Methods

2021-10-20

Methods

General description

We built an R (R Core Team, 2020) based framework with the goal of simplifying two aspects of systematic reviews: record acquisition and classification. The framework is composed of several modules which communicate through intermediate outputs stored in common formats, which make it possible for users to extend the framework or easily integrate it with other tools in their pipeline. See Supplemental Material S1 for an in-depth description of the framework and how to use it.
The tasks carried out by the framework are grouped into "sessions," i.e., a set of actions that starts from a search query to obtain a set of scientific citation data (records), which is then labelled as relevant ("positive" in the rest of the text) or not ("negative") for the topic of interest (Fig. 1). From this labelled set, the framework can generate a new query and perform a new session to find records possibly missed by the first query.
The researcher initiate the process with a starting query derived by domain knowledge from which she expects a high relevant/non-relevant record ratio.
Follows a description of the framework's components.

Record's acquisition and initial labelling

We built a set of tools to let users automatically search and download citation data from three major scientific databases ("sources"): MEDLINE (https://pubmed.ncbi.nlm.nih.gov/), Web Of Science (WOS, https://apps.webofknowledge.com/) and the Institute of Electrical and Electronics Engineers (IEEE, https://ieeexplore.ieee.org/Xplore/home.jsp). The framework takes care of authorization management for non-open databases like WOS and IEEE. It is also possible to download and import records in the framework manually. This is particularly useful to acquire records from the SCOPUS (https://www.scopus.com/search/form.uri?display=basic#basic) and EMBASE databases (https://www.embase.com/#advancedSearch/default), for which a comprehensive API interface was not easy to build. A short guide on how to set up the framework for each supported database is available in Supplemental Material S3.
The acquired records are merged into a single database, resolving duplicates and different formatting between sources. The records are ordered according to the frequency of the positive query terms (e.g., not preceded by a *NOT* modifier) in the title and abstract ("simple query ordering").

The researches is then asked to label a number of records to create the "initial training set" needed to start the automatic classification. We suggest to manually label the first 250 records (see "hyperparameter optimization" later). The simple query ordering increases the positivity rate in the initial training set (Wallace et al., 2010), which provide higher sensitivity during automatic classification (Chawla et al., 2004).

Text feature extraction

The collected citation data have a number of fields characterizing a scientific publication. The framework models the relevance of a record based on the following fields: title, abstract, authors, keywords, MESH terms (Lipscomb, 2000). A series of Natural Language Processing (NLP) techniques (Ananiadou & McNaught, 2006; Baeza-Yates et al., 1999; Marshall & Wallace, 2019) are employed to transform the textual information

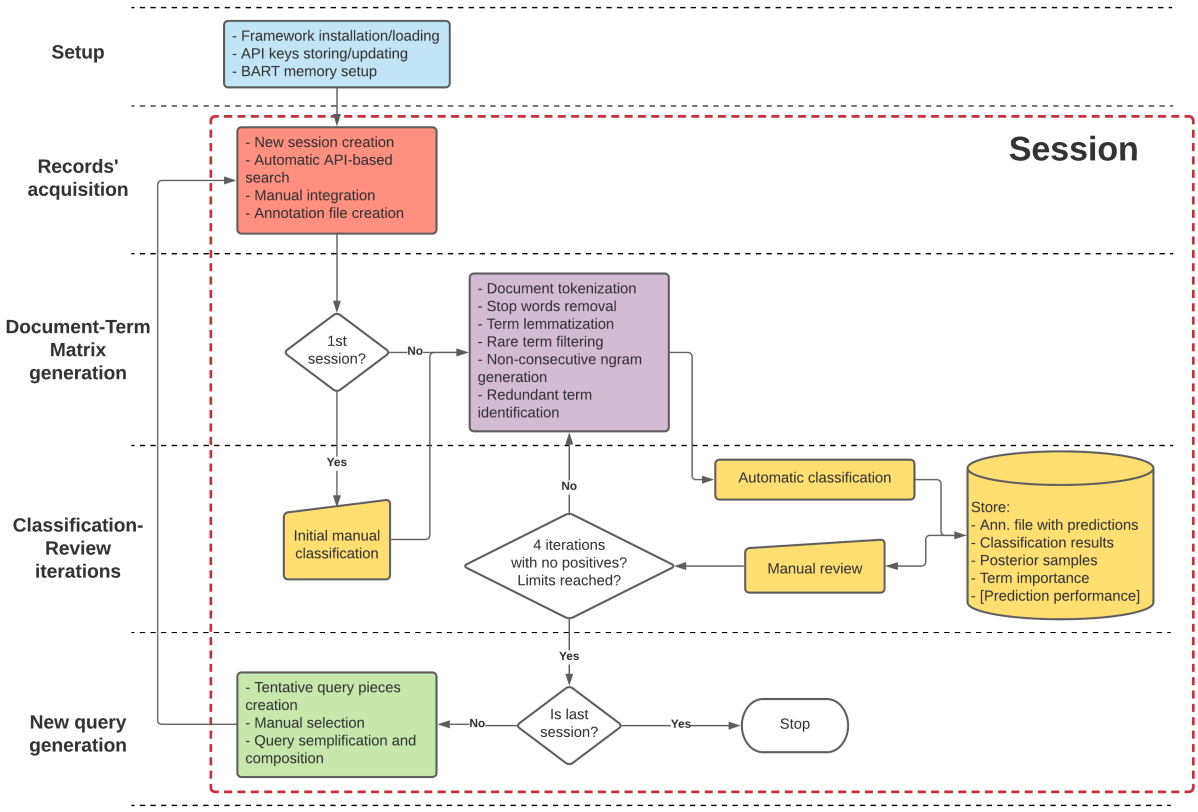

Figure 1.Framework's visual depiction.

in these fields into features for machine learning through a bag-of-words approach (Marshall & Wallace, 2019). The processing of free text fields (title, abstract) includes tokenization (i.e., extracting the terms), common stopwords (i.e. sentence components bringing no meaning) removal, part-of-speech filtering (only nouns, adjectives, verbs and untagged terms are retained), and lemmatization of the terms (i.e. reduction to their base grammar form). Text processing for authors, keywords and MESH terms identifies logical units (e.g., author's full names, composite keywords) and extracts them.
Terms appearing in less than 5% of the labelled documents are removed from negative records. All terms in the positive set are kept to increase sensitivity at the cost of specificity.
Some terms tend to co-appear in records (non-consecutive ngrams, nc-ngrams), often carrying a particular meaning when copresent. To detect nc-ngrams, we generated a word network representation (Rousseau, 2015) posing edges between terms with a cosine similarity in terms of record co-occurrence > 0.5. We extracted the maximal cliques in the network (Eppstein et al., 2010) representing highly correlated groups of terms; These groups terms are added to the data set as individual feature. Only nc-ngrams comprising a maximum of ten terms are kept.
A second network is built using a co-occurrence threshold of 0.9. In this case, the cliques represent terms that always appear together and therefore can be considered redundant (i.e. they don't need to be considered separately). These terms are merged to increase computation efficiency and reduce overfitting.
The output is a Document-Term Matrix (DTM), with N_d rows representing the records (D_i), N_t terms column for the t_{field} terms (divided by record field) and $0, 1$ values whether $t_{field} \in D_i$. We also enriched the DTM with features referencing the number of terms in each field to help the model scale term importance based on the field length.

Label prediction

We used a Bayesian Additive Regression Trees (BART) machine learning model (Chipman et al., 2010) (in the implementation of Kapelner & Bleich, 2013) to predict the probability of a record of being relevant, given the information coded into the enriched DTM. We set up the BART model to use 2000 MCMC iterations (after 250 burn-in iterations) and 50 trees; we used a k value of 2 to regularized extreme prediction and let the model use missing fields in the DTM as features (Kapelner & Bleich, 2015). Since the goal is to find all positive matches (i.e., focus on sensitivity), positive records are oversampled ten times.
The output is a posterior predictive distribution (PPD) of each record's probability of being relevant (i.e., a positive match). An ensemble of ten models was fitted to improve prediction stability by averaging the PPD between models (Dietterich, 2000; Zhou, 2021).

To assign the labels we employed an "active learning" Miwa et al. (2014) approach, where a human reviews a specific subset of predictions made from the machine, which is then retrained on the manually reviewed dataset. This process proceed iteratively, decreasing prediction uncertainty.
Label assignment is done through the identification of an "uncertainty zone," whose construction is possible due to the Bayesian nature of BART which provides full PPDs instead of point-wise predictions for each record.
To describe the process formally, we define

$$\pi_i = \frac{1}{M} \sum_{j=1}^{M} Pr(L_i = 1 | DTM, m_j)$$

as the PPD of a record D_i being relevant (i.e, having a positive label, $L_i = 1$), averaging the PPDs of the ensemble of $M = 10$ models m, and

$$\pi_{i,l} = \{\pi_i : Pr(\pi_i) = 1\%\}$$
$$\pi_{i,u} = \{\pi_i : Pr(\pi_i) = 99\%\}$$

as respectively the lower and upper boundaries of the 98% quantile interval of π_i (98% Predictive Interval,

98% PrI).
Then we identify the "uncertainty zone" as

$$U_\pi = [\max \vec{\pi}_u^-, \min \vec{\pi}_l^+]$$

with $\vec{\pi}_u^-$ being the vector of $\pi_{i,u}$ with a negative label and $\vec{\pi}_l^+$ the vector of $\pi_{i,l}$ with a positive label. That is, U_π defines a range of values between the smallest $\pi_{i,l}$ in the set of already labelled positive records L_p and the largest $pi_{i,u}$ related to the negative ones L_n, noting that the two limits can appear in any order.
Consequently, a record D_i will be labelled as positive if

$$\pi_{i,l} > \max_{\pi \in U_\pi} \pi$$

that is, the record lower 98% PrI boundary should be higher than every value in the uncertainty zone. In other words, for a record to be labelled positive, its PPD should be within the range of the mixture of PPD of the previously labelled positive records and not cross the distributions of the negative records.
Conversely, a record is labelled as negative if

$$\pi_{i,u} < \min_{\pi \in U_\pi} \pi$$

All other records are labelled as "uncertain."
Manual review and labelling is then necessary for: 1) uncertain records, 2) positive records (to avoid false positives), 3) records whose predicted label differs from the existing manual one. This last rules helps identifying human errors or inconsistent labelling criteria.

The automatic classification task and the manual review step continue in loop (CR iterations) until no new positive matches are found in four consecutive iterations.

New search query generation

We created an algorithm that generates a new search query to acquire further relevant publications missed during the first search, possibly at a reasonable cost in specificity (i.e., a higher number of negative results). The algorithm encompasses a number of steps:

- We fit a partition tree (Therneau & Atkinson, 2019) between the DTM and 800 samples from the PPD; if a term is present multiple times in the DTM (e.g. both title and abstract), they are counted just one, and field term count features are removed. This step generates a list of rules composed by AND/NOT "conditions" made of terms/authors/keywords/MESH tokens, which together identify a group of records.
- For each rule, negative conditions (i.e., NOT statements) are added iteratively starting from the most specific one, until no conditions are found that would not also remove positive records.
- The extended set of rules is sorted by positive-negative record difference in descending order. The cumulative number of unique positive records is computed and used to group the rules. Rules inside the each group are ordered by specificity.
- The researcher is then asked to review the rule groups, selecting one or more rules (useful if they convey different meaning) from each, or edit them (in case too specific positive or negative conditions were included). It is possible to exclude a group of rules altogether, especially the those with the worse sensitivity/specificity ratio.
- The selected rules are joined together by OR statements, defining a subset of records with a sensibly higher proportion of positive records than the original one.
- Redundant rules (i.e., rules whose positive records are already included in more specific ones) and conditions (i.e., conditions that once removed do not decrease the total number of positive or do not increase the negative records) are removed.
- Finally, the rules are re-elaborated in a query usable on the major scientific databases.

Since the algorithm is data-driven, it creates queries that are effective in selecting positive records the input dataset but may be not specific enough once applied to actual research databases. Therefore we appended an extra subquery in AND, which specifies the general topics of our search and delimitates the search domain. The new query was used to initiate a second search session.

Performance evaluation

Each literature corpus is unique per database/search query, so we did noWe focused on estimating the expected total sensitivity in the dataset, using a surrogate [] Bayesian logistic regression to model the record labels on the lower bound of the [98% PrI] produced by our algorithm and then using this last model to produce a predictive cumulative distribution of the number of missed positive records. For this model, we used weakly regularizing, robust priors for the intercept (Student T with $\nu = 3, \mu = 0, \sigma = 2.5$) and the linear coefficient (Student T with $\nu = 3, \mu = 0, \sigma = 1.5$). Given that this model is conditional only on the BART predictions and not on the DTM, it is characterized by more uncertainty, providing a plausible worst-case scenario. The quality of the model was evaluated through Bayesian R^2 (Gelman et al., 2019) of which we reported the posterior median and 90% Credible Interval [90% CrI].
The predictive distribution of the number of missed positive records allows to estimate the expected long-run *Sensitivity* and the *Work saved over random* (WSoR) of the algorithm. The WSoR is based on a negative hypergeometric model to estimate the number of records to manually label to find the same number of positives if records were evaluated in random order (Chae, 1993); the WSoR is then one minus the ratio of the reviewed records over this number. For the number of predicted positive records, the sensitivity and WSoR, we reported the truncated 90% PrI [trunc. 90% PrI], which is the uncertainty interval bounded at the number of observed total positive records: since each positive match is manually verified, the probability of a number of total positive records lower than observed is zero.

Hyperparameter evaluation

Our classification algorithm has a number of hyperparameters:

- The number of ensemble models;
- The source of randomness between models in the ensemble, that is, either MCMC sampling only (Robert et al., 2004), or MCMC plus data bootstrapping (Breiman, 1996) before training;
- The oversampling rate of positive records;
- The PrI quantiles to define the uncertainty zone;
- The size of the initial training set.

To evaluate the hyperparameter effect of performance, we set up a "grid search" (Claesen & De Moor, 2015; Yang & Shami, 2020) on a prelabelled subset made of the first 1200 records from the first session dataset. See Supplemental Material S1 for the combinations of hyperparameter used and other details on the method. The framework tested each hyperparameter combination until four CR iterations with no positive records were returned or the whole dataset got labelled.
For each combination, a performance score was computed as the product of *Efficiency* (1 minus the ratio of records that required review over the total) and *Sensitivity* (number of positive records found over the total of positives). We then identified homogeneous "performance clusters" of hyperparameter values and scores using a partition tree. For the rest of our study we chose the best cluster of parameter value combinations and, inside the cluster, the best combination according to Sensitivity and Efficiency in this order.

Ananiadou, S., & McNaught, J. (2006). *Text mining for biology and biomedicine*. Citeseer.

Baeza-Yates, R., Ribeiro-Neto, B., & others. (1999). *Modern information retrieval* (Vol. 463). ACM press New York.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Chae, K.-C. (1993). Presenting the negative hypergeometric distribution to the introductory statistics courses. *International Journal of Mathematical Education in Science and Technology*, *24*(4), 523–526.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, *6*(1), 1–6.

Chipman, H. A., George, E. I., McCulloch, R. E., & others. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, *4*(1), 266–298.

Claesen, M., & De Moor, B. (2015). Hyperparameter search in machine learning. *arXiv Preprint arXiv:1502.02127*.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*, 1–15.

Eppstein, D., Löffler, M., & Strash, D. (2010). Listing all maximal cliques in sparse graphs in near-optimal time. *International Symposium on Algorithms and Computation*, 403–414.

Gelman, A., Goodrich, B., Gabry, J., & Vehtari, A. (2019). R-squared for bayesian regression models. *The American Statistician*.

Kapelner, A., & Bleich, J. (2013). bartMachine: Machine learning with bayesian additive regression trees. *arXiv Preprint arXiv:1312.2171*.

Kapelner, A., & Bleich, J. (2015). Prediction with missing data via bayesian additive regression trees. *Canadian Journal of Statistics*, *43*(2), 224–239.

Lipscomb, C. E. (2000). Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, *88*(3), 265.

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*, 1–10.

Miwa, M., Thomas, J., O'Mara-Eves, A., & Ananiadou, S. (2014). Reducing systematic review workload through certainty-based screening. *Journal of Biomedical Informatics*, *51*, 242–253.

R Core Team. (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Robert, C. P., Casella, G., & Casella, G. (2004). *Monte carlo statistical methods* (Vol. 2). Springer.

Rousseau, F. (2015). *Graph-of-words: Mining and retrieving text with networks of features* [PhD thesis]. Ph. D. dissertation.

Settles, B. (2009). *Active learning literature survey*.

Therneau, T., & Atkinson, B. (2019). *Rpart: Recursive partitioning and regression trees*. https://CRAN.R-project.org/package=rpart

Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2010). Active learning for biomedical citation screening. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 173–182.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316.

Zhou, Z.-H. (2021). Ensemble learning. In *Machine learning* (pp. 181–210). Springer.