbert
lr	max * cluster	sbert
lr	max * uncertainty	sbert
lr	max * random	sbert
lr	cluster * uncertainty	sbert
lr	cluster * random	sbert
lr	cluster	embeddingIdf
lr	max	embeddingIdf
lr	max * cluster	embeddingIdf
lr	max * uncertainty	embeddingIdf
lr	max * random	embeddingIdf
lr	cluster * uncertainty	embeddingIdf
lr	cluster * random	embeddingIdf

References

[1] Christian Appenzeller-Herzog. *Data from Comparative Effectiveness of Common Therapies for Wilson Disease: A Systematic Review and Meta-analysis of Controlled Studies*. Zenodo, Jan. 2020.

[2] Christian Appenzeller-Herzog et al. “Comparative Effectiveness of Common Therapies for Wilson Disease: A Systematic Review and Meta-Analysis of Controlled Studies”. en. In: *Liver International* 39.11 (2019), pp. 2136–2152. ISSN: 1478-3231. DOI: 10.1111/liv.14179.

[3] Leo Breiman. “Random Forests”. en. In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324.

[4] A.M. Cohen et al. “Reducing Workload in Systematic Review Preparation Using Automated Citation Classification”. In: *Journal of the American Medical Informatics Association : JAMIA* 13.2 (2006), pp. 206–219. ISSN: 1067-5027. DOI: 10.1197/jamia.M1929.

[5] Gordon V. Cormack and Maura R. Grossman. "Evaluation of Machine-Learning Protocols for Technology-Assisted Review in Electronic Discovery". In: *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR '14. Gold Coast, Queensland, Australia: Association for Computing Machinery, July 2014, pp. 153–162. ISBN: 978-1-4503-2257-7. DOI: 10.1145/2600428.2609601.

[6] Tivadar Danka and Peter Horvath. "modAL: A Modular Active Learning Framework for Python". In: ().

[7] Tracy Hall et al. "A Systematic Literature Review on Fault Prediction Performance in Software Engineering". In: *IEEE Transactions on Software Engineering* 38.6 (Nov. 2012), pp. 1276–1304. ISSN: 2326-3881. DOI: 10.1109/TSE.2011.103.

[8] H. Kilicoglu et al. "Towards Automatic Recognition of Scientifically Rigorous Clinical Research Evidence". en. In: *Journal of the American Medical Informatics Association* 16.1 (Jan. 2009), pp. 25–31. ISSN: 1067-5027, 1527-974X. DOI: 10/bjkhh9.

[9] Quoc V. Le and Tomas Mikolov. "Distributed Representations of Sentences and Documents". en. In: *arXiv:1405.4053 [cs]* (May 2014). arXiv: 1405.4053 [cs].

[10] David D. Lewis and Jason Catlett. "Heterogeneous Uncertainty Sampling for Supervised Learning". en. In: *Machine Learning Proceedings 1994*. Ed. by William W. Cohen and Haym Hirsh. San Francisco (CA): Morgan Kaufmann, Jan. 1994, pp. 148–156. ISBN: 978-1-55860-335-6. DOI: 10.1016/B978-1-55860-335-6.50026-X.

[11] Makoto Miwa et al. "Reducing Systematic Review Workload through Certainty-Based Screening". en. In: *Journal of Biomedical Informatics* 51 (Oct. 2014), pp. 242–253. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2014.06.005.

[12] Rosanna Nagtegaal et al. *Nudging Healthcare Professionals towards Evidence-Based Medicine: A Systematic Scoping Review*. Version V1. tyep: dataset. 2019.

[13] Rosanna Nagtegaal et al. "Nudging Healthcare Professionals towards Evidence-Based Medicine: A Systematic Scoping Review". In: *Journal of Behavioral Public Administration* 2.2 (2019). DOI: doi.org/10.30636/jbpa.22.71.

[14] Alison O'Mara-Eves et al. "Using Text Mining for Study Identification in Systematic Reviews: A Systematic Review of Current Approaches". In: *Systematic Reviews* 4.1 (Jan. 2015), p. 5. ISSN: 2046-4053. DOI: 10.1186/2046-4053-4-5.

[15] F. Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[16] PRISMA-P Group et al. "Preferred Reporting Items for Systematic Review and Meta-Analysis Protocols (PRISMA-P) 2015 Statement". en. In: *Systematic Reviews* 4.1 (Dec. 2015), p. 1. ISSN: 2046-4053. DOI: 10.1186/2046-4053-4-1.

[17] Juan Ramos et al. "Using Tf-Idf to Determine Word Relevance in Document Queries". In: *Proceedings of the First Instructional Conference on Machine Learning*. Vol. 242. Piscataway, NJ. 2003, pp. 133–142.

[18] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[19] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks". In: *arXiv:1908.10084 [cs]* (Aug. 2019). arXiv: 1908.10084 [cs].

[20] Byron C. Wallace et al. "Semi-Automated Screening of Biomedical Citations for Systematic Reviews". In: *BMC Bioinformatics* 11.1 (Jan. 2010), p. 55. ISSN: 1471-2105. DOI: 10.1186/1471-2105-11-55.

[21] Zhe Yu, Nicholas A. Kraft, and Tim Menzies. "Finding Better Active Learners for Faster Literature Reviews". In: *Empirical Software Engineering* 23.6 (Mar. 2018), pp. 3161–3186. ISSN: 1573-7616. DOI: 10.1007/s10664-017-9587-0.

[22] Zhe Yu and Tim Menzies. “FAST2: An Intelligent Assistant for Finding Relevant Papers”. en. In: *Expert Systems with Applications* 120 (Apr. 2019), pp. 57–71. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2018.11.021.

[23] Harry Zhang. “The Optimality of Naive Bayes”. In: *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004*. Vol. 2. Jan. 2004.