A novel application of machine learning and zero-shot classification methods for automated abstract screening in systematic reviews.

MORENO-GARCIA, C.F., JAYNE, C., ELYAN, E. and ACEVES-MARTINS, M.

2023

Decision Analytics Journal 6 (2023) 100162

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

A novel application of machine learning and zero-shot classification methods for automated abstract screening in systematic reviews

Carlos Francisco Moreno-Garcia [a,*], Chrisina Jayne [b], Eyad Elyan [a], Magaly Aceves-Martins [c]

[a] School of Computing, Robert Gordon University, Garthdee Road, Aberdeen, AB10 7QB, Scotland, UK
[b] Teesside University, Southfield Road, Middlesbrough, TS1 3BX, England, UK
[c] The Rowett Institute, University of Aberdeen, Ashgrove Road West, Aberdeen, AB25 2ZD, Scotland, UK

ARTICLE INFO

Keywords:
Machine learning
Systematic review
Abstract screening
Class imbalance
Zero-shot classification

ABSTRACT

Zero-shot classification refers to assigning a label to a text (sentence, paragraph, whole paper) without prior training. This is possible by teaching the system how to codify a question and find its answer in the text. In many domains, especially health sciences, systematic reviews are evidence-based syntheses of information related to a specific topic. Producing them is demanding and time-consuming in terms of collecting, filtering, evaluating and synthesising large volumes of literature, which require significant effort performed by experts. One of its most demanding steps is abstract screening, which requires scientists to sift through various abstracts of relevant papers and include or exclude papers based on pre-established criteria. This process is time-consuming and subjective and requires a consensus between scientists, which may not always be possible. With the recent advances in machine learning and deep learning research, especially in natural language processing, it becomes possible to automate or semi-automate this task. This paper proposes a novel application of traditional machine learning and zero-shot classification methods for automated abstract screening for systematic reviews. Extensive experiments were carried out using seven public datasets. Competitive results were obtained in terms of accuracy, precision and recall across all datasets, which indicate that the burden and the human mistake in the abstract screening process might be reduced.

1. Introduction

Review articles are common and crucial sources of knowledge across different domains. They provide a comprehensive study of an area (e.g. health-related topics, medical interventions, social sciences, etc.) and serve as a rich source of information for researchers in the respective field. In health sciences, but also in other domains such as social sciences [1], a systematic review (SR) approach is often followed to construct such comprehensive studies aiming to collect, summarise, critically evaluate, and synthesise knowledge. This practice is crucial to keeping clinicians and medical experts informed of the latest development in the field. The type of SR will depend on the research question and the type of data available to answer such questions. Based on the data, there are different approaches to conducting SRs [2]. Typical examples include the work presented by Aceves-Martins et al. [3], where the authors consulted different databases for articles exploring the relationship between obesity and oral health in Mexican children.

Most SR approaches consist of the following steps [4]:

1. identify relevant databases of published peer-reviewed literature
2. use specific keywords and Boolean connectors to search for potentially relevant papers
3. screen titles and abstracts retrieved from the searches
4. read full-text papers from relevant abstracts to identify those meeting the inclusion criteria
5. extract, analyse and synthesise data from included full-text papers

Producing SRs is a demanding task in terms of collecting, filtering and evaluating large volumes of literature. Typically, this task will require many steps performed by researchers. However, with the growing amount of literature, and the recent advances in the domain of machine learning (ML) and deep learning (DL) research, especially in the area of natural language processing (NLP) [5], it becomes possible to automate or semi-automate this task [6]. Several studies have been presented showing how to use ML-based methods along with NLP techniques to

[*] Corresponding author.
E-mail addresses: c.moreno-garcia@rgu.ac.uk (C.F. Moreno-Garcia), c.jayne@tees.ac.uk (C. Jayne), e.elyan@rgu.ac.uk (E. Elyan), magaly.aceves@abdn.ac.uk (M. Aceves-Martins).
URLs: http://cfmgcomputing.blogspot.com/p/home.html (C.F. Moreno-Garcia), https://research.tees.ac.uk/en/persons/chrisina-jayne (C. Jayne), https://rgu-repository.worktribe.com/person/74625/eyad-elyan (E. Elyan), https://www.abdn.ac.uk/hsru/who-we-are/people/profiles/magaly.aceves (M. Aceves-Martins).

https://doi.org/10.1016/j.dajour.2023.100162
Received 4 September 2022; Received in revised form 3 January 2023; Accepted 7 January 2023
Available online 11 January 2023

automate or reduce the workload of producing systematic reviews [4]. A comprehensive review of the latest developments in the NLP area and how it is applied to speed up the process of screening papers and other tasks relevant to building SR can be found in O'Mara-Eves et al. [7] and, most recently, by Blaizot et al. [8], Kebede et al. [9] and Khalil et al. [10], with a particular focus on the latest software tools released for this purpose.

Automatic screening and classification of literature are becoming increasingly important, as they might be one of the key indicators of a quality SR [11]. In automated literature classification tasks, first, the text (e.g. abstract, title, or parts of the paper) is preprocessed and transformed into a vector representation with corresponding labels (e.g. relevant, not-relevant). Then, a supervised ML model is trained to identify the relevant literature based on certain criteria. Typical models include Support Vector Machine (SVM), Random Forest (RF), Neural Networks (NNs) and others [12].

Various tools already exist in the public domain that aims at reducing the time to screen abstracts for inclusion in SRs. A typical example is *Abstrackr*, which was also used by Gates et al. [13] to semi-automate title and abstract screening to construct SRs. Interestingly, the authors reported that performance in relation to workload saving varies based on the screening tasks. Cleo et al. [14] presented an experimental study to assess the usability and accessibility of four common tools that are used to perform key steps in the creation of SRs and reported that all the tools were found to be easy to use and helpful by participants.

Yu et al. [15] presented a method to create SR of genetic association and human genome discoveries. The authors used SVM and Logistic Regression (LR) to perform text classification, identify the relevant literature, and reported comparable results to human performance (97% recall, 98.3% specificity). However, low precision of 31.9% was observed. Results also showed that SVM outperforms LR in this task.

Similarly, Cohen et al. [4] used supervised classification models to reduce the workload required to review papers for inclusion in SRs of drug class efficacy. In order to test their approach, the authors built a dataset for testing using 15 review topics, used traditional-based features representation to represent the data (e.g. bag-of-words) and used a voting perception-based algorithm to train a classifier. Interestingly, results showed that the classifier's performance varies from one topic to another.

Marshall et al. [16] trained an SVM, a convolutional neural network (CNN) and an ensemble-based method on a set of titles and abstracts of literature related to randomised controlled trials and compared their methods against traditional database search filters. The authors reported that ML-based approaches outperformed other traditional search methods in identifying relevant papers.

Przybyla et al. [17] presented a large experimental study to create SRs using a web-based tool called *RobotAnalyst*, that combines text-mining and ML functionalities. The authors used more than 43,610 abstracts (mostly related to health topics) and reported significant gains in terms of time needed to screen abstracts improvement over random sampling.

In another large-scale experiment, Xiong et al. [18] used ML-based methods to select and identify relevant papers out of a collection of 4177. The topics that were covered related to diabetes mellitus and its association with atrial fibrillation. Results showed that the 29 studies identified by ML-based methods were completely consistent with the manual screening.

Karasalo et al. [19] presented a framework for *horizon scanning*, aiming at discovering changes, and trends with the potential to influence a particular area. Using general search criteria, the authors used a top-down approach to scan and collect literature (from Thomson. Reuters Web of Science). Then Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) was used to cluster the collected literature into different topics, and finally, articles within each cluster were ranked based on scientific citation statistics (e.g. citations in a short period of time, strong citation trend, being cited in top journals) to identify the most significant contributions.

To speed up the creation of SRs, Pradhan et al. [20] developed a framework to extract quantitative data from relevant studies. The authors used *clinicalTrials.gov* to evaluate the performance of the developed methods and reported 100% when compared with a manual approach. However, only three published reviews were used for testing. Other most recent approaches found in the literature involve the famous Bidirectional Encoder Representations from Transformers (BERT) model, using a knowledge graph corpus specifically built for medical text classification in Chinese texts [21].

In this work, we propose a novel application of traditional ML and zero-shot classification methods for the task of automated abstract screening for SR. Applying directly traditional ML methods to classify datasets of abstracts for SR is not trivial as these datasets are very highly imbalanced. Typically a small number of abstracts will be labelled for inclusion in the SR, and the rest will be labelled to be excluded. In this paper, we use a recent novel CDSMOTE method presented in [22] to address the imbalanced problem and apply ML traditional methods. We also propose a novel application of zero-shot text classification [23] approach to the problem of automated abstract screening for SR. This method of classifying text does need any training. To improve further the classification results, we explore and evaluate a hybrid approach, where we combine zero-shot classification and traditional machine learning algorithms. We conduct a number of experiments and evaluate the proposed methods on several datasets from different SRs in diverse areas of health sciences.

The structure of the paper is as follows. Section 2 presents the proposed methods, Section 3 describes the prepossessing steps of converting the abstract into text embeddings, datasets, and the CDSMOTE method used to tackle the imbalance in the datasets, Section 4 covers the experimental work, Section 5 provides the results and discussion, and Section 6 gives the conclusion.

2. Methods

2.1. Traditional ML methods

A wide range of supervised ML algorithms can be applied to map an instance x to a particular class label y. In this paper, and based on the experiments carried out in the past by Fernández-Delgado et al. [24], the aim is to use two widely adopted ML algorithms for classification: Random Forest (RF) and Support Vector Machine (SVM). These methods need to be applied to classify abstracts for inclusion or exclusion in an SR. Still, the abstract texts have to be first converted into vector representations, which will be then used for training and testing by the SVM and RF models. In the following lines, we will explain both classifiers and point out their strengths and weaknesses for the task at hand.

2.1.1. Random Forests

RF is an ensemble classification and regression technique introduced by Breiman et al. [25] that has proved to be a highly accurate prediction and classification technique. The ensemble is designed to train more than one classifier, and then aggregate the predictions of all models and perform predictions by majority voting. A good ensemble needs models to be diverse enough and independent from each other to ensure good performance. Broadly speaking, diversifying the ensemble can either include training more than one type of ML algorithm (e.g. SVM, LR, etc...) or alternatively, training one machine learning algorithm on various and diverse subsets of the training set. RF generates a diversified ensemble using Bootstrap aggregating (bagging). Bagging is a sampling method that samples data from the training set with replacement. With such an approach, an instance in the dataset can be sampled more than for the same model. At the same time, other instances may not appear at all during the training process. It is estimated that following this approach, more than 63% of unique instances from the training set will be used during the training process, while almost 37% of the instances will not be sampled at all, and will be used to estimate the "out-of-bag" error. In addition, and to ensure a more diversified ensemble RF at each node split, only a subset of features is drawn randomly to assess the quality of each feature. According to the winning solutions in *kaggle.com*, the state-of-the-art ensemble methods are RF [25] and Gradient Boosting trees [26]. In one of the largest experiments ever carried out in literature [24], where more than 179 classifiers were used on 121 different datasets from the UCI repository [27], RF came first, followed by SVM with Gaussian Kernels.

2.1.2. Support Vector Machine

SVM [28] is another supervised ML algorithm that boosts classification accuracy by projecting the data points to a higher dimensional space, finding an optimal hyperplane that separates positive and negative classes. It has also proven its superiority over other classification methods in terms of speed, simplicity, and precision–recall balance. In [27] and when compared to other widely adopted learning algorithms, SVM with Gaussian kernel ranked second after RF without statistically significant difference. A recent SR of the literature [29] shows that SVM is considered one of the most common approaches in handling class imbalanced datasets.

2.2. Zero-shot classification

In this paper, zero-shot text classification [23] refers to classifying text into a category without any training. The benefit of using this approach is that labels are not needed and with a single model, we can categorise text as belonging to different categories. Moreover, it allows us to start the classification of abstracts without explicitly having to split part of the available data for training, which normally can take some relevant and informative data out of the testing section.

The proposed method is based on pre-trained natural language inference (NLI) models. A text is evaluated as whether it belongs to a specific category or topic a hypothesis is constructed and a probability score is generated. One of the most popular models for zero-shot text classification is the Hugging Face's *facebook/bart-large-mnli* model, based on the BART-large transformer architecture which has over 400 million parameters [23]. A transformer architecture is based on a DL model using an encoder–decoder through a so-called attention mechanism [30]. Fig. 1 illustrates the transformer model architecture. Attention refers to assigning more importance to certain features and less important to others. The attention mechanism is used in the transformer architecture to make features context-aware i.e. word embeddings that capture the semantic relationships between different words taking into account also the position of the sentence. Transformers were initially designed to solve the problem of machine translation [31] or tasks that transform an input sequence into an output sequence. Transformers have become widely used for NLP problems, and are more popular than the traditional sequence type models such as Recurrent Neural networks (RNN) and long short-term memory (LSTM) [32]. The architecture of transformer-based models is shown in Fig. 1 and is based on the groundbreaking work outlining the initial proposal of transformer models [30]. The first examples of pre-trained transformer-based NLP systems are Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformers (GPT), which were trained with large language datasets, such as the Wikipedia Corpus and Common Crawl, and can be fine-tuned for more specific tasks such as image generation.

In this paper, we use the BART transformer encoder–decoder (sequence-to-sequence) model [33] with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder. It is pre-trained for the English language, which has proven to work well for text-related tasks such as summarisation, translation and comprehension including text classification [33]. Moreover, we use $transformers$, $pipeline$ and $classifier = pipeline(\varepsilon zeroshotclassification\varepsilon, model = \varepsilon facebook/bart\text{-}large - mnli\varepsilon)$ parameters. The abstracts of the papers are passed to this pipeline to classify into any of the specified categories or a primary label. Notice the importance of how the primary label (category) is defined, as it affects the inclusion or exclusion of the abstract. Using this approach, we apply the zero-shot classification to the set of abstracts for each topic and evaluate the performance using the true class (include/exclude) by calculating the weighted precision, recall and F1 score. The model pipeline produces a score which is the probability of the abstract belonging to the specified primary label. By setting different thresholds for the probability, we evaluate the accuracy of the prediction to assign an abstract to include or exclude a class. If the probability score is set to a higher threshold then more abstracts will be excluded. A high-level diagram of the zero-shot classification is presented in Fig. 2.

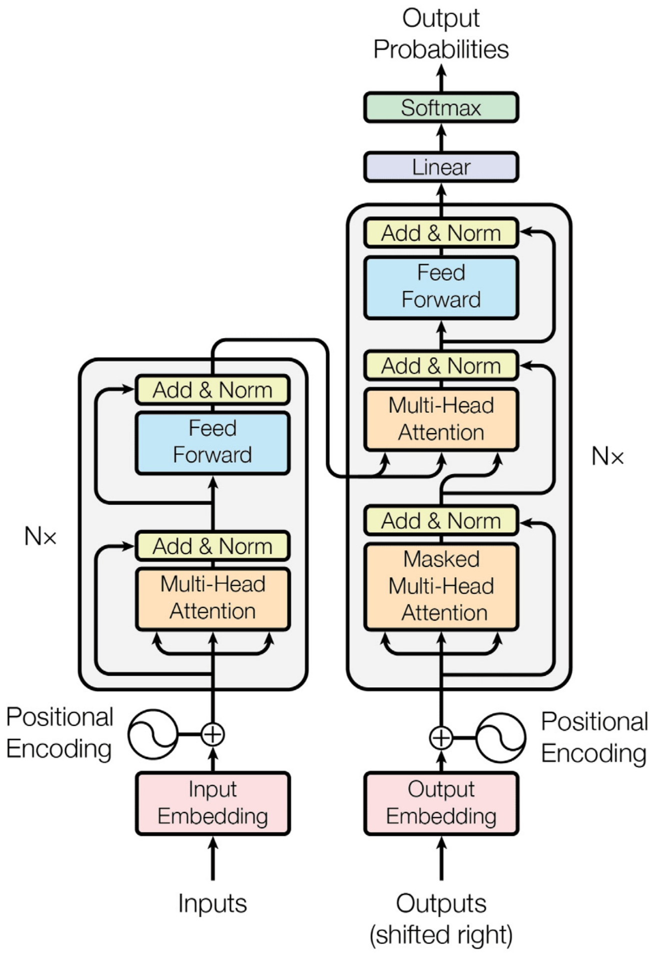

Fig. 1. Transformer model architecture [30].

2.3. Hybrid approach

For the hybrid approach, we combine the zero-shot classification method explained in 2.2 and the traditional machine learning algorithms RF and SVM. The main reason we used the hybrid method is to improve the accuracy of the zero-shot classification but, at the same time, greatly reduce the number of abstracts for manual screening. This can help simulate how a systematic reviewer would use this method to reduce workload. That is, first selecting a subset of abstracts for inclusion using the zero-shot classification to annotate them. Afterwards, this smaller subset of abstracts can be manually screened and used to train a supervised learning algorithm to be used for the entire dataset.

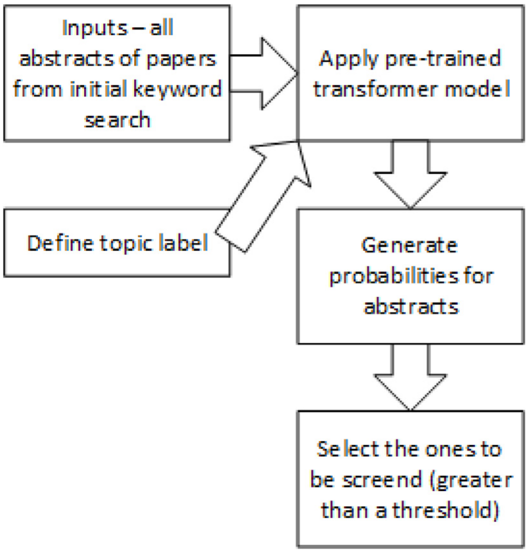
Inputs – all abstracts of papers from initial keyword search
Apply pre-trained transformer model
Define topic label
Generate probabilities for abstracts
Select the ones to be screend (greater than a threshold)

Fig. 2. High-level diagram using zero-shot classification.

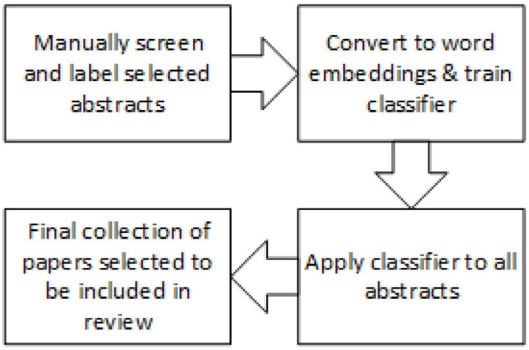
Manually screen and label selected abstracts
Convert to word embeddings & train classifier
Final collection of papers selected to be included in review
Apply classifier to all abstracts

Fig. 3. High-level diagram using the hybrid approach.

To do so, we apply the zero-shot classification to the set of all abstracts for a particular dataset. All that have scored more than a specific threshold τ are identified as abstracts to be included, i.e. assigned label 1. If we set the threshold, τ, of the probability too high, then only very few abstracts will be labelled to be included. If it is set too low, the manual labelling required will be higher. For the experiments in this paper, we set the threshold to 0.5 to find out whether we can improve on the zero-shot classification with some additional manual labelling. We then assume that a manual process is applied to these abstracts identified as included by the zero-shot classification. After that manual labelling of this part of the dataset, we use it as a training set and apply the traditional ML approach, i.e. convert the text of the abstracts into vector representations and then train SVM and RF models. The trained models are evaluated on the remaining dataset of abstracts identified initially as excluded from the zero-shot classification by calculating the weighted precision, recall and F1 score. It is important to note that this approach performs the manual labelling abstract screening on a smaller dataset than when we apply the traditional ML approach from the beginning. The hybrid approach can be considered a fine-tuning of the zero-shot classification. A high-level diagram of the hybrid approach is presented in Fig. 3.

3. Data preparation

3.1. Text embeddings

To apply ML to classify the abstracts, these have to be represented as numeric data. One way to convert text into vector representation with numbers is to use one-hot encoding, i.e. associate a unique integer number with every word and turn the integer index into a binary vector. This results in encoding a text with very high dimensional vectors (i.e., the vocabulary size). Another way is to use text embeddings, i.e. encoding words or phrases from a language vocabulary to vectors of real numbers. Text embeddings encode very large vocabularies in low-dimensional vectors learnt from data.

In this paper, we use widely established models for converting text data into numerical representations based on Word2Vec algorithms introduced by Mikolov et al. in [34,35]. We relied on the Python-based implementation in the *Gensim* library [36] and embeddings based on Global Vectors for Word Representation (GloVe), FastText and Doc2Vec models. These three word embedding models are briefly explained in the following lines.

3.1.1. GloVe

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. It is based on a global log bi-linear regression model that combines global matrix factorisation and local context window methods [37]. The GloVe model is trained on aggregated global word–word co-occurrence matrix from a corpus which captures the frequency of words that co-occur with one another in a given corpus. GloVe6.b provides pre-trained word vectorisations with 100, 200, 300 dimensions trained over large corpora, including Wikipedia 2014, Gigaword 5 and Twitter content [37]. In this particular work, we use a word vectorisation with dimension 300. Using the *gensim.scripts.glove2word2vec* method, we convert GloVe vectors into the Word2Vec ones.

3.1.2. FastText

FastText is an open-source library for efficient learning of word representations and sentence classification developed by Facebook [38]. FastText enables representing sentences with bag of words and bag of n-grams and enriching word vectors with sub-word information. It can be used to learn word vector embeddings with the additional ability to obtain word vectors for out-of-vocabulary words.

3.1.3. Doc2Vec

Doc2Vec is a model that represents each document as a vector based on Word2Vec. Doc2Vec is based on the paper Distributed Representations of Sentences and Documents Mikolov et al. [35]. The idea of the *Gensim Doc2Vec* model is that a word vector is generated for each word, and a document vector is generated for each document. The model also trains weights for a softmax hidden layer. After training, when a new document is presented, the vector representation is calculated after training.

3.2. Datasets

To validate our inclusion/exclusion classification methodology, we collected a total of five datasets from different SRs performed in diverse health sciences domains. Most of the datasets (except for Aceves-Martins2021, which is available in Doc2Vec format on our GitHub demo [39]) can be obtained from here [40]. A summary of their characteristics is presented in Table 1. We also include the question that the experts were looking for during the abstract screening process, the number of excluded and included studies, and finally, the imbalance ratio (IR). We extracted the features to train and test the classification algorithms for each of these abstract datasets, as described in Section 3.1. For each dataset, we obtained 200 Doc2Vec features, 300 FastText features and 300 GloVe features.

Table 1
Summary of the characteristics of the datasets used for experimentation.

Abv.	Topic	Question asked at abstract level to include/exclude the manuscript	Excluded	Included	IR
Aceves-Martins2021	Oral health	Was the study performed in Mexico (with Mexican children)?	789	18	43.83
Bannach-Brown2016	Animal depression	Does the study provide primary data of an animal model of depression or depressive-like phenotype with an appropriate control group?	1713	280	6.11
Cohen2006A	Atypical antisychotics	Where the patients in the study exposed to (atypical) antisychotics?	757	363	2.08
Cohen2006C	Calcium channel blockers	Where the patients in the study exposed to calcium channel blockers?	939	279	3.36
Cohen2006O	Oral hypoglycemics	Where the patients in the study exposed to (oral) hypoglycemics?	364	139	2.61

3.3. Data balancing

The datasets in the problem of abstract screening are highly imbalanced, i.e. only a small number of instances belong to one of the classes of included abstracts. A common approach to address the imbalanced dataset for classification is using the Synthetic Minority Oversampling Technique (SMOTE) [41]. This method is designed to synthesise new data points by interpolating neighbouring instances. The method has been effective in handling class imbalance and has been used across a wide range of real-world applications [42–44]. In addition, various extensions have been proposed based on the original methods, including DBSMOTE [45], SLSMOTE [46], MWMOTE [47] and others. In this work, however, we opted for a more recent and novel enhancement method called Class Decomposition SMOTE (CDSMOTE) presented in [22]. CDSMOTE is based on two key ideas which are commonly used in these settings; class-decomposition [48,49], and data augmentation via SMOTE [41].

CDSMOTE works by under-sampling the majority class instances using unsupervised learning algorithms (e.g. kmeans) and over-sampling the minority class instances based on some heuristics using SMOTE. The motivation behind this choice is to minimise information loss, which often happens due to other common undersampling methods. This is mainly because under-sampling in CDSMOTE refers to clustering the majority class instances into sub-clusters, which results in less imbalanced datasets and, at the same time, provides more fine-grained training to the learning algorithms. The CDSMOTE method presented in [22] is comprised of two steps: (1) class decomposition to redistribute the number of samples per class without losing any sample and (2) oversampling the new minority class(es) to reduce the dominance of the new majority class(es). Regarding the first step, class decomposition can be broadly described as clustering class instances into smaller groups employing unsupervised learning algorithms. As a result, the dominance of a class can be greatly reduced without losing any information. We use two methods for clustering kmeans and DBSCAN, which we denote with *CDSMOTE-kmeans* and *CDSMOTE-DBSCAN*, respectively. A detailed description of the algorithm can be found in [22,50].

4. Experiment

We evaluated the performance of the following six settings using the true class (include/exclude) for the abstracts by calculating the weighted precision, recall and F1-score:

1. SVM classification on the datasets with 5-fold cross-validation. The default Python SVM scikit-learn package parameters [51] were used for the classifier.
2. RF classification on the datasets with 5-fold cross-validation. The default Python RF scikit-learn package parameters [51] were used for the classifier.
3. SVM classification trained with the labels generated by the CDSMOTE method (as described in Section 3.3, tested using the original labels) with 5-fold cross-validation. Default SVM parameters were used, and for the CDSMOTE algorithm, we present two alternatives: a) kmeans for the decomposition step with $k = 2$ and b) DBSCAN for the decomposition step with $eps = 0.1$
4. RF classification trained with the labels generated by the CDSMOTE method (as described in Section 3.3, tested using the original labels) with 5-fold cross-validation. Default RF parameters were used, and for the CDSMOTE algorithm, we present two alternatives: a) kmeans for the decomposition step with $k = 2$ and b) DBSCAN for the decomposition step with $eps = 0.1$
5. Zero-shot classification with a primary label defined for each dataset and as described in Section 2.2. In this case, there is no need for a train/test split as the entire dataset is used to validate the classification. Three different thresholds τ for the probabilities to assign an abstract as belonging to the defined primary label were tested for each dataset (only the best three are presented for each SR dataset). The primary labels are defined as follows for the datasets used in the experiments:
   - Aceves-Martins2021 - [children obesity Mexico]
   - Bannach-Brown2016 - [animal depression]
   - Cohen2006 A - [patients Atypical Antipsychotic]
   - Cohen2006C - [calcium channel blockers clinical evidence]
   - Cohen2006O - [exposed to oral hypoglycaemics]
6. zero-shot classification with a primary label followed by SVM and RF to classify the result (i.e. the Hybrid approach described in Section 2.3). Again, default package parameters [51] were used for the SVM and RF classifiers, and the threshold used was $\tau = 0.5$ for the probabilities. To tackle imbalance in the dataset the standard class, *imblearn.over_sampling. RandomOverSampler* was used before applying SVM and RF.

A guided sample of the code used in these experiments can be found here [39].

5. Results and discussion

Table 2 shows the results of implementing methods 1 and 2 as described in Section 4. Notice that the best results achieved within the three types of extracted features are marked in bold. We notice here that precision is typically better using SVM, while the recall is mostly tied between the two classifiers. As a result, the F1 measure is best for SVM. Moreover, we notice that, in general, the Doc2Vec extractor yields the best performance overall. This initial study is useful for us to understand what to expect as the more sophisticated methodologies are applied.

Table 3 shows the results of classifying the SR datasets with the CDSMOTE variants of the labels for the training data. This time, the

Table 2
Results of implementing methods 1 and 2 (classification of original datasets). The best-aggregated results achieved for each dataset and within each classifier are highlighted in bold.

Abv.	Embedding	SVM			RF		
		Prec	Rec	F1	Prec	Rec	F1
Aceves-Martins2021	GloVe	0.959	0.979	0.969	0.959	0.977	0.968
	FastText	0.959	0.979	0.969	0.959	0.979	0.969
	Doc2Vec	**0.964**	**0.957**	**0.96**	**0.959**	**0.979**	**0.969**
Bannach-Brown2016	GloVe	0.739	0.859	0.795	0.738	0.859	0.794
	FastText	0.739	0.859	0.794	0.739	0.859	0.794
	Doc2Vec	**0.875**	**0.877**	**0.876**	**0.739**	**0.859**	**0.795**
Cohen2006A	GloVe	0.456	0.676	0.545	0.456	0.675	0.545
	FastText	0.456	0.676	0.544	0.456	0.676	0.545
	Doc2Vec	**0.675**	**0.683**	**0.678**	**0.522**	**0.676**	**0.546**
Cohen2006C	GloVe	0.594	0.77	0.671	0.594	0.77	0.671
	FastText	0.594	0.77	0.671	0.594	0.77	0.671
	Doc2Vec	**0.708**	**0.713**	**0.711**	**0.594**	**0.77**	**0.671**
Cohen2006O	GloVe	0.521	0.722	0.605	**0.596**	**0.722**	**0.61**
	FastText	0.521	0.722	0.605	0.55	0.722	0.612
	Doc2Vec	**0.699**	**0.677**	**0.686**	0.521	0.722	0.605

Table 3
Results of implementing methods 3 and 4 (classification of balanced datasets using CDSMOTE). The best-aggregated results achieved for each dataset and within each classifier are highlighted in bold.

Abv.	Embedding, CDSMOTE Parameters	SVM			RF		
		Prec	Rec	F1	Prec	Rec	F1
Aceves-Martins2021	GloVe, kmeans = 2	0.877	0.834	0.851	0.927	0.921	0.921
	GloVe, DBSCAN = 0.1	0.871	0.827	0.852	0.936	0.928	0.928
	FastText, kmeans = 2	0.825	0.708	0.762	0.929	0.926	0.926
	FastText, DBSCAN = 0.1	0.76	0.414	0.539	0.946	0.94	0.94
	Doc2Vec, kmeans = 2	0.978	0.977	0.977	0.95	0.95	0.95
	Doc2Vec, DBSCAN = 0.1	**0.983**	**0.982**	**0.982**	**0.964**	**0.962**	**0.962**
Bannach-Brown2016	GloVe, kmeans = 2	0.739	0.739	0.738	0.733	0.731	0.726
	GloVe, DBSCAN = 0.1	0.725	0.724	0.723	0.747	0.745	0.745
	FastText, kmeans = 2	0.312	0.559	0.401	0.753	0.754	0.752
	FastText, DBSCAN = 0.1	0.601	0.497	0.554	0.754	0.753	0.752
	Doc2Vec, kmeans = 2	**0.872**	**0.871**	**0.871**	0.817	0.808	0.803
	Doc2Vec, DBSCAN = 0.1	0.599	0.496	0.553	**0.817**	**0.815**	**0.815**
Cohen2006A	GloVe, kmeans = 2	0.586	0.582	0.561	0.642	0.641	0.637
	GloVe, DBSCAN = 0.1	0.611	0.609	0.608	0.649	0.647	0.647
	FastText, kmeans = 2	0.281	0.53	0.367	0.64	0.634	0.622
	FastText, DBSCAN = 0.1	0.499	0.499	0.401	0.656	0.655	0.655
	Doc2Vec, kmeans = 2	0.713	0.711	0.711	0.687	0.653	0.626
	Doc2Vec, DBSCAN = 0.1	**0.713**	**0.712**	**0.712**	**0.683**	**0.68**	**0.679**
Cohen2006C	GloVe, kmeans = 2	0.619	0.612	0.594	0.642	0.626	0.603
	GloVe, DBSCAN = 0.1	0.624	0.623	0.622	0.668	0.666	0.665
	FastText, kmeans = 2	0.284	0.533	0.371	0.646	0.631	0.609
	FastText, DBSCAN = 0.1	0.535	0.526	0.457	0.667	0.665	0.664
	Doc2Vec, kmeans = 2	**0.741**	**0.74**	**0.739**	0.727	0.665	0.63
	Doc2Vec, DBSCAN = 0.1	0.731	0.73	0.73	**0.728**	**0.726**	**0.725**
Cohen2006O	GloVe, kmeans = 2	0.657	0.641	0.625	0.673	0.671	0.671
	GloVe, DBSCAN = 0.1	0.638	0.603	0.575	0.687	0.68	0.678
	FastText, kmeans = 2	0.277	0.527	0.363	0.664	0.663	0.663
	FastText, DBSCAN = 0.1	0.661	0.537	0.428	0.72	0.715	0.714
	Doc2Vec, kmeans = 2	0.78	0.769	0.768	**0.754**	**0.754**	**0.754**
	Doc2Vec, DBSCAN = 0.1	**0.789**	**0.786**	**0.786**	0.746	0.74	0.739

numbers marked in bold indicate the results that were better than the experiments on the original versions (i.e. methods 1 and 2 shown in Table 2). Notice that we can improve our results for all datasets except for Bannach-Brown2016, where the difference was just 0.002 for all metrics used, and the recall of Cohen2006C, with a difference of 0.3 with regards to the original dataset. Moreover, we find that SVM with Doc2Vec features is the best available combination for improved performance. Finally, we notice that for all datasets (except for Bannach-Brown2016 and Cohen2006C for a slight margin), the use of DBSCAN for the decomposition step was better than kmeans. This indicates that these particular datasets may be more challenging for the clustering algorithms and thus, the lack of improvement with respect to the original versions.

Table 4 shows the results of applying methods 5 and 6, which are based on the zero-shot classification method presented in this paper. The results show that the hybrid approach outperforms the previous approaches (i.e. methods 1–4 shown in Tables 2 and 3) in most cases. The hybrid approach shows the best possible results for all the datasets considered, except for the precision of Cohen2006C and the precision and F1-measure of Cohen2006O. However, these differences are smaller by a minimal margin with respect to their CDSMOTE counterparts. Once again, the results suggest that the Doc2Vec and SVM combination yields the best results overall, except for Aceves-Martins2021 and Bannach-Brown2016, where the best results are obtained using RF as the classifier, given by the larger size and amount of information contained in the abstracts to be handled. Nonetheless, we have to consider that for these methods, there is no need to split the dataset

Table 4
Results of implementing methods 5 and 6 (zero-shot based). The best-aggregated results achieved for each dataset and within each method are highlighted in bold.

Abv.						Hybrid Method			
	zero-shot classification					Threshold = 0.5			
	Embedding	Threshold	Prec	Rec	F1	Embedding, Classifier	Prec	Rec	F1
Aceves-Martins2021	Glove	0.2	0.978	0.726	0.822	GloVe, SVM	0.99	0.74	0.84
						GloVe, RF	0.99	0.96	0.97
	FastText	0.5	0.976	0.939	0.954	FastText, SVM	0.99	0.89	0.94
						FastText, RF	0.99	0.95	0.97
	Doc2Vec	0.6	**0.981**	**0.971**	**0.975**	Doc2Vec, SVM	0.99	0.93	0.96
						Doc2Vec, RF	**0.99**	**0.99**	**0.99**
Bannach-Brown2016	GloVe	0.9	0.828	0.771	0.793	GloVe, SVM	0.89	0.77	0.82
						GloVe, RF	0.9	0.9	0.9
	FastText	0.8	0.832	0.708	0.749	FastText, SVM	0.89	0.76	0.81
						FastText, RF	0.89	0.9	0.9
	Doc2Vec	0.5	**0.835**	**0.612**	**0.671**	Doc2Vec, SVM	0.92	0.91	0.92
						Doc2Vec, RF	**0.92**	**0.94**	**0.92**
Cohen2006A	GloVe	0.02	0.698	0.375	0.271	GloVe, SVM	0.1	0.32	0.15
						GloVe, RF	0.62	0.64	0.63
	FastText	0.1	**0.597**	**0.472**	**0.475**	FastText, SVM	0.1	0.32	0.15
						FastText, RF	0.63	0.65	0.64
	Doc2Vec	0.2	0.567	0.501	0.517	Doc2Vec, SVM	**0.73**	**0.74**	**0.73**
						Doc2Vec, RF	0.65	0.69	0.61
Cohen2006C	GloVe	0.2	0.667	0.573	0.605	GloVe, SVM	0.73	0.64	0.67
						GloVe, RF	0.72	0.77	0.73
	FastText	0.5	0.675	0.663	0.669	FastText, SVM	0.72	0.64	0.66
						FastText, RF	0.73	0.78	0.74
	Doc2Vec	0.6	**0.669**	**0.678**	**0.674**	Doc2Vec, SVM	**0.74**	**0.75**	**0.75**
						Doc2Vec, RF	0.68	0.78	0.7
Cohen2006O	GloVe	0.3	0.678	0.431	0.422	GloVe, SVM	0.73	0.75	0.74
						GloVe, RF	0.75	0.78	0.76
	FastText	0.4	0.648	0.449	0.456	FastText, SVM	0.72	0.71	0.71
						FastText, RF	0.69	0.64	0.66
	Doc2Vec	0.5	**0.649**	**0.508**	**0.531**	Doc2Vec, SVM	**0.78**	**0.8**	**0.78**
						Doc2Vec, RF	0.72	0.78	0.71

Table 5
Summary of the best results obtained for each dataset.

Abv.	SVM			CDSMOTE SVM			Hybrid Method		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Aceves-Martins2021	0.959	0.979	0.969	0.983	0.982	0.982	**0.99**	**0.99**	**0.99**
Bannach-Brown2016	0.875	0.877	0.876	0.872	0.871	0.871	**0.92**	**0.94**	**0.92**
Cohen2006A	0.675	0.683	0.678	0.713	0.712	0.712	**0.73**	**0.74**	**0.76**
Cohen2006C	0.708	0.713	0.711	0.741	0.74	0.739	**0.74**	**0.75**	**0.75**
Cohen2006O	0.699	0.677	0.686	0.789	0.786	0.786	**0.78**	**0.8**	**0.78**

in training and testing; therefore, all samples are being used, which means that there is more data in the testing set. Further hyperparameter optimisation may improve these results, which will be considered in our future work. Most important is to note that with zero-shot classification, manual labelling is eliminated or with the hybrid approach reduced significantly. Applying the hybrid approach requires some manual labelling. The proportions of abstracts for each dataset that were manually labelled, i.e. used for training of the SVM and RF models, are Aceves-Martins2021 (7%), Bannach-Brown2016 (46%), Cohen2006 A (44%), Cohen2006C (25%) and Cohen2006O (58%). These proportions depend on the threshold τ chosen for the probability of which abstracts will be included after performing the zero-shot classification. The larger the threshold, the smaller proportions of the abstracts will be marked with label 1 (i.e. to be included). The threshold τ in our experiment was set to 0.5. In the traditional machine learning approach, the proportion of the datasets used for training was 70%, i.e., the proportion that requires manual labelling. The final summary of the results can be seen in Table 5.

6. Conclusion

In this paper, we applied traditional ML methods and zero-shot classification methods based on transformer DL architecture for automating abstract screening of the SR process. We evaluated the performance of these methods by conducting experiments using five datasets from different SRs in diverse health science domains. We evaluated three algorithms, GloVe, FastText and Doc2Vec, for converting the text of the abstracts to vector representations. Tables 2−4 show that Doc2Vec text embedding provides the best results, which is explained by the fact that these vector representations can capture the semantics, i.e. the meaning, of the input texts. Table 5 shows that the best results are obtained using the hybrid method, which combines traditional ML and zero-shot classification. The advantage of the hybrid method is that it allows the use of unlabelled data in the initial step of scanning abstracts, thus saving significant time and effort. Moreover, we can notice that for the most recent datasets, namely Aceves-Martins2021 and Bannach-Brown2016, we obtained results over 90% in all metrics (compared to 73% to 80% in the Cohen datasets), since these datasets were compiled most recently, with larger information and from domains where authors tend to write more information in a more systemic and organised manner. In our future work, we will explore further levels of embeddings (character, word, sentence, document) and attention mechanisms to improve further the results and reduce further the need for manual labelling. Beyond aiding in abstract screening for the SR process, the methods presented in this paper could also be used for classification of text in more general tasks, such as acceptance/rejection of manuscripts in terms of appropriateness to a given journal or conference, prescreening of legal documents related to a certain court case, amongst others.

CRediT authorship contribution statement

Carlos Francisco Moreno-Garcia: Run experiments, Wrote the paper, Developed the dataset. **Chrisina Jayne:** Run experiments, Wrote the paper, Developed the dataset. **Eyad Elyan:** Wrote and proofread the paper. **Magaly Aceves-Martins:** Provided data, Wrote and proofread the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data and code are available in one of the references of the manuscript [39], which points to a GitHub site.

References

[1] A. Legate, K. Nimon, (Semi)automated approaches to data extraction for systematic reviews and meta-analyses in social sciences: A living review protocol [version 1; peer review: 3 approved with reservations] 11:1036, 2022, http://dx.doi.org/10.12688/f1000research.125198.1.

[2] G.M. Tawfik, K.A.S. Dila, M.Y.F. Mohamed, D.N.H. Tam, N.D. Kien, A.M. Ahmed, N.T. Huy, A step by step guide for conducting a systematic review and meta-analysis with simulation data, 47 (1) (2019) http://dx.doi.org/10.1186/s41182-019-0165-6.

[3] M. Aceves-Martins, L. López-Cruz, M. García-Botello, Y.Y. Gutierrez Gómez, C.F. Moreno-García, Interventions to Treat Obesity in Mexican Children and Adolescents: Systematic Review and Meta-Analysis, Nutr. Rev. (2021) http://dx.doi.org/10.1093/nutrit/nuab041, nuab041.

[4] A.M. Cohen, W.R. Hersh, K. Peterson, P.-Y. Yen, Reducing workload in systematic review preparation using automated citation classification, 13 (2), 2006, pp. 206–219, http://dx.doi.org/10.1197/jamia.m1929.

[5] D. Mowery, B.R. South, M. Kvist, H. Dalianis, S. Velupillai, Recent advances in clinical natural language processing in support of semantic analysis, 24 (01), 2015, pp. 183–193, http://dx.doi.org/10.15265/iy-2015-009.

[6] C.F. Moreno-García, M. Aceves-Martins, F. Serratosa, Unsupervised Machine Learning Application to Perform a Systematic Review and Meta-Analysis in Medical Research, Comput. Y Sistemas 20 (1) (2016) 7–17, http://dx.doi.org/10.13053/CyS-20-1-2360.

[7] A. O'Mara-Eves, J. Thomas, J. McNaught, M. Miwa, S. Ananiadou, Using text mining for study identification in systematic reviews: a systematic review of current approaches, Syst. Rev. 4 (1) (2015) http://dx.doi.org/10.1186/2046-4053-4-5.

[8] A. Blaizot, S.K. Veettil, P. Saidoung, C.F. Moreno-García, N. Wiratunga, M. Aceves-Martins, N.M. Lai, N. Chaiyakunapruk, Using artificial intelligence methods for systematic review in health sciences: A systematic review, Res. Synthesis Methods (2022) http://dx.doi.org/10.1002/jrsm.1553.

[9] M.M. Kebede, C. Le Cornet, R. Turzanski Fortner, In-depth evaluation of machine learning methods for semi-automating article screening in a systematic review of mechanistic literature automated screening, citation screening, machine learning, natural language processing, NLP, systematic review, text min, Res. Syn. Meth. (2022) http://dx.doi.org/10.1002/jrsm.1589, URL https://onlinelibrary.wiley.com/doi/10.1002/jrsm.1589.

[10] H. Khalil, D. Ameen, A. Zarnegar, Tools to support the automation of systematic reviews: A scoping review, J. Clin. Epidemiol. 144 (2022) 22–42, http://dx.doi.org/10.1016/j.jclinepi.2021.12.005.

[11] D. Denyer, D. Tranfield, Producing a systematic review, 2009.

[12] M.F. Delgado, E. Cernadas, S. Barro, D.G. Amorim, Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 15 (2014) 3133–3181.

[13] A. Gates, C. Johnson, L. Hartling, Technology-assisted title and abstract screening for systematic reviews: a retrospective evaluation of the Abstrackr machine learning tool, System. Rev. 7 (1) (2018) http://dx.doi.org/10.1186/s13643-018-0707-8.

[14] G. Cleo, A.M. Scott, F. Islam, B. Julien, E. Beller, Usability and acceptability of four systematic review automation software packages: a mixed method design, Syst. Rev. 8 (1) (2019) http://dx.doi.org/10.1186/s13643-019-1069-6.

[15] W. Yu, M. Clyne, S.M. Dolan, A. Yesupriya, A. Wulf, T. Liu, M.J. Khoury, M. Gwinn, GAPscreener: An automatic tool for screening human genetic association literature in PubMed using the support vector machine technique, 9 (1), 2008, http://dx.doi.org/10.1186/1471-2105-9-205.

[16] I.J. Marshall, A. Noel-Storr, J. l Kuiper, J. Thomas, B.C. Wallace, Machine learning for identifying randomized controlled trials: An evaluation and practitioner's guide, Res. Synth. Methods 9 (4) (2018) 602–614, http://dx.doi.org/10.1002/jrsm.1287.

[17] P. Przybyła, A.J. Brockmeier, G. Kontonatsios, M.-A. Le Pogam, J. McNaught, E. von Elm, K. Nolan, S. Ananiadou, Prioritising references for systematic reviews with RobotAnalyst: A user study, Res. Synth. Methods 9 (3) (2018) 470–488, http://dx.doi.org/10.1002/jrsm.1311, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jrsm.1311.

[18] Z. Xiong, T. Liu, G. Tse, M. Gong, P.A. Gladding, B.H. Smaill, M.K. Stiles, A.M. Gillis, J. Zhao, A machine learning aided systematic review and meta-analysis of the relative risk of atrial fibrillation in patients with diabetes mellitus, Front. Physiol. 9 (2018) 835, http://dx.doi.org/10.3389/fphys.2018.00835, URL https://www.frontiersin.org/article/10.3389/fphys.2018.00835.

[19] M. Karasalo, J. Schubert, Developing horizon scanning methods for the discovery of scientific trends, in: 2019 International Conference on Document Analysis and Recognition, ICDAR, 2019, pp. 1055–1062, http://dx.doi.org/10.1109/ICDAR.2019.00172.

[20] R. Pradhan, D.C. Hoaglin, M. Cornell, W. Liu, V. Wang, H. Yu, Automatic extraction of quantitative data from ClinicalTrials.gov to conduct meta-analyses, J. Clin. Epidemiol. 105 (2019) 92–100, http://dx.doi.org/10.1016/j.jclinepi.2018.08.023, URL https://www.sciencedirect.com/science/article/pii/S0895435617313069.

[21] Y. He, C. Wang, S. Zhang, N. Li, Z. Li, Z. Zeng, KG-MTT-BERT: Knowledge graph enhanced BERT for multi-type medical text classification, 2022, http://dx.doi.org/10.48550/ARXIV.2210.03970, URL https://arxiv.org/abs/2210.03970.

[22] E. Elyan, C.F. Moreno-García, C. Jayne, CDSMOTE: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification, Neural Comput. Appl. (2020) http://dx.doi.org/10.1007/s00521-020-05130-z.

[23] W. Yin, J. Hay, D. Roth, Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach, 2019, arXiv:1909.00161.

[24] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? J. Mach. Learn. Res. 15 (2014) 3133–3181, http://dx.doi.org/10.1016/j.csda.2008.10.033.

[25] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.

[26] J.H. Friedman, Stochastic gradient boosting, Comput. Statist. Data Anal. 38 (4) (2002) 367–378.

[27] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 15 (2014) 3133–3181.

[28] Y. Zhang, Support vector machine classification algorithm and its application, in: Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings, Part II, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 179–186, http://dx.doi.org/10.1007/978-3-642-34041-3_27.

[29] P. Vuttipittayamongkol, E. Elyan, A. Petrovski, On the class overlap problem in imbalanced data classification, Knowl.-Based Syst. 212 (2021) 106631, http://dx.doi.org/10.1016/j.knosys.2020.106631, URL http://www.sciencedirect.com/science/article/pii/S0950705120307607.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[31] A. Graves, Sequence transduction with recurrent neural networks, 2012, CoRR abs/1211.3711 arXiv:1211.3711.

[32] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45, http://dx.doi.org/10.18653/v1/2020.emnlp-demos.6, URL https://aclanthology.org/2020.emnlp-demos.6.

[33] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019, arXiv:1910.13461.

[34] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv:1301.3781.

[35] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, 2013, arXiv:1310.4546.

[36] R. Ř.ehůřek, P. Sojka, Software Framework for Topic Modelling with Large Corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, ELRA, Valletta, Malta, 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[37] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543, http://dx.doi.org/10.3115/v1/D14-1162, URL https://www.aclweb.org/anthology/D14-1162.

[38] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, 2016, http://dx.doi.org/10.48550/ARXIV.1607.04606, URL https://arxiv.org/abs/1607.04606.

[39] C.F. Moreno-Garcia, Zero shot abstract classification, 2022, https://github.com/carlosfmorenog/Zero-Shot_Abstract_Classification. (Accessed 30 June 2022).

[40] ASReview, Asreview systematic review datasets, 2022, https://github.com/asreview/systematic-review-datasets. (Accessed 30 June 2022).

[41] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.

[42] J. Sun, J. Lang, H. Fujita, H. Li, Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates, Inform. Sci. 425 (2018) 76–91, http://dx.doi.org/10.1016/j.ins.2017.10.017, URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85042332829&doi=10.1016%2fj.ins.2017.10.017&partnerID=40&md5=6cd752a20a6505030c067df5d29a4d9f, cited By 103.

[43] M. Ijaz, G. Alfian, M. Syafrudin, J. Rhee, Hybrid prediction model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over sampling technique (SMOTE), and random forest, Appl. Sci. 8 (8) (2018) 1325.

[44] S. Wang, D. Wang, J. Li, T. Huang, Y.-D. Cai, Identification and analysis of the cleavage site in a signal peptide using SMOTE, dagging, and feature selection methods, Molecular Omics 14 (1) (2018) 64–73.

[45] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, DBSMOTE: density-based synthetic minority over-sampling technique, Appl. Intell. 36 (3) (2012) 664–684.

[46] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2009, pp. 475–482.

[47] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, D. Huang, NI-MWMOTE: An improving noise-immunity majority weighted minority oversampling technique for imbalanced classification problems, Expert Syst. Appl. (2020) 113504.

[48] E. Elyan, M.M. Gaber, A genetic algorithm approach to optimising random forests applied to class engineered data, Inform. Sci. 384 (Supplement C) (2017) 220–234, http://dx.doi.org/10.1016/j.ins.2016.08.007.

[49] E. Elyan, M.M. Gaber, A fine-grained random forests using class decomposition: an application to medical diagnosis, Neural Comput. Appl. 27 (8) (2016) 2279–2288, http://dx.doi.org/10.1007/s00521-015-2064-z.

[50] C.F. Moreno-García, C. Jayne, E. Elyan, Class-Decomposition and Augmentation for Imbalanced Data Sentiment Analysis, in: International Joint Conference on Neural Networks, IJCNN, IEEE, 2021, pp. 1–7.

[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.