Methods

2021-08-31

Methods

General description

We built an R (R Core Team, 2020) based framework with the aim of simplifying two aspects of systematic reviews: record acquisition and relevance classification. The framework is composed by several components which act together while being independent enough to be in principle be substituted by alternative implementations, given that the structure of the intermediate data outputs is respected. Check 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," that is a set of actions that starts from a search query with which collect a set of unlabelled records and ends with the having a fully labelelled set (Fig. 1). From this labelled set, the framework allows to generate a new query and perform a new session.
It is advisable that the research starts using the framework with a specific query from which she expects a high relevant/non-relevant record ratio.
Follows a description of the framework's components.

Record's acquisition and inital labeling

We built a set of tools to let user automatically search and download records data from three major scientific databases: 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 user needs to input a search query and a date range. The query may contain boolean operators AND, OR, NOT and nested parentheses. The dabatase will be also called "sources" in the rest of the text.
For WOS an Application Programming Interface (API) key is necessary to use the automatic search tools; for IEEE, if an API key is not available, a slower, webscraping-based solution will be employed; for MEDLINE the API key is required for high frequency requests to the NCBI server (Sayers, 2010), which may happen if the chosen query produces a large number of records, since our tool splits a big API requests in multiple smaller ones.
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; the framework will be able to import the manually downloaded results seamlessly.A short guide on how to setup the search for each supported database is available in Supplemental Material S3.
Once the records are downloaded and acquired, the framework merges them into a single database, resolving duplicates and different formatting between sources, and ordering the records by simple query term frequency, putting the most likely relevant on top. The output is an "Annotation file."

To allow the automatic classification of the records, a first initial manual input is needed. We suggest to manually label as relevant ("positive" in the rest of the text) or not ("negative") the first 250 records (see "hyperparameter optimization" later).

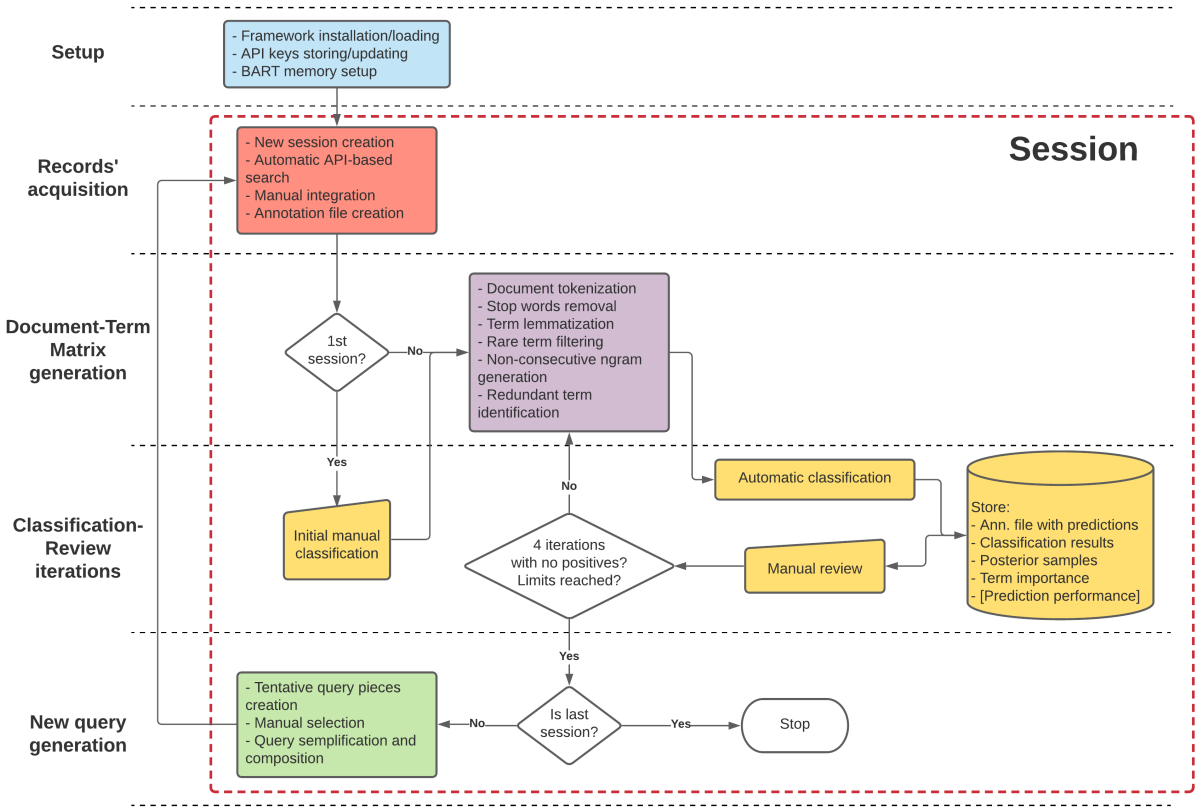

Figure 1: Figure 1. Framework visual depiction.

Document-Term matrix generation

The annotation file produced in the previous step has a number of fields describing a publication. We use the following fields to model its relevance: title, abstract, authors, keywords, MESH terms (Lipscomb, 2000). We use a series of Natural Language Processing (NLP) techniques (*should I put a general reference?*) to transform the textual information in these fields into machine learning features. The first steps are tokenizing the text (i.e., separate the terms), remove common stopwords (i.e. grammar term bringing no meaning) and lemmatizing the remaining terms (i.e. reduce them to their base grammar form). Tokenization for authors, keywords and MESH terms works differently than for title and abstract text, since it keep a whole unit without breaking it in its single terms.
To reduce noise and save computation time, terms that appeared in less than 5% of the corpus of labelled documents (positive and negatives) were removed from those labelled as negative. All terms in the positive set are kept to increase sensitivity at the cost of specificity.
Some terms tend to co-appears in records even if not consecutively in the text (non-consecutive ngrams, nc-ngrams); when they do so they often carry a particular meaning. To detect these terms we generated a network representation of all terms based on their cosine similarity in terms of co-presence in a document, keeping only pairs with cosine similarity > 0.5. From this network we extracted the maximal cliques (Eppstein et al., 2010), that is group of terms which all tend to appear together often; These generated terms are added to the dataset. To avoid overfitting, we kept nc-ngrams of maximum 10 terms.
The similarity network is rebuilt also considering the nc-ngrams, this time using a similarity of .9 as threshold and finding the cliques again. In this case, the cliques represent terms that always appear together and therefore can be considered synonyms. These terms are merged in the dataset to increase computation efficiency and reduce overfitting.
The final output is a matrix, also called a Document-Term Matrix (DTM), with N_d rows for each record D, N_t terms column for each t_{field} (divided by record field) and $0, 1$ values whether $t_{field} \in D$. We also enriched the DTM with a feature for each record field recording the number of terms in each, to relate the terms relative importance (e.g., the same term can have different relevance given on the length of a document).

Label prediction

We used a Bayesian Additive Regression Trees (BART) model (Chipman et al., 2010) (in the implementation of Kapelner & Bleich, 2013) to model the probability of a record of being relevant for the systematic review, given the information coded into the enriched DTM. BART models have a number of advantages; as other boosted trees techniques (Hastie et al., 2009) they can model complex non-linearities, perform variable selection, manage missing data while sporting high performance in predictive power. The Bayesian framework they are built on provides further advantages, namely, less sensitivity on hyperparameter choices, natural regularization, and, most of all, predictive distributions as output in place of point-wise predictions.
We setup the BART model to use 2000 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 for our purpose it is more important to find all positive matches (i.e., focus on sensitivity).
The output is a posterior predictive distribution of the probability of a positive match for each record. To further improve prediction stability, an ensemble of ten models was fitted, averaging the predictions between models.
To choose how to label a record we exploit the uncertainty typical of Bayesian estimates, assigning a positive label to records whose posterior distribution of the probability of being positive ends up almost totally in the range of probabilities of records already labelled as positives, without crossing the range of the records previously labelled as negatives.

To describe the process formally, first we define

$$\pi_i = Pr(L_{D_i} = \text{pos} | DTM, model)$$

as the posterior probability of a record D_i of being assigned a positive label (L_{D_i}) 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 (predictive interval, PrI). A record will be labelled as positive if

$$\pi_{i,l} > max(\max_{i:L_{D_i}=\text{neg}} \pi_{i,u}, \min_{i:L_{D_i}=\text{pos}} \pi_{i,l})$$

with L being a manually assigned/reviewed label. That is, to be positive, a record lower 98% PrI boundary should be higher than both the highest among the upper boundaries of negative records and the lowest of boundaries of the positive ones (*I wonder whether a picture could help...*), in other words, its PrI should be included in the mixture of distributions of the already labbeled positive records and not cross those of the negative ones.
Conversely, a record is labelled as negative if

$$\pi_{i,u} < min(\min_{i:L_{D_i}=\text{pos}} \pi_{i,l}, \max_{i:L_{D_i}=\text{neg}} \pi_{i,u})$$

All other records are labelled as urcertain, because their PrI crosses the "uncertainty zone" defined as:

$$U = \pi : \pi \in [\min_{i:L_{D_i}=\text{pos}} \pi_{i,l}, \max_{i:L_{D_i}=\text{neg}} \pi_{i,u}]$$

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.

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.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Boosting and additive trees. In *The elements of statistical learning* (pp. 337–387). Springer.

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.

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

Sayers, E. (2010). A general introduction to the e-utilities. *Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US)*.