Journal of Intelligent Information Systems (2023) 60:571–591
https://doi.org/10.1007/s10844-022-00768-8

Multi-class classification of COVID-19 documents using machine learning algorithms

Gollam Rabby[1] · **Petr Berka**[1]

Received: 10 June 2022 / Revised: 16 November 2022 / Accepted: 17 November 2022 /
Published online: 29 November 2022

Abstract

In most biomedical research paper corpus, document classification is a crucial task. Even due to the global epidemic, it is a crucial task for researchers across a variety of fields to figure out the relevant scientific research papers accurately and quickly from a flood of biomedical research papers. It can also assist learners or researchers in assigning a research paper to an appropriate category and also help to find the relevant research paper within a very short time. A biomedical document classifier needs to be designed differently to go beyond a "general" text classifier because it's not dependent only on the text itself (i.e. on titles and abstracts) but can also utilize other information like entities extracted using some medical taxonomies or bibliometric data. The main objective of this research was to find out the type of information or features and representation method creates influence the biomedical document classification task. For this reason, we run several experiments on conventional text classification methods with different kinds of features extracted from the titles, abstracts, and bibliometric data. These procedures include data cleaning, feature engineering, and multi-class classification. Eleven different variants of input data tables were created and analyzed using ten machine learning algorithms. We also evaluate the data efficiency and interpretability of these models as essential features of any biomedical research paper classification system for handling specifically the COVID-19 related health crisis. Our major findings are that TF-IDF representations outperform the entity extraction methods and the abstract itself provides sufficient information for correct classification. Out of the used machine learning algorithms, the best performance over various forms of document representation was achieved by Random Forest and Neural Network (BERT). Our results lead to a concrete guideline for practitioners on biomedical document classification.

Keywords Multi-class classification · Machine learning algorithms · Text mining · COVID-19

✉ Gollam Rabby
rabg00@vse.cz

Petr Berka
berka@vse.cz

[1] Department of Information and Knowledge Engineering, Prague University of Economics and Business, Prague, Czech Republic

1 Introduction

Text mining or text analytics can be understood as data mining on textual documents. So, text mining aims at discovering novel, interesting and useful patterns and new insights in large collections of texts. Typical text mining tasks are text categorization (i.e., classification of documents into different classes), document clustering (i.e., grouping documents according to their similarity), or document filtering (e.g., classification of documents into two classes like interesting vs. uninteresting documents, or spam vs. ham). The main problem in text mining is the question of the suitable representation of unstructured texts to be analyzed using machine learning (ML) algorithms. Most of the algorithms can process only structured data organized into a single data table containing a fixed number of columns – features representing the instances (documents in the case of text mining) that are stored as rows in the table. A typical pipeline of text processing that looks for suitable features consists of lexical analysis (to identify individual words), lemmatization or stemming (to transform inflected words to their base form), and stop-words removal (to remove words not related to the content of the document). This process results in obtaining tokens (terms, words) that are used as features in the data table. Each document is then represented as a vector of a fixed length that contains information about the tokens that occur in the document for the bag-of-word (BOW) representation. The tokens can be encoded in the vector as binary values (yes/no occurrence in the document), numbers of occurrences in the document, or TF-IDF (term frequency-inverse document frequency) values. So, each document is represented using a large (many components) sparse (most values are zero as a particular token does not appear in the document) numeric vector. As this yields the "Curse of dimensionality" phenomenon, some dimensionality reduction methods are usually applied to the document vectors, either feature selection or feature transformation.

There are several drawbacks to the BOW representation. BOW cannot handle multi-word phrases and, in its basic form, cannot reflect the different importance of various parts of the text. The first problem can be dealt with by n-grams; instead of using single tokens to create features, we can create tokens for N subsequent words. The second problem can be handled by weighting tokens according to their appearance in different parts of the document or by considering only important parts of the documents. So e.g., we can expect that title, abstract, and keywords, which are obligatory parts of scientific papers, are closely related to the content of the paper; so only these parts can be considered when defining the tokens. There are also some techniques that can enhance the basic BOW representation: entity detection or semantic expansion (Li et al., 2021). It is also possible to use word embeddings. This relatively new concept from computational linguistics aims at describing semantic similarities between linguistic items (words) using their co-occurrence in large textual databases.

Text mining can be applied in a wide range of application domains. As our work is oriented on the multi-class classification of medical papers, we review some work in this area. Gani et al. (2016) used Naïve Bayes, Decision Tree, Support Vector Machine, and Stochastic Gradient Descent algorithm to classify documents related to 23 cardiovascular diseases taken from the MEDLINE database. They used BOW representation, where tokens are represented using TF-IDF values, and achieved the best classification accuracy of 76% for SVM and 3000 selected features (Jindal & Taneja, 2015b). Yan et al. describes a Convolutional Neural Network framework (B-CNN) for biomedicine semantic indexing (Yan et al., 2018). The proposed CNN architecture can adaptively deal with features of documents and can capture context information. They extend the features created using the BOW model by word sequence embedding. As they reported, this extension can improve the classification performance of CNN models but not of simpler models like Naive Bayesian classifier or

Logistic Regression (Balaji et al., 2020). Another example application of the CNN model to text classification is presented in Balaji et al. (2020). The authors of this paper used the word embedding method Word2vec for texts retrieved from the PubMed databases word embeddings were used to train a sentence-level classifier for texts that belong to one of 26 medical categories. Mujtaba et al. used a small subset of MEDLINE documents belonging to top-10 disease categories to Compare Bayesian Network, Decision Tree, and Random Forest models on a multi-class classification problem. They report, that in this particular task, Bayesian Network, when used on BOW representation without stemming, achieved the best performance (Mujtaba et al., 2019). Jindal and Taneja propose a lexical approach to text categorization in the biomedical domain. They represent the documents using tokens that are derived from words in the abstract by matching them with keywords taken from MeSH. They then used a modified K-Nearest Neighbor algorithm for the multi-class classification of the documents (Jindal & Taneja, 2015a). Elberichi, Amel, and Malika used MeSH ontology to enhance document representation of papers taken from biomedical benchmark text corpus Ohsumed (Elberrichi et al., 2012). They show that using hypernyms derived from ontology as additional features in document representation can improve classification performance.

2 Objectives

The work reported in the paper deals with the multi-class classification of biology and medical research papers published on issues related to the COVID-19 pandemic. As this pandemic threatens people in countries all over the world, a huge number of papers that refer to COVID-19 have been published. It is impossible to read all these research papers. So in our work, we try to classify these research papers depends not only on the bibliometric information but also on utilizing the internal context of these research papers. We then performed several ML experiments for different methods of document representation with the aim to identify the suitable combination of representation scheme and algorithm to classify COVID-19 related research papers into these classes.

3 Methods

The document processing pipeline is shown in Fig. 1. After the data enhancement, we extracted different types of entities from the different parts of the research papers: title, and abstract. We also consider various methods of encoding the features to represent the documents. These text pre-processing steps result in different data tables used in the machine learning experiments. We assess not only the quality of the created models but also aim at the interpretation of the results. The next subsections show details for each of these steps.

3.1 Corpora details

The LitCovid corpus is a collection of published PubMed research papers that are directly related to the novel Coronavirus that was discovered in 2019. The collection comprises over 23,000 articles, with about almost 2,000 new articles uploaded every week, making it a comprehensive resource for scholars keeping up to date on the current COVID-19 situation. The whole article or at least the abstract can be downloaded straight from LitCovid's website for a major number of research papers. We chose 23,038 articles with full texts or abstracts

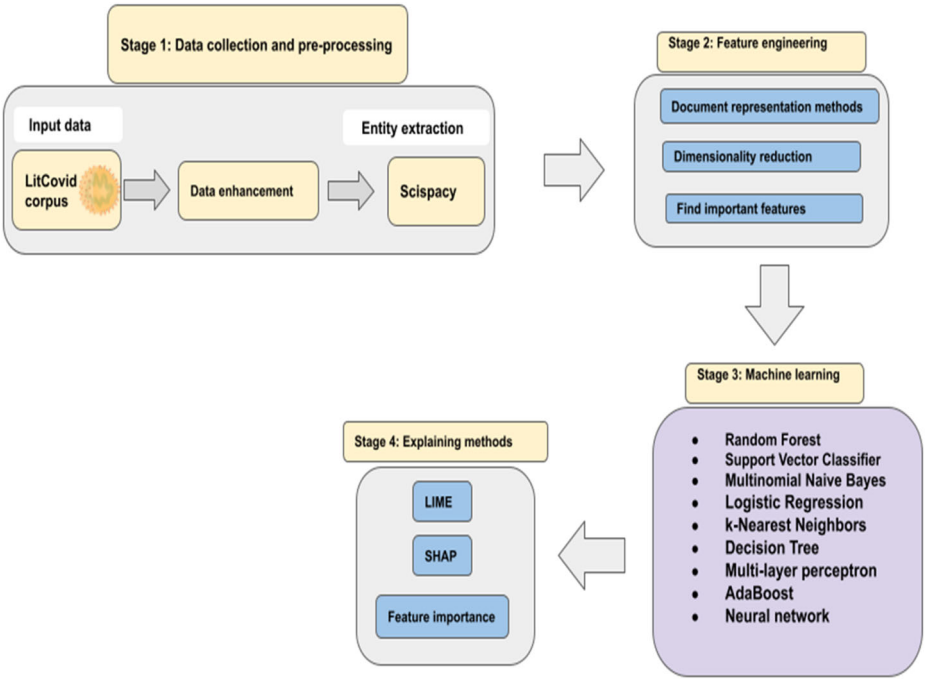

Fig. 1 Overview of methodological pipeline

from more than 35,000 articles for our document classification corpus. These articles have an average of 74 sentences and 1,399 tokens, indicating a fairly equal split between abstracts and full articles based on visual assessment. Prevention, Treatment, Diagnosis, Mechanism, Case Report, Transmission, Forecasting, and Generalcitep are the eight topic descriptors allocated to each research paper in LitCovid (Chen et al., 2021a, 2021b). Despite the fact that every research paper in the corpus can be tagged with numerous tags, the majority of research papers (about 76 %) only have one. We only utilized the research papers with one label in our experiment.

3.1.1 Data enhancement

We obtained and verify the authors, published year, type of paper, citations, references, type of journal, and journal name based on data from Google scholar (https://scholar.google.com/) and CORD-19 (CORD-19: COVID-19 Open Research Dataset)(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7251955/) in order to enhance the bibliometric data (Fig. 2).

3.1.2 Target classes

We used the LitCovid topic descriptors to define classes for our multi-class classification task. Out of the eight topic descriptors, we worked only with the Prevention, Treatment, Diagnosis, and Case Report classes. As we described before, our focus is on these four classes, and the majority of research papers (about 76 %) only have these classes. Table 1 shows the distribution of classes in the data that we used for the experiment. The classes are

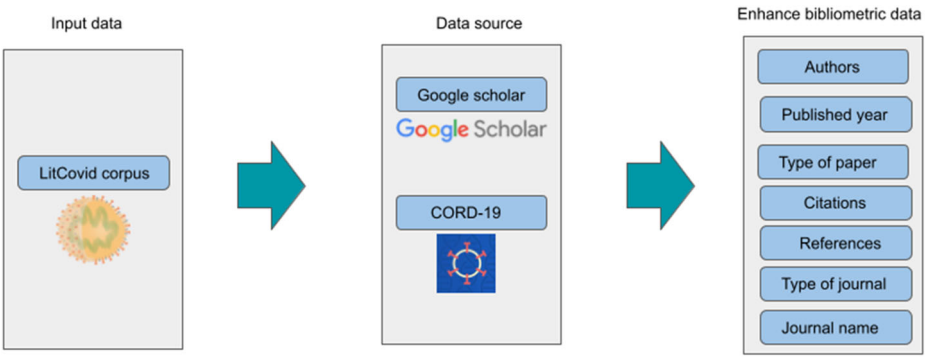

Fig. 2 Process of data enhancement

highly imbalanced with the most commonly occurring label appearing about five times as frequently as the least frequent one. We used a state-of-the-art algorithm SMOTE (Chawla et al., 2002) for the over-sampling of minority classes, to handle this problem. In the over-sampling technique with SMOTE, the synthetic samples are generated for the minority class. Instead of simply duplicating minority class examples, the algorithm generates new examples similar to those, existing in the data. To do this the algorithm first finds a close neighbor of a minority class example and then creates a new example that lies in between. So new minority class examples are obtained as interpolations of existing ones (Chawla et al., 2002).

3.2 Data pre-processing and feature engineering

At the beginning of the feature engineering, we remove the unwanted characters from the title and abstract. This helps to get more accurate features from the corpus. Also, we convert all the uppercase words to lowercase. We also drop the duplicate data (abstract and title) and remove the "NaN" values from the entire corpus. Also, removing the stop words did not create any huge impact on the accuracy but it had a huge impact on improving the model interpretability. After completing the basic data cleaning, the first set of features was derived from the bibliometric indicators and placed into the "Bibliometric data" table. This included the following information:

- Authors' number,
- Age of the publication,
- Type of paper,
- Number of citation,
- Number of references,

Table 1 Label count (used for this experiment) of the LitCovid corpus

Label	Count
Prevention	2599
Treatment	1454
Diagnosis	658
Case_report	542

- Type of publication,
- Tokenized journal name

As we described before, for our experiment we use the LitCovid corpus where we utilize the abstracts, title, and bibliometric information but are not able to process full text because of the low computational power and the full text is always not easy to find for reproducibility, where abstracts are almost always available.

3.2.1 Entity extraction

We used ScispaCy (Neumann et al., 2019), to extract entities from the research papers. ScispaCy is one of the most robust model pipelines for a variety of natural language processing tasks focused on biomedical text. It contains modules for part of speech tagging, dependency parsing, named entity recognition, and sentence segmentation (Neumann et al., 2019). We used four pre-trained ScispaCy models for named entity recognition in the research papers. We employed a transition-based method based on the chunking model for entity extraction (Lample et al., 2016) as implemented in ScispaCy. All four pre-trained ScispaCy models are depicted in Table 2. A partial sample of entities for a research paper "Gene expression in epithelial cells in response to pneumovirus infection" could be Respiratory syncytial virus, RSV, pneumonia virus, mice, PVM, viruses, family Paramyxoviridae, subfamily pneumovirus, respiratory infections, etc. All the entities were employed in forming the model (see Section 3.2.2).

3.2.2 Document representation

We used three alternative ways to represent the text-related features: binary word incidence approach (Zhang et al., 2010), TF-IDF approach (Aizawa, 2003; Muralikumar et al., 2017), and the embeddings-based approach (Tenney et al., 2019). The first two ways are related to the Bag-of-words (BOW) model. BOW is one of the simplest methods for document representation. In this model, a document is represented as a multi-set of words appearing in the document. In binary representation, only the presence or absence of a word in the document is encoded. TF-IDF reflects both the occurrence of a term in a particular document (as term frequency, TF) and the occurrence of this term in the whole collection of documents (as

Table 2 ScispaCy entity recognition systems used corpus

Training corpus	Entity types
CRAFT	GGP, SO, TAXON, CHEBI, GO, CL
JNLPBA	DNA, CELL_TYPE, CELL_LINE, RNA, PROTEIN
BC5CDR	DNA, CELL_TYPE, CELL_LINE, RNA, PROTEIN
BIONLP13CG	AMINO_ACID, ANATOMICAL_SYSTEM, CANCER, CELL, CELLULAR_COMPONENT, DEVELOPING_ANATOMICAL_STRUCTURE, GENE_OR_GENE_PRODUCT, IMMATERIAL_ANATOMICAL_ENTITY, MULTI-TISSUE_STRUCTURE, ORGAN, ORGANISM, ORGANISM_SUBDIVISION, ORGANISM_SUBSTANCE, PATHOLOGICAL_FORMATION, SIMPLE_CHEMICAL, TISSUE

List adapted from https://allenai.github.io/scispacy/

inverse document frequency, IDF). For both representations, we used the N-gram approach for the article titles and abstracts. According to multiple research projects, N-gram is one of the most used methods in the field of computational linguistics (Brown et al., 1992). For our experiment, we applied uni-gram, bi-gram, and tri-gram to find out the best possible entities from the title and abstract of a research paper.

Bidirectional Encoder Representations from Transformers (BERT) were employed for the embeddings-based method (Devlin et al., 2018). The BERT model is made by stacking up multiple encoders of the transformer architecture on top of one another. The BERT architecture is designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right contexts in all layers. We applied BERT Tokenizer based on WordPiece (Muller et al., 2019) for the title, and abstracts from the LitCovid corpus. We used a pre-trained model bert-base-uncased (Geetha & Renuka, 2021) and the pre-training was performed on a large corpus of English data (BookCorpus and English Wikipedia) in a self-supervised fashion (Geetha & Renuka, 2021; Turc et al., 2019).

3.2.3 Dimensionality of input corpus

To reduce the dimensionality, for the document representation input data table we used tree derived importance feature selection method, which is a very straight forward, fast, and general approach for selecting the good features for machine learning methods. In Table 3, we show the results of reduction applied to different input data tables. In their experiment, Beranova et al. demonstrated that 1500 features outperformed other numbers of features for the ScispaCy-related input data table (Beranová et al., 2022). For that reason, we utilize the 1500 features for all of our ScispaCy entity-based experiments. But for others (TF-IDF and BOW), we utilized all the features for our experiment. In Table 3, we also include the total feature count before the feature selection, which was considered as an input to reduce the size of the dataset to address the training time and scalability issues encountered with the different machine learning-based methods. Also, for the Neural Network-based

Table 3 Overview of input corpus in the machine learning methods

Corpus (Input data table)	Used text	Columns	Original columns
ScispaCy	Abstract	1500	195434
TF-IDF	Abstract	20939	20939
BOW	Abstract	20939	20939
ScispaCy	Title	1500	3430
TF-IDF	Title	326	326
BOW	Title	326	326
ScispaCy	Title_Abstract	1500	195760
TF-IDF	Title_Abstract	21264	21264
BOW	Title_Abstract	21264	21264
BOW	Abstract_BibliometricFeatures	20945	20945
TF-IDF	Abstract_BibliometricFeatures	20945	20945
BERT	Title	3072	30522
BERT	Title_and_Abstract	3072	30522
BERT	Title_and_Abstract_and_ BibliometricFeatures	3072	30522

model, the input data tables have been reduced because of training in a reasonable time. For the BOW Input data table and the TF-IDF Input data table, we utilize all features. As we described before, for the ScispaCy-related input data table we utilize the 1500 features and utilize the threshold value like Beranová et al. (2022). We also investigated the relationship between the dimension of the input data table and the accuracy of the Random Forest model trained on it. Table 7 shows the best accuracy with the different document representations with different machine learning-based methods, where the highest accuracy of 92% is most stably attained for a vector length of about 21264 for the TF-IDF document representation with the Random Forest method. Also, the ScispaCy related document representation (binary) with 1500 features got 79% accuracy. We used MDI feature importance scores to select the most 1500 important features. Other input data tables (such as Title and Abstract BibliometricFeatures) are constructed by merging the related input data tables.

3.3 Machine learning experiments

For the machine learning experiments, we used the Random Forest (Liaw et al., 2002; Breiman, 2001), Linear Support Vector classifier (Linear SVC) (Suthaharan, 2016), Multinomial Naive Bayes (Kibriya et al., 2004), Logistic Regression (Sperandei, 2014), K-Nearest-Neighbors (Fukunaga & Narendra, 1975), Support Vector Classifier (SVC) (Suthaharan, 2016), Decision Tree (Safavian & Landgrebe, 1991), Multi-layer Perceptron (MLP) (Taud & Mas, 2018) and Adaptive Boosting (AdaBoost) (Margineantu & Dietterich, 1997) classifier as implemented using the scikit-learn (https://scikit-learn.org/), a free software machine learning library for python. For training and testing purposes, we used the title, abstract and bibliometric information from each research paper. Because of the limited computational power, we were not able to use the full-body text for our experiments. Table 3, corpus(Input data table) column shows different variants of article representations we used as input data tables for machine learning. Here TF-IDF, BOW, and ScispaCy refer to methods used to represent information taken from the abstract or title of the articles (BOW stands for binary representation), and BibliometricFeatures refers to bibliometric information about the articles.

Convolutional Neural Networks (CNNs) outperform alternative neural network architectures such as LSTMs and Recurrent Neural Networks for classification tasks (Gu et al., 2018; Prusa & Khoshgoftaar, 2017). To encode our data, we created a BERT tokenizer and a pre-trained BERT model configuration. We used a function called "batch_encode_plus" to encode all of the titles and abstracts from the research paper, and we trained and validated the data individually. Table 4 shows how the model was learned for various combinations of hyperparameters.

We created eleven variants of the input data table altogether as input for machine learning methods. In Section 4.2, we will present a comparison regarding the variants of data tables and find which data table provides the best accuracy in our full experiment.

We also tuned different types of hyperparameters to find the best result using different machine learning methods. We utilize the K-fold cross-validation from the scikit-learn.[1] It provides us the cross-validation with random search and grid search hyperparameter optimization via the RandomizedSearchCV[2] and GridSearchCV[3] classes respectively. We used

[1] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Table 4 Overview of input Parameter grid

Machine learning algorithm	Parameter grid
Random Forest	• 'max_depth': 10,150,500,1000 • 'max_features': 30,500,3000 • 'min_samples_leaf': 1,10,100 • 'min_samples_split': 2,10,100 • 'n_estimators': 10, 100
Logistic Regression	• 'random_state': 0
K-Nearest-Neighbors	• 'n_neighbors': 3
SVC	• 'gamma': 2 • 'C': 0.025, 1 • 'kernel': linear
Decision Tree	• 'max_depth': 5, 10, 15
Multi-layer Perceptron	• 'alpha': 1 • 'max_iter': 1000
Neural Network (BERT)	• 'max_length': 256 • 'epochs': 5 • 'lr': 1e-5 • 'eps': 1e-8

the inner loop of nested cross-validation where the training dataset was defined by the outer loop. We also configure the hyperparameter search to refit a final model with the entire training dataset using the best hyperparameters. As we describe before, we utilize nested cross-validation for fine-tuning the hyperparameters. Nested cross-validation is an approach for model hyperparameter optimization that attempts to overcome the problem of overfitting the training dataset. The procedure involves treating model hyperparameter optimization as part of the model itself and evaluating it within the broader K-fold cross-validation procedure for evaluating models for comparison and selection. A set of different hyperparameters for the different machine learning methods were optimized according to the grid that we present in Table 4.

3.4 Explanation algorithms

The question of interpretability or explainability of the created models becomes a hot topic in the area of machine learning. The end-users are interested not only in the quality of the models but also in the insight into the classification process. Some models are easy to understand by their nature (typical examples are Decision Trees), but some models, typically Neural Networks, work as a black-box model.

The Random Forest method was designed for the calculation of the feature importance scores. However, due to the number of trees with their complexity, and also because the multiple trees can take part in a decision, direct interpretation of the Random Forest models is not possible (Breiman, 2001). In our work, we adopt the original method for computing the feature importance scores of Random Forest, which is based on the Mean Decrease of

Impurity (MDI). For this method, it has been shown that the MDI importance of a relevant feature is invariant with respect to the removal or addition of irrelevant features and that the importance of a feature is zero if and only if the feature is irrelevant (Louppe et al., 2013). LIME (Ribeiro et al., 2016) and SHAP (Lundberg et al., 2020) methods were also used to interpret the created models.

LIME Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016) is an interpretability surrogate model which can be used on any black-box model to provide local interpretability for the result of prediction or classification of a single instance. The idea is to explain this prediction using a simpler (usually linear) model that has been created for a sample of the original data. Higher weights are assigned to examples that are like the instance in question. Based on the linear model, we can assess the contribution of each individual feature to the result of classification.

SHAP SHapley Additive exPlanations (SHAP) (Lundberg et al., 2020) is a game theory-based method for interpreting any machine learning model's output. It uses the traditional Shapley values from game theory and their related extensions to correlate optimal credit allocation with local explanations.

4 Results

4.1 Evaluation metrics

In order to evaluate the performance of our proposed model, we used overall Accuracy (A), Precision (P), Recall (R), and F1-score (F1). Overall accuracy is simply the proportion of correct classifications. Precision and Recall have been proposed in the area of information retrieval to evaluate the quality of search in a corpus of documents. In this original setting, Precision represents the proportion of relevant documents in the set of retrieved documents, and Recall represents the proportion of retrieved relevant documents in the set of all relevant documents. When adapted to evaluate results of classification, Precision is defined as the proportion of examples correctly classified to a given class in the set of examples classified to this class (formula (2)), and Recall is defined as the proportion of examples correctly classified to a given class in the set of all examples of this class (formula (3)). Here, TP stands for the number of examples correctly classified to a given class, FP stands for the number of examples incorrectly classified to a given class (so TP + FP is the number of examples classified to this class) and FN stands for the number of examples that were not classified to a given class (so TP + FN is the number of examples that belong to this class). Finally, F1-score is the harmonic mean of Precision and Recall (formula (4)).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 * TP}{2 * TP + FP + FN} \tag{4}$$

Precision, Recall, and F1-score are defined for each class separately. To evaluate a multi-class classifier, these characteristics can be extended to macro average or weighted average values. The macro average is just an arithmetic mean. A weighted average is obtained as a weighted sum where weights are relative frequencies of classes.

4.2 Predictive performance

We used 70% training data and 30% test data to evaluate the effectiveness of different machine learning based approaches. The overall accuracy was used to evaluate the results, but we also computed the per-class accuracy. As previously stated, this dataset was unbalanced, and we used the oversampling approach to solve this. Tables 5 and 6 shows the accuracy (per-class accuracy) of the Neural Network (BERT) and Random Forest approach, whereas Table 7 shows the overall accuracy of the other methods.

As we see in Table 7, out of the traditional multi-purpose machine learning algorithms, the Random Forest method outperforms others. The BERT over Neural Network, a modern method used for NLP, achieved similar performance. But not only the classification accuracy should be considered when comparing the results of different methods. Another issue that should be taken into account is the complexity of the used data and the complexity of created models. Concerning the complexity of data, Table 3 indicates, that BERT was trained on less complex data representation than was the TF-IDF representation used for Random Forest. The complexity of the models can be assessed according to the number of nodes (neurons in the Neural Network or branching nodes in the trees in the forest). The last important issue when evaluating the models is their understandability and interpretability. More on this is presented in the next subsection.

4.3 Model interpretation

As stated earlier, we used SHAP and LIME to get more insight into the created models. As an illustration, we present an interpretation of the Random Forest model. Feature importance scores were used to analyze Random Forest models. We utilized the MDI method to find

Table 5 Accuracy with Neural Network (BERT)

Input data table		Per class accuracy	Accuracy
Title	prevention	0.92	
	treatment	0.77	0.76
	diagnosis	0.69	
	case_report	0.67	
Title_and_Abstract	prevention	0.95	
	treatment	0.80	0.87
	diagnosis	0.86	
	case_report	0.85	
Title_and_Abstract_and_biblo	prevention	0.92	
	treatment	0.67	0.74
	diagnosis	0.59	
	case_report	0.79	

Table 6 Accuracy with Random Forest

Input data table	Used text	Class	Per class accuracy	Accuracy
ScispaCy	Abstract	prevention	0.76	
		treatment	0.69	0.74
		diagnosis	0.75	
		case_report	0.74	
TF-IDF	Abstract	prevention	0.96	
		treatment	0.92	0.92
		diagnosis	0.90	
		case_report	0.89	
BOW	Abstract	prevention	0.93	
		treatment	0.90	0.90
		diagnosis	0.92	
		case_report	0.87	
ScispaCy	Title	prevention	0.64	
		treatment	0.50	0.57
		diagnosis	0.62	
		case_report	0.47	
TF-IDF	Title	prevention	0.83	
		treatment	0.81	0.80
		diagnosis	0.81	
		case_report	0.77	
BOW	Title	prevention	0.71	
		treatment	0.66	0.70
		diagnosis	0.78	
		case_report	0.64	
ScispaCy	Title_and_Abstract	prevention	0.81	
		treatment	0.76	0.79
		diagnosis	0.80	
		case_report	0.79	
TF-IDF	Title_and_Abstract	prevention	0.96	
		treatment	0.92	0.92
		diagnosis	0.91	
		case_report	0.9	
BOW	Title_and_Abstract	prevention	0.93	
		treatment	0.90	0.91
		diagnosis	0.92	
		case_report	0.87	
BOW	Title_and_Abstract_and_	prevention	0.82	
	Bibliometric Features	treatment	0.73	0.73
		diagnosis	0.67	
		case_report	0.72	

Table 6 (continued)

Input data table	Used text	Class	Per class accuracy	Accuracy
TF-IDF	Title_and_Abstract_and_ Bibliometric Features	prevention	0.96	0.92
		treatment	0.93	
		diagnosis	0.90	
		case_report	0.89	

the relevant characteristics, as explained previously. Table 8 shows three example matrices from a total of eleven matrices.

SHAP can detect the direction of feature significance in the same way as the MDI technique can. A SHAP plot in Figs. 3 and 4 illustrates the significance of specific TF-IDF (title) and TF-IDF (abstracts). Additionally, it is the first evidence of the connection between a feature's importance and its influence on a prediction. This SHAP plot combines the significance of the features with their impacts. A Shapley value for a feature and an instance may be found at each point on the summary plot. The feature determines the location on the

Table 7 Best accuracy with different machine learning algorithm for different text representation method

Input data table	Used text	Machine learning algorithms	Accuracy	Macro average (F1 score)	Weighted average (F1 score)
ScispaCy	Abstract	Random Forest Classifier	0.74	0.74	0.74
TF-IDF	Abstract	Random Forest Classifier	0.92	0.92	0.92
BOW	Abstract	Random Forest Classifier	0.90	0.90	0.90
ScispaCy	Title	Multinomial NB	0.68	0.68	0.68
TF-IDF	Title	Random Forest Classifier	0.80	0.80	0.80
BOW	Title	Random Forest Classifier	0.70	0.70	0.70
ScispaCy	Title and Abstract	Random Forest Classifier	0.79	0.79	0.79
TF-IDF	Title and Abstract	Random Forest Classifier	0.92	0.92	0.92
BOW	Title and Abstract	Random Forest Classifier	0.91	0.91	0.91
BOW	Abstract with Bibliometric Features	Random Forest Classifier	0.73	0.73	0.73
TF-IDF	Abstract with Bibliometric Features	Random Forest Classifier	0.92	0.92	0.92
Bidirectional Encoder Representations	Title_Abstract	Neural Network (BERT)	0.87	0.87	0.87

Table 8 Top most important features (MDI method) by input data table

ScispaCy	Imp.	TF-IDF	Imp.	TF-IDF and Bibliometric	Imp.
Pandemic	0.004	year old	0.005	case	0.005
Fever	0.003	report case	0.004	year	0.005
Patient	0.003	fever	0.004	pcr	0.004
Respiratory distress syndrome	0.003	report	0.003	old	0.003
Drug	0.002	cough	0.003	present case	0.003
Covid-19	0.002	trials	0.003	acute	0.003
Hospital	0.002	measures	0.003	report	0.003
Patient	0.002	inhibitors	0.003	sars cov	0.003
Patient covid-19	0.002	cov	0.003	personal	0.003
Cell	0.002	polymerase chain reaction	0.003	sars	0.003
People	0.002	year	0.003	ct	0.003
Ace2	0.001	rt	0.002	inhibitors	0.003
Region	0.001	antiviral	0.002	covid 19 pandemic	0.002
Pneumonia	0.001	cov infection	0.002	clinical trials	0.002
Coronavirus	0.001	covid 19 pandemic	0.002	ace2	0.002
Protein	0.001	sars cov	0.002	cells	0.002
Recipient	0.001	receptor	0.002	specificity	0.002
Cytokine	0.001	angiotensin converting enzyme	0.002	distancing	0.002
Infection	0.001	therapeutic	0.002	report case	0.002
Protease	0.001	ground	0.002	healthcare	0.002
Coronavirus 2019 disease	0.001	anti	0.002	therapeutic	0.002

y-axis, while the Shapley value determines the position on the x-axis. From low to high, the color indicates the value of the feature. We can observe how the distribution of the Shapley values for each feature is distributed since overlapping points are jittered in the y-axis direction. The features are listed in ascending order of significance.

To get an overview of which features are most important for a model we plot the SHAP values of important features for titles and abstracts using TF-IDF. Figures 3 and 4 sort features by the sum of SHAP value magnitudes over all samples and uses SHAP values to show the distribution of the impacts the features have on the model's output. The color represents the feature value whereas the red color shows greater values than values in blue. Case, patient, covid 19, infection, etc are one of the most important features for the titles using TF-IDF, and year old, case, stars cov, etc are one of the most important features for the abstracts using TF-IDF for this research paper's classification task. We also include the LIME figure in Figs. 5 and 6 to explain the prediction of the Random Forest model for a sample research article based on its title and abstract.

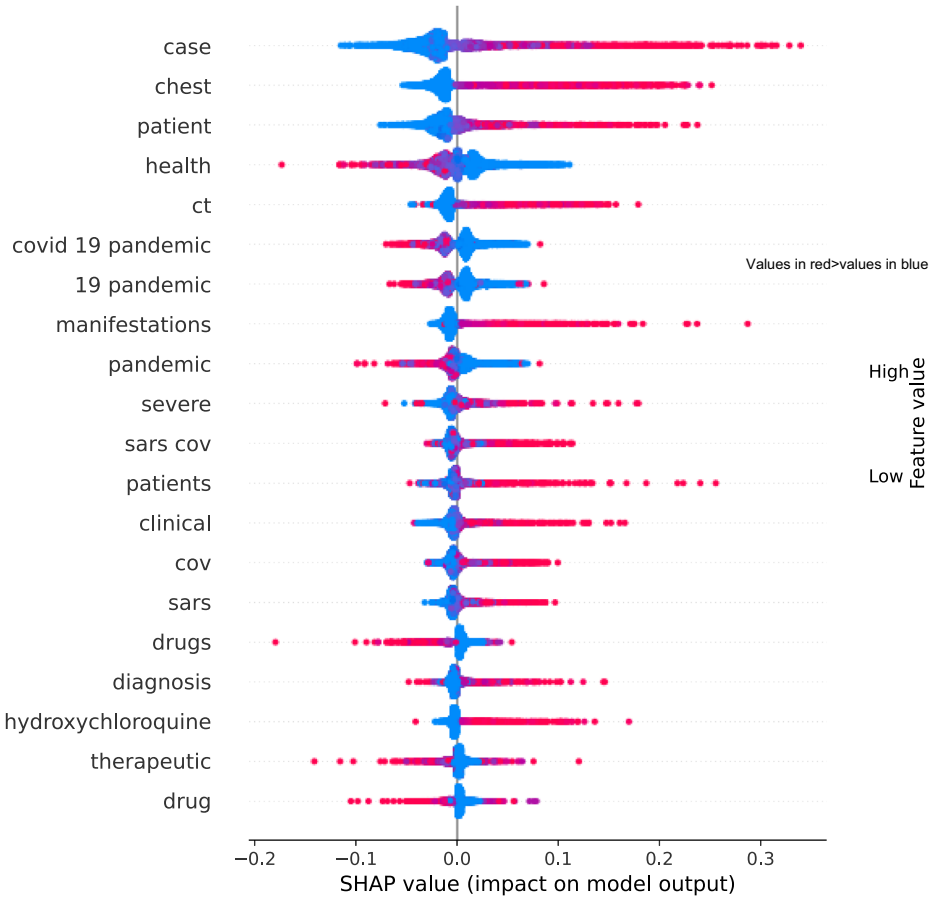

Fig. 3 SHAP plot for titles (using TF-IDF)

5 Discussion

The average overall accuracy was very different depending on the form of document representation; values ranged from 0.41 to 0.92. The document was represented using TF-IDF and bibliometric features in several input data tables with the abstract, title_abstract, and title_abstract_bibliometric features. The high-dimensional input data table is used for a wide range of purposes. The major goal is to see how the entity extraction method, document representation method, and bibliometric features affect the multi-class classification of the research paper. The interesting finding is that when we only use the retrieved entities (ScispaCy entities) from the abstract and title, we always obtain low accuracy. As a result, only utilized entities have little bearing on the multi-class classification task. As we discussed before, we used multiple input data tables for all of the available bibliometric features, and the best result was 0.92 (TF-IDF with title_abstract_bibliometric features). The abstract only and title-and-abstract combined input data tables yield comparable results. When we simply use the title data table, we receive less accuracy than when we only use the abstract

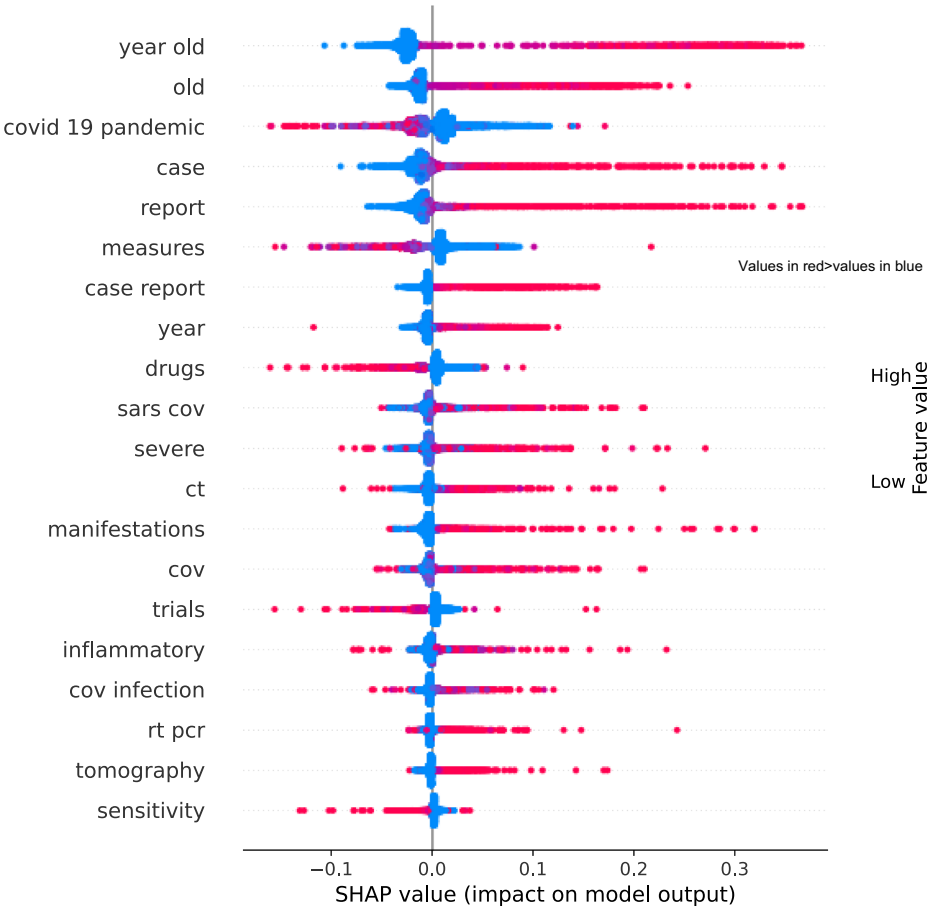

Fig. 4 SHAP plot for abstracts (using TF-IDF)

data table. We may conclude from this experiment that the title has a smaller influence on classification accuracy than the abstract alone.

In order to generalize our findings by evaluating several document representation types, we discovered that

Fig. 5 LIME plot for title

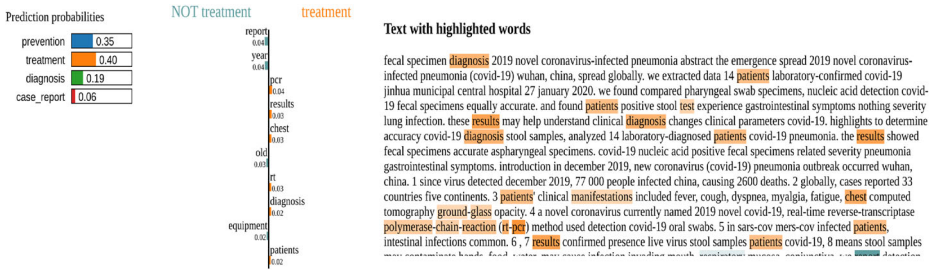

Fig. 6 LIME plot for abstract

- The context (Such as title, abstract, etc) of a research paper creates more influence than the bibliometric information on the research paper's classification.
- The information derived only from abstracts is more relevant than information derived only from the title,
- Extracted entities from either abstract or title with binary document representation are less effective than the use of abstract or title with TF-IDF document representation.

On the other side, the Neural Network (BERT) outperforms (with the exception of the Random Forest method) the other machine learning methods when using TF-IDF document representation, where we utilize three distinct input data tables (title, title_abstract, and title_abstract_some_bibliometric info). In this case, the title_abstract outperforms the others, and also it is closely similar to the Random Forest approach with TF-IDF document representation. We were also able to identify algorithms that outperformed the others for a specific document representation scheme. However, Random Forest and Neural Networks (BERT) display high performance across all document representation types.

5.1 Error analysis

To further understand the performance of the top-scoring Random Forest models, we additionally try to analyze the errors they made using TF-IDF document representation. First, we observe that these models frequently correlate several categories namely, diagnosis and treatment much more closely than is required. Despite the fact that there is some overlap

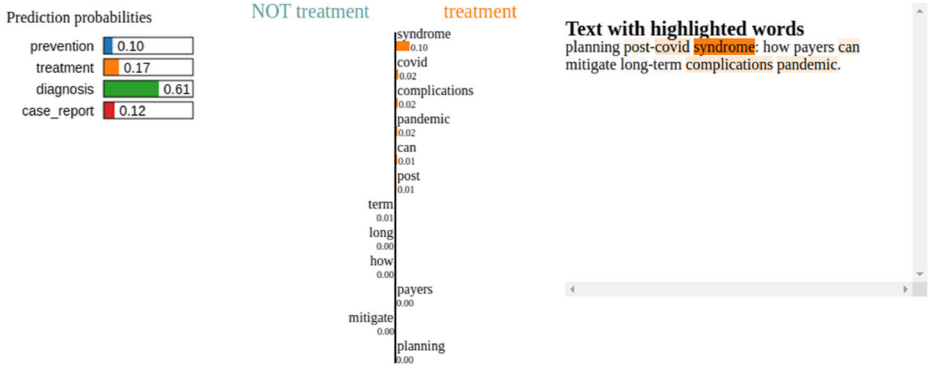

Fig. 7 LIME plot for an example error

Table 9 LitCovid error samples

DOI	Title	Label	Prediction
https://doi.org/10.1136/emermed-2020-209797	strategic planning response covid-19 london em...	Prevention	Treatment
https://doi.org/10.1007/s11606-020-06042-3	planning post-covid syndrome: how payers can m...	Treatment	Diagnosis
https://doi.org/10.1007/s10877-020-00550-7	covid-19: pulse oximeters spotlight.	Treatment	Diagnosis

and a semantic relationship between these groups. Future work should attempt to explicitly model correlation between categories to help the model recognize the particular cases in which labels should occur together. Finally, we observe that models have trouble identifying discriminative sections of the research paper due to how much introductory content on the pandemic can be found in most articles. Future work also should explicitly model the gap in relevance between introductory sections and crucial sentences such as research paper abstracts and titles. In Table 9, we provide three different example error samples. In the first research paper, the Random Forest method classified it as a treatment instead of prevention. Also for the second and third research papers, Random Forest classified it as a prevention and diagnosis instead of treatment. In the future, We also try to find the errors with the impact of the outside knowledge base in this classification task. In some cases, full text from the research papers is needed to make better predictions with the title, abstract and bibliometric information. We also include a LIME figure in Fig. 7 is an example of an error to explain the prediction of the Random Forest model for a sample research article based on its title.

6 Conclusion and future work

Our work deals with classifying the whole documents into classes that reflect important COVID-19 related issues like diagnosis, treatment, case reports, or prevention. The proposed pipeline can thus help medical practitioners to filter out research papers dealing with these issues from a massive amount of COVID-19 related research papers. In the future, we also plan to apply different rule mining methods to find the different patterns from the issues like diagnosis, treatment, case report, or prevention. Also, the found papers can present different opinions on the same issue bringing controversial information, the next step in using text mining techniques to support work with the documents can be sentiment analysis (Liu, 2012). Sentiment analysis can be turned into the question of whether a piece of text is expressing positive, negative, or neutral sentiment towards the discussed topic and can be thus understood as a knowledge-based classification problem. Sentiment analysis can be performed at the document level, at the sentence level, or at the aspect level. To apply sentiment analysis techniques, not only the texts themselves but also some sentiment lexicons that contain representative words for different opinion polarities must be used. Also, the MeSH thesaurus can be useful for sentiment analysis of COVID-19 related documents when considering the aspect level.

Acknowledgements The authors also would like to thank Vojtěch Svátek, Tomáš Kliegr, and Lucie Beranová for providing helpful feedback on the first version of the manuscript and again Vojtěch Svátek for the second version of the manuscript.

Author Contributions Gollam Rabby works with conception, writing code, experiments, manuscript text, and editing. Petr Berka works with the conception, manuscript text, managed the research, and editing of the article. All authors read and approved the final manuscript.

Funding Gollam Rabby was partly supported by grant IGA 40/2021 “Action rules for text”. Petr Berka was supported by the Faculty of Informatics and Statistics, Prague University of Economics and Business, through long-term support for research activities.

Data Availability The research has been conducted only with openly available software packages and all the scripts, results, dataset after pre-processing and feature engineering will be publicly available in (https://github.com/corei5/Multi-Class-Classification-of-COVID-19-Documents) from the authors after accepting the manuscript.

Declarations

Consent for Publication We give our consent for the publication of identifiable details, which can include all the text ("Material") to be published in the above Journal and Article.

Competing interests All authors declare that they have no conflicts of interest.

References

Aizawa, A. (2003). An information-theoretic perspective of TF–IDF measures. *Information Processing & Management*, *39*(1), 45–65. https://doi.org/10.1016/S0306-4573(02)00021-3.

Balaji, V., Suganthi, S., Rajadevi, R., & et al. (2020). Skin disease detection and segmentation using dynamic graph cut algorithm and classification through naive bayes classifier. *Measurement*, *163*, 107–122. https://doi.org/10.1016/j.measurement.2020.107922.

Beranová, L., Joachimiak, M. P., Kliegr, T., & et al. (2022). Why was this cited? explainable machine learning applied to COVID-19 research literature. *Scientometrics*, 1–37. https://doi.org/10.1007/s11192-022-04314-9.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324.

Brown, P. F., Della Pietra, V. J., Desouza, P. V., & et al. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, *18*(4), 467–480. https://aclanthology.org/J92-4003.pdf.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & et al. (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953.

Chen, Q., Allot, A., Leaman, R., & et al. (2021a). Overview of the BioCreative VII LitCovid track: multi-label topic classification for COVID-19 literature annotation. In *Proceedings of the 7th BioCreative challenge evaluation workshop*. https://doi.org/10.1093/database/baac069.

Chen, Q., Allot, A., & Lu, Z. (2021b). LitCovid: an open database of COVID-19 literature. *Nucleic Acids Research*, *49*(D1), D1534–D1540. https://doi.org/10.1093/nar/gkaa952.

Devlin, J., Chang, M. W., Lee, K., & et al (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:181004805, https://doi.org/10.48550/arXiv.1810.04805.

Elberrichi, Z., Amel, B., & Malika, T. (2012). Medical documents classification based on the domain ontology mesh. arXiv:12070446, https://doi.org/10.48550/arXiv.1207.0446.

Fukunaga, K., & Narendra, P. M. (1975). A branch and bound algorithm for computing k-nearest neighbors. *IEEE Transactions on Computers*, *100*(7), 750–753. https://doi.org/10.1109/T-C.1975.224297.

Gani, A., Siddiqa, A., Shamshirband, S., & et al. (2016). A survey on indexing techniques for big data: taxonomy and performance evaluation. *Knowledge and Information Systems*, *46*(2), 241–284. https://doi.org/10.1007/s10115-015-0830-y.

Geetha, M., & Renuka, D. K. (2021). Improving the performance of aspect based sentiment analysis using fine-tuned bert base uncased model. *International Journal of Intelligent Networks*, *2*, 64–69. https://doi.org/10.1016/j.ijin.2021.06.005.

Gu, J., Wang, Z., Kuen, J., & et al. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354–377. https://doi.org/10.1016/j.patcog.2017.10.013.

Jindal, R., & Taneja, S. (2015a). A lexical approach for text categorization of medical documents. *Procedia Computer Science*, *46*, 314–320. https://doi.org/10.1016/j.procs.2015.02.026.

Jindal, R., & Taneja, S. (2015b). Ranking in multi label classification of text documents using quantifiers. In *2015 IEEE international conference on control system, computing and engineering (ICCSCE)* (pp. 162–166). IEEE, https://doi.org/10.1109/ICCSCE.2015.7482177.

Kibriya, A. M., Frank, E., Pfahringer, B., & et al. (2004). Multinomial naive bayes for text categorization revisited. In *Australasian joint conference on artificial intelligence* (pp. 488–499). Springer. https://doi.org/10.1007/978-3-540-30549-1_43.

Lample, G., Ballesteros, M., Subramanian, S., & et al. (2016). Neural architectures for named entity recognition. arXiv:160301360, https://doi.org/10.18653/v1/N16-1030.

Li, W., Saigo, H., Tong, B., & et al. (2021). Topic modeling for sequential documents based on hybrid inter-document topic dependency. *Journal of Intelligent Information Systems*, *56*(3), 435–458. https://doi.org/10.1007/s10844-020-00635-4.

Liaw, A., Wiener, M., & et al. (2002). Classification and regression by randomforest. *R News*, *2*(3), 18–22. https://cogns.northwestern.edu/cbmg/LiawAndWiener2002.pdf.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, *5*(1), 1–167. https://doi.org/10.2200/S00416ED1V01Y201204HLT016.

Louppe, G., Wehenkel, L., Sutera, A., & et al. (2013). Understanding variable importances in forests of randomized trees. *Advances in Neural Information Processing Systems, 26*. https://doi.org/10.5555/2999611.2999660.

Lundberg, S. M., Erion, G., Chen, H., & et al. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, *2*(1), 2522–5839. https://doi.org/10.1038/s42256-019-0138-9.

Margineantu, D. D., & Dietterich, T. G. (1997). Pruning adaptive boosting. In *ICML* (pp. 211–218). Citeseer. https://doi.org/10.5555/645526.757762.

Mujtaba, G., Shuib, L., Idris, N., & et al. (2019). Clinical text classification research trends: Systematic literature review and open issues. *Expert Systems with Applications*, *116*, 494–520. https://doi.org/10.1016/j.eswa.2018.09.034.

Muller, B., Sagot, B., & Seddah, D. (2019). Enhancing BERT for lexical normalization. In *The 5th workshop on noisy user-generated text (W-NUT)*. https://doi.org/10.18653/v1/D19-5539.

Muralikumar, J., Seelan, S. A., Vijayakumar, N., & et al. (2017). A statistical approach for modeling inter-document semantic relationships in digital libraries. *Journal of Intelligent Information Systems*, *48*(3), 477–498. https://doi.org/10.1007/s10844-016-0423-6.

Neumann, M., King, D., Beltagy, I., & et al. (2019). Scispacy: fast and robust models for biomedical natural language processing. arXiv:190207669, https://doi.org/10.48550/arXiv.1902.07669.

Prusa, J. D., & Khoshgoftaar, T. M. (2017). Improving deep neural network design with new text data representations. *Journal of Big Data*, *4*(1), 1–16. https://doi.org/10.1186/s40537-017-0065-8.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). https://doi.org/10.1145/2939672.2939778.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems Man, and Cybernetics*, *21*(3), 660–674. https://doi.org/10.1109/21.97458.

Sperandei, S. (2014). Understanding logistic regression analysis. *Biochemia Medica*, *24*(1), 12–18. https://doi.org/10.11613/BM.2014.003.

Suthaharan, S. (2016). Machine learning models and algorithms for big data classification. *Integr Ser Inf Syst*, *36*, 1–12. https://link.springer.com/book/10.1007/978-1-4899-7641-3.

Taud, H., & Mas, J. (2018). Multilayer perceptron (mlp). pp 451–455. https://doi.org/10.1007/978-1-4842-4470-8_31.

Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. arXiv:190505950, https://doi.org/10.18653/v1/P19-1452.

Turc, I., Chang, M. W., Lee, K., & et al. (2019). Well-read students learn better: On the importance of pre-training compact models. arXiv:190808962, https://paperswithcode.com/paper/?openreview=BJg7x1HFvB.

Yan, Y., Yin, X. C., Yang, C., & et al. (2018). Biomedical literature classification with a CNNS-based hybrid learning network. *PloS ONE*, *13*(7), 93–97. https://doi.org/10.1371/journal.pone.0197933.

Zhang, Y., Jin, R., & Zhou, Z.H. (2010). Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, *1*(1), 43–52. https://doi.org/10.1007/s13042-010-0001-0.