Data Sampling and Supervised Learning for HIV Literature Screening

Hayda Almeida, Marie-Jean Meurs*, Leila Kosseim, and Adrian Tsang

***Abstract*—This paper presents a supervised learning approach to support the screening of HIV literature. The manual screening of biomedical literature is an important task in the process of systematic reviews. Researchers and curators have the very demanding, time-consuming and error-prone task of manually identifying documents that should be included in a systematic review concerning a specific problem. We developed a supervised learning approach to support screening tasks, by automatically flagging potentially relevant documents from a list retrieved by a literature database search. To overcome the main issues associated with the automatic literature screening task, we evaluated the use of data sampling, feature combinations, and feature selection methods, generating a total of 105 classification models. The models yielding the best results were composed of a Logistic Model Trees classifier, a fairly balanced training set, and feature combination of Bag-Of-Words and MeSH terms. According to our results, the system correctly labels the great majority of relevant documents, making it usable to support HIV systematic reviews to allow researchers to assess a greater number of documents in less time.**

***Index Terms*—Artificial intelligence, health information management, HIV, machine learning, text classification, triage**

I. Introduction

OPEN literature repositories are usually the main source of knowledge used by scientific researchers. Life science and biomedical databases contain a large number of documents, and are rapidly growing reflecting the pace of scientific publications and easy access to online repositories. The screening of scientific literature is typically performed by researchers to identify relevant studies for a given topic and support systematic reviews. PubMed [1], one of the largest open scientific databases, contained over 25 million citations of biomedical literature as of January 2016. Research programs dedicated to study public health generally need to manipulate and analyze large amounts of data to support processes such as systematic reviews of biomedical literature [2]. Following the publication speed of scientific literature, the available literature related to HIV and AIDS research is vast and rapidly increasing. In the year 2000, around 10k HIV related articles were added to PubMed, while over 16k HIV related articles were included in 2014. Querying the PubMed database with the string `HIV`, retrieves over 295k documents, while the query `AIDS` brings more than 238k documents.

Manuscript received April, 2016; revised June, 2016. *Asterisk indicates corresponding author*.

H.Almeida and A. Tsang are with the Centre for Structural and Functional Genomics (CSFG), Concordia University, Montreal, QC, Canada.

*M.-J. Meurs is with the Department of Computer Science, Université du Québec à Montréal, Montreal, QC, Canada, and a member of CSFG.

Leila Kosseim is with the Department of Computer Science and Software Engineering, Concordia University, Montreal, QC, Canada.

Performing systematic reviews of such volumes of data can be overwhelming for scientific researchers. A detailed description of the systematic review workflow is given in [3]. The initial step of the process is to define the research problem, and to search for eligible literature by querying scientific databases. The next step is to select studies, a task that requires exhaustive screening of a document list that was retrieved by strategic searches made by researchers. The potentially interesting documents retrieved for a given topic are usually numerous. Only after reviewing their abstracts is it possible to determine their potential to be considered relevant for the research topic. This task is a severe bottleneck in the information discovery process. Such a challenging manual task could greatly benefit from an automatic approach to support researchers in the literature triage. Automating literature screening can affect directly the coverage and quality of knowledge discovery in scientific research programs, since the number of documents to be evaluated during the filtering step is commonly very high.

The evaluation of biomedical data is highly relevant to assist the information discovery process in biomedical research (e.g. [4], [5]). In addition, several studies described the usefulness of automating the process of literature handling and screening (e.g. [6]–[8]). Machine learning approaches have been applied to support systematic reviews by performing literature screening (e.g. [9], [10]). In particular, supervised learning approaches can be beneficial for this task, since the use of classification models allows scientists to quickly evaluate many documents, reducing their manual effort. Automatic literature classification also reduces the possibility of missing relevant information, as a system-based screening might be less error-prone than a manual screening [11].

Substantial efforts have been put into extracting and annotating information on life science related documents [12], [13], with the use of natural language processing approaches [14]. The BioCreative initiative [15] represents the current extensive effort in the study of biomedical text classification. Automatic classification of bioliterature was specially evaluated at some of the BioCreative challenges [16], [17]. The tools developed at BioCreative were fairly generic. The systems proposed so far focused more on providing users with off-the-shelf solutions, based on a large set of generic supporting tools for biomedical literature classification tasks, as opposed to tools tailored to meet the particular issues related to automatic screening. This project specifically addresses the problem of bioliterature text classification, and the specific task challenges, by designing a problem-oriented supervised learning model.

The development of an automatic approach to perform literature screening can pose several challenges. One of the main issues is the underlying distribution of the data. Given a list of documents retrieved by a query search, researchers usually label most of them as *excluded*, and only a small portion is selected as relevant, and labeled as *included*. Since only a few documents are considered important, and many are filtered out at this phase, the literature screening task generally handles datasets presenting a class distribution imbalance. A dataset is considered imbalanced when the difference between the number of documents belonging to each class is so severe that it interferes in the machine learning process [18]. The class imbalance introduces noise in the dataset, and affects directly the performance of supervised learning methods. Classification algorithms tend to maximize the overall accuracy, therefore favouring the most frequent class while overlooking the least represented class [19]. Studies on imbalanced learning have evaluated different techniques to overcome the effect of the difference between class distributions (see Section II).

Another challenge related to automatic literature screening is the definition of a relevant feature subset. The use of large datasets in classification tasks results in models with an extensive number of features. Many of these features will likely be noisy or barely discriminative, thus only adding computational cost to the task. Moreover, a highly dimensional classification model may use an excessive number of features, which over-fits the training data, and interferes in the performance of classification algorithms. Feature selection methods are strategies applied in classification models to identify a subset of features that most suits a given task. Feature selection reduces the size of the feature space by keeping only the most relevant features for a specific problem.

In this work, we investigated the use of imbalanced learning strategies and feature selection methods applied to text classification with the goal of supporting HIV literature screening.

II. Related Work

Designing a supervised learning model to support the manual screening of biomedical literature can be challenging. The two main issues related to this task are the imbalanced class distribution in the dataset, and the selection of a relevant subset of features. We studied imbalanced learning and feature selection techniques as methods to overcome these conditions.

A. Imbalanced Learning

A dataset with a realistic class distribution of HIV literature screening presents a strong imbalance between *included* and *excluded* class labels. Datasets with imbalanced class distributions are commonly found in a variety of fields such as speech recognition [20], medical diagnosis [21], and fraud and image detection [22].

The class imbalance in the data greatly affects the classifier performance because *excluded* documents are massively represented in the dataset when compared to the number of documents belonging to the *included* class. Therefore, the classification model has many more examples of the majority class to learn from, and this introduces a bias in the prediction process. The imbalance dataset issue has been studied, and pointed out as an important issue in supervised learning (e.g. [23], [24]). Various approaches have been proposed in the field to overcome this issue. Cost-sensitive classifiers and data-sampling are the most common methods used. Cost-sensitive methods are implemented at the algorithm level, while sampling methods are implemented at the data level. The strategy used by cost-sensitive classifiers [25] is to lower classification errors in the minority class by intentionally introducing a bias during the learning phase so that classification errors made in the minority class are more costly than errors made in the majority class. Data sampling methods were first presented and discussed by [26] which describes the two most popular sampling strategies: undersampling and oversampling. Oversampling consists of adding documents to the minority class by generating new synthetic documents; whereas, undersampling consists of discarding documents from the majority class. Both techniques are used until a certain class distribution balance is reached.

Maloof [25] and Borrajo et al. [27] pointed out that the performance of sampling is comparable to other state-of-the-art imbalanced strategies, and the method is less restrictive than the cost-sensitive approach [28]. In addition, the fact that sampling is performed as a pre-processing step makes it more flexible than the cost-sensitive approach. Weiss et al. [28] illustrated that by using undersampling methods, time and computational resources required by the learning phase are reduced because less data is handled by the classification algorithm. Undersampling methods outperformed oversampling methods in tasks handling datasets from various domains (e.g. [29], [30]). In addition, undersampling was shown to improve performance in classification tasks using datasets with an imbalanced ratio equal to or more severe than 1:2 [31]. For these reasons, we implemented a progressive undersampling technique to tackle the imbalance class distribution problem that could affect the performance of an automatic approach to support HIV literature screening.

B. Feature Selection

The selection of features is usually performed according to an evaluation metric used to assess the relevance of features. By using feature selection, it is possible to identify a subset of features which are more relevant to a given task, and reduce the size of the feature space in an informed manner. With a smaller and tailored subset of features, the learning phase requires less computational resources, and the classification model reduces the number of noisy or irrelevant attributes. By removing the least discriminative features, the model is also less likely to over-fit the training data. Several feature selection metrics have been proposed in the literature, and evaluated in text classification tasks (e.g. [32], [33]). Among the most popular ones are: Information Gain, Chi-Square test, Term Frequency, Document Frequency, Inverse Document Frequency and Odds Ratio. Comparative studies to evaluate the use of feature selection metrics are not clear about which metric is the most appropriate for text classification problems in general. Therefore, a reasonable choice of feature selection

metric can be made by taking into account the characteristics of the specific classification task. In this work, the Odds Ratio (OR) and Inverse Document Frequency (IDF) were applied as feature selection metrics. Odds Ratio [34] was selected because it evaluates how strongly the occurrence of a feature is associated to a particular document class. The Odds Ratio OR of a term t given a class C can be computed as follows:

$$OR_{(t,C)} = \frac{\frac{n_{Ct}}{n_C} / \frac{n_{C\overline{t}}}{n_C}}{\frac{n_{\overline{C}t}}{n_{\overline{C}}} / \frac{n_{\overline{C}\overline{t}}}{n_{\overline{C}}}}$$

where:

n_{Ct} is the number of times term t appears in class C
n_C is the number of documents in class C
$n_{C\overline{t}}$ is the number of documents in class C without term t
$n_{\overline{C}t}$ is the number of documents with term t, not in class C
$n_{\overline{C}\overline{t}}$ is the number of documents without term t, not in class C
$n_{\overline{C}}$ is the number of documents not in class C

Inverse Document Frequency [35] was selected because it evaluates the specificity of a given feature. Rarer terms yield higher IDF values, indicating that they are more discriminative. The Inverse Document Frequency (idf_t) of a feature t in a document collection is computed as:

$$idf_t = log\frac{N}{df_t}$$

where N is the number of documents in the collection and df_t represents the number of documents that contain term t.

III. Methodology

A. Corpus and Data Sampling

The experiments were conducted on the *SHARE* corpus (http://www.hivevidence.ca). *SHARE* is an easy-to-search and regularly updated repository of synthesized research evidence addressing topics related to HIV/AIDS. *SHARE* includes HIV-relevant systematic reviews and products derived from findings of systematic reviews. Systematic reviews provide a synthesis of individual studies addressing a specific research question. To identify documents to be included in *SHARE*, curators conduct searches in Medline (http://www.nlm.nih.gov/pubs/factsheets/medline.html), Embase (http://www.elsevier.com/solutions/embase), and the Cochrane Library (http://www.cochranelibrary.com). These searches are periodically updated to ensure that the most recent HIV-relevant syntheses are identified. Two reviewers independently assess all records identified through the searches to determine whether they should be included in *SHARE*. During the review process, they include any records that address a topic focused on HIV, and are either a systematic review, an overview of systematic reviews, a policy brief, a treatment guideline, or a systematic review protocol. Currently, the document collection is composed of 18,703 scientific abstracts retrieved from the PubMed database.

The distribution of documents in *SHARE* represents the same ratio of *included* and *excluded* abstracts that scientific researches encounter when performing literature screening for HIV systematic reviews. As the statistics about *SHARE* in Table I show, the class distribution is highly imbalanced with only ≈7% of documents being *included*.

TABLE I
STATISTICS ON *SHARE*

Attribute	Number	%
Total number of documents	18,703	100%
Excluded documents	17,402	93.05%
Included documents	1,301	6.95%
Unique words in abstracts	31,632	-
Unique words in titles	6,821	-
Unique MeSH terms in documents	17,971	-

TABLE II
TRAINING SETS: UNDERSAMPLING APPROACH

Set	Included	%	Excluded	%
1	991	10%	8,915	90%
2	991	20%	3,965	80%
3	991	30%	2,319	70%
4	991	40%	1,487	60%
5	991	50%	991	50%

In order to perform supervised learning, we split the document collection into two parts. The first part contains the documents used as the test set. The test set represents ≈10% of the entire collection, randomly selected to avoid any bias. It contains 1,588 documents (110 were labelled as *included* and 1,478 labelled as *excluded*). The class distribution in the test set is similar to the distribution in the entire document collection. The original distribution of the task is maintained in the test set because our goal is to design a model that will perform best when handling imbalanced data. After isolating the test set documents, five training sets were generated through a random undersampling approach, to progressively discard documents from the majority class. Table II shows the progressive undersampling approach used to generate all training sets, while Figure 1 illustrates the balance of *included*/*excluded* documents in each training set. To perform progressive undersampling in the training sets, we randomly removed documents from the majority class. Our goal was to reach a equal class distribution, and compare the performance of the classification models in order to identify which one is the most appropriate for this task. Several training sets were generated to perform experiments. The training set distribution first starts with the representation of the real imbalanced scenario, i.e. 90% of *excluded* and 10% of *included* documents. Then, gradually, several under-sampling factors are applied to

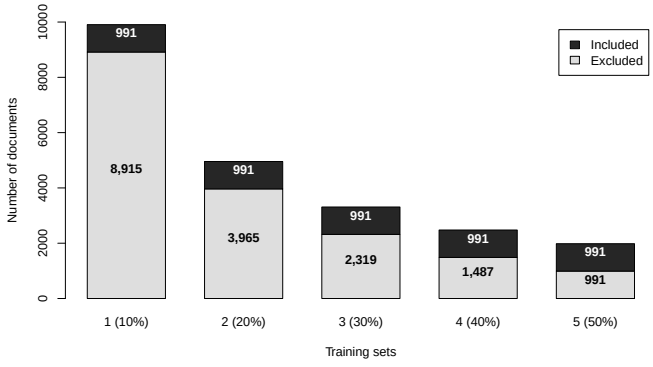

Fig. 1. Progressive undersampling on training sets

the dataset, until a balanced distribution is reached, i.e. 50% of *excluded* and 50% of *included* documents.

B. Feature Extraction and Selection

1) Extraction: To build several classification models and compare their performance, we extracted different types of features, from the baseline Bag-Of-Words (BOW) to MeSH terms [36], and a set of domain keywords identified by researchers working on HIV systematic reviews. The features were mainly extracted from the PubMed XML <AbstractText> and <ArticleTitle> text fields. Each document was represented as a feature vector that account for the number of occurrences of each feature in a given document. A large matrix of documents by features was created and used to feed the classification algorithms. The following feature types were extracted from *SHARE*:
Feature type #1: Bag-Of-Words of the abstract and document title, considering only words with an occurrence of at least 2, and a length of at least 3 characters;
Feature type #2: MeSH terms list, considering only terms with an occurrence of at least 2;
Feature type #3: Domain keywords relevant to HIV systematic reviews. The keywords list is available along with the source code (see Section *"Reproducibility"* at the end of this article).

2) Selection: Since the dataset contains over 18,000 documents, the feature extraction may generate a large and sparse features by documents matrix. In addition to requiring extra computational resources, such a matrix can also interfere with the classifier performance by introducing a bias and overfit the training data. To overcome this, we investigated the use of feature selection before feeding the data to the classification algorithms. We aimed to identify the most suitable feature subset for supporting HIV literature screening by comparing the results obtained when using Odds Ratio and IDF, as described in Section II-B, to filter out the less discriminative features in the classification models.

To perform feature selection using Odds Ratio as a metric, the odds ratio value was computed for each feature extracted from a training set, then a confidence interval for each odds ratio value was computed, using a confidence level of 95%. Two conditions were considered to perform filtering: features with 1) a confidence interval that includes the null hypothesis; or 2) an odds ratio value that is less than or equal to the null hypothesis were discarded, and the remaining features were used to build the models. To perform feature selection using IDF as a metric, we first computed the inverse document frequency of each feature in a given training set considering the occurrence in both *included* and *excluded* classes. Then, similarly to the odds ratio filtering, all features with an IDF value smaller than 1.0 were discarded (this value was experimentally set).

C. Classification Algorithms

In our experiments, we made use of three different classification algorithms: Naïve Bayes (NB), Logistic Model Trees (LMT) and Support Vector Machine (SVM). NB was used as a baseline evaluation of our sampling and feature selection strategies. NB assumes a strong conditional independence of the features. This means that in a feature vector F, the features $f_1, ..., f_n$ are assumed to be conditionally independent given a class C. By this assumption, Naïve Bayes implies that the presence of one word (one feature) is not correlated with the presence or absence of another word, within a class. LMT [37] was previously described by [38] as being able to efficiently handle tasks with imbalanced datasets. It consists of a combination of Decision Tree and LogitBoost algorithms, being a classification tree, with logistic regression models on its nodes. At each node of the decision tree, the LogitBoost algorithm is used to train a data subset for a certain number of iterations, and to define a logistic regression model for the current node. A Decision Tree criterion is then applied to split the current data subset. SVM [39] was also recommended by previous work (e.g. [40], [41]) when dealing with imbalanced data. SVM computes the margin maximum classifier [42], which is the largest radius around a classification boundary, and tries to separate data points on a dimensional space, to identify the different classes to which they belong.

D. Evaluation Metrics

The experimental results were evaluated in terms of precision (P), recall (R), F_1, and F_2. Precision accounts for the number of correct predictions between all correct and incorrect predictions made by the classifier for a specific class. Recall is calculated by the ratio of relevant predictions actually made by the classifier as compared to all existing relevant documents that should have been identified. The $F_{\boldsymbol{\beta}}$ score is the weighted harmonic mean between precision and recall, and is defined as:

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{\beta^2 \times Precision + Recall}$$

where $\boldsymbol{\beta}$ is the relative weight of recall over precision.

In our experiments, we used $\boldsymbol{\beta}$ = 1, leading to the F_1 score. In addition, since we focus on evaluating the model capability of identifying the entire universe of relevant documents, we emphasized recall by also using $\boldsymbol{\beta}$ = 2, leading to the F_2 score.

IV. Experiments and Results

A. Experiments

In total, we generated 105 classification models. These were utilized to analyze the influence of undersampling (different class distributions); the discriminative capability of feature types; and the impact of the feature selection methods. These three aspects were analyzed with the three classification algorithms. The models were then created using various combinations of the settings described hereafter.
Training set balances:
10% included (IN) + 90% excluded (EX) (task distribution);
20% IN + 80% EX;
30% IN + 70% EX;
40% IN + 60% EX;
50% IN + 50% EX.
Feature configurations:
#1 BOW; #2 BOW + MeSH terms; #3 Keywords.

TABLE III
SUMMARY OF EXPERIMENTAL RESULTS USING UNDERSAMPLING

Feature Configuration	Balance	Classifier	P	R	F_1	F_2
#1: Bag-Of-Words	90% - 10%	LMT	0.562	0.664	0.608	0.641
#1: Bag-Of-Words	90% - 10%	NB	0.218	0.855	0.347	0.540
#1: Bag-Of-Words	90% - 10%	SVM	0.733	0.500	0.595	0.534
#1: Bag-Of-Words	80% - 20%	LMT	0.543	0.691	0.608	0.660
#1: Bag-Of-Words	80% - 20%	NB	0.213	0.891	0.344	0.540
#1: Bag-Of-Words	80% - 20%	SVM	0.617	0.673	0.643	0.660
#1: Bag-Of-Words	70% - 30%	LMT	0.481	0.800	0.601	0.706
#1: Bag-Of-Words	70% - 30%	NB	0.213	0.909	0.345	0.550
#1: Bag-Of-Words	**70% - 30%**	**SVM**	**0.540**	**0.800**	**0.645**	**0.730**
#1: Bag-Of-Words	60% - 40%	LMT	0.395	0.891	0.547	0.712
#1: Bag-Of-Words	60% - 40%	NB	0.200	0.864	0.324	0.519
#1: Bag-Of-Words	60% - 40%	SVM	0.473	0.864	0.611	0.741
#1: Bag-Of-Words	50% - 50%	LMT	0.385	0.900	0.540	0.710
#1: Bag-Of-Words	50% - 50%	NB	0.210	0.927	0.342	0.551
#1: Bag-Of-Words	50% - 50%	SVM	0.399	0.900	0.553	0.719
#2: Bag-Of-Words + MeSH	90% - 10%	LMT	0.584	0.664	0.621	0.646
#2: Bag-Of-Words + MeSH	90% - 10%	NB	0.233	0.818	0.363	0.545
#2: Bag-Of-Words + MeSH	90% - 10%	SVM	0.000	0.000	0.000	0.000
#2: Bag-Of-Words + MeSH	80% - 20%	LMT	0.542	0.764	0.634	0.710
#2: Bag-Of-Words + MeSH	80% - 20%	NB	0.233	0.818	0.363	0.540
#2: Bag-Of-Words + MeSH	80% - 20%	SVM	0.000	0.000	0.000	0.000
#2: Bag-Of-Words + MeSH	70% - 30%	LMT	0.481	0.800	0.601	0.706
#2: Bag-Of-Words + MeSH	70% - 30%	NB	0.144	0.918	0.250	0.442
#2: Bag-Of-Words + MeSH	70% - 30%	SVM	1.000	0.009	0.018	0.011
#2: Bag-Of-Words + MeSH	**60% - 40%**	**LMT**	**0.467**	**0.900**	**0.615**	**0.759**
#2: Bag-Of-Words + MeSH	60% - 40%	NB	0.162	0.900	0.275	0.471
#2: Bag-Of-Words + MeSH	60% - 40%	SVM	0.070	0.882	0.129	0.266
#2: Bag-Of-Words + MeSH	50% - 50%	LMT	0.385	0.900	0.540	0.710
#2: Bag-Of-Words + MeSH	50% - 50%	NB	0.126	0.936	0.222	0.410
#2: Bag-Of-Words + MeSH	50% - 50%	SVM	0.069	1.000	0.130	0.270
#3: Keywords	90% - 10%	LMT	0.635	0.491	0.554	0.514
#3: Keywords	90% - 10%	NB	0.304	0.618	0.407	0.512
#3: Keywords	90% - 10%	SVM	0.711	0.291	0.413	0.330
#3: Keywords	80% - 20%	LMT	0.509	0.491	0.500	0.490
#3: Keywords	80% - 20%	NB	0.312	0.664	0.424	0.540
#3: Keywords	80% - 20%	SVM	0.613	0.418	0.497	0.450
#3: Keywords	70% - 30%	LMT	0.462	0.655	0.541	0.604
#3: Keywords	70% - 30%	NB	0.283	0.691	0.401	0.536
#3: Keywords	70% - 30%	SVM	0.516	0.591	0.551	0.574
#3: Keywords	60% - 40%	LMT	0.427	0.691	0.528	0.615
#3: Keywords	60% - 40%	NB	0.288	0.673	0.403	0.531
#3: Keywords	60% - 40%	SVM	0.436	0.655	0.524	0.595
#3: Keywords	50% - 50%	LMT	0.321	0.818	0.462	0.625
#3: Keywords	50% - 50%	NB	0.288	0.700	0.408	0.544
#3: Keywords	50% - 50%	SVM	0.305	0.755	0.435	0.583

Feature selection:
Odds Ratio; Inverse Document Frequency.
Classification algorithms:
Naïve Bayes; Logistic Model Tree; Support Vector Machine.

First, a set of experiments was executed to evaluate the undersampling technique, and therefore the use of various class balances in the training set across the different feature types for the three classifiers. Next, we ran new experiments using the same undersampled training sets and classifiers, but this time applying feature selection to the feature configurations that demonstrated the best performance.

B. Results

Figures 2 and 3 respectively show a summary of the F_1 and F_2 scores obtained in these experiments. Since the focus of our work is to analyze the capability of a model to identify *included* documents, we evaluated the model performance in terms of the results obtained only for the *included* class (the overall performance obtained in the *excluded* class remains generally over 90%). Table III shows the results of the models

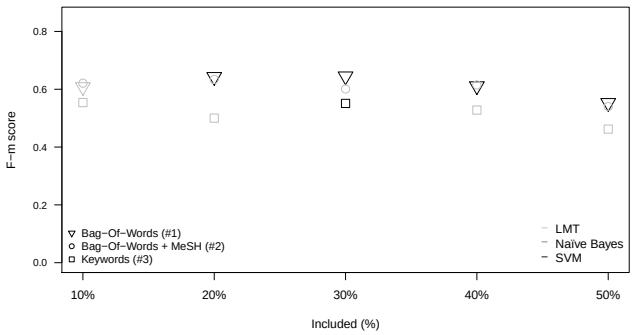

Fig. 2. F_1 results using undersampling

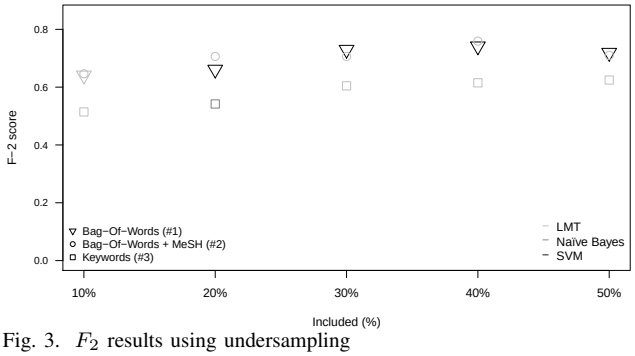

Fig. 3. F_2 results using undersampling

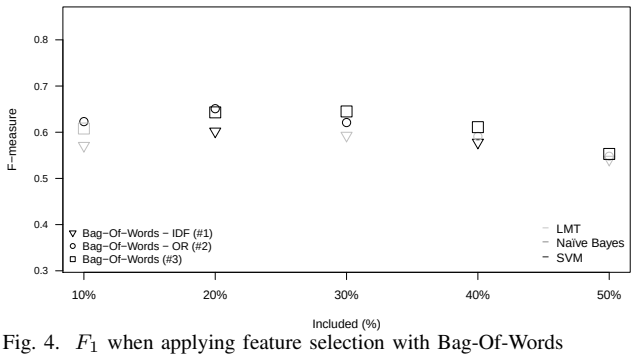

Fig. 4. F_1 when applying feature selection with Bag-Of-Words

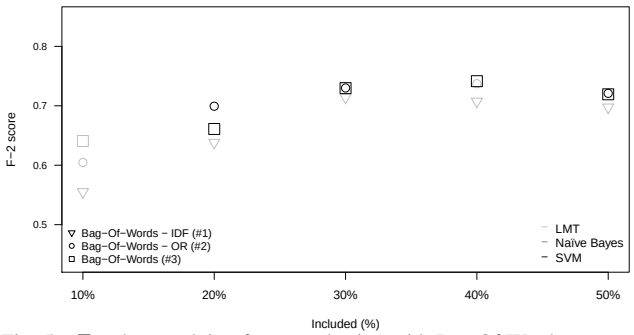

Fig. 5. F_2 when applying feature selection with Bag-Of-Words

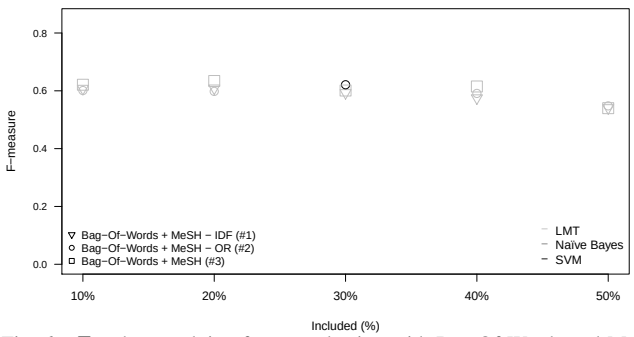

Fig. 6. F_1 when applying feature selection with Bag-Of-Words and MeSH

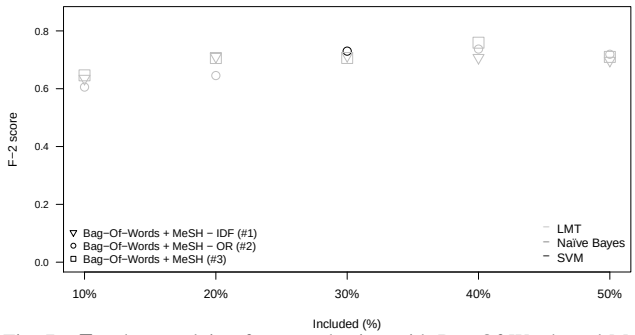

Fig. 7. F_2 when applying feature selection with Bag-Of-Words and MeSH

generated using undersampling, the three feature configurations and three classifiers. The best F_1 (first bold line in Table III) was obtained by a classification model composed of feature configuration #1 (Bag-Of-Words), the SVM classifier, and a training dataset containing 30% of *included* documents. However, we are more interested in the F_2 score because it emphasizes recall over precision, and indicates the capability of a model to identify the greatest number of *included* documents. The F_2 results demonstrate that the best model (second bold line in Table III) is composed of the feature configuration #2 (Bag-Of-Words + MeSH), the LMT classifier and a training set containing 40% of *included* documents. This model achieved 0.467 in precision, 0.9 in recall, 0.615 in F_1 and 0.759 in F_2 score. We call this model *HM1*. Models based on features #2, SVM and 90%-10% or 70%-30% balance show results equal (or very close) to zero because these models classified almost all documents as *excluded*.

As the models with the best F_1 and F_2 were associated to configurations #1 and #2, we applied feature selection to all models that used these configurations. Tables IV and V show the results of these models using IDF and Odds Ratio, respectively. Table VI shows the reduction in the feature space obtained, across the five different training sets. To summarize the effect of the feature selection methods, we show the feature space size for the largest training set (with 10% of *included* documents), the best models (40% of *included* documents), and the smallest training set (with 50% of *included* documents). In general, Odds Ratio reduced the feature space size of configuration #1 by ≈80%, while IDF reduced it by less than 1%. For configuration #2, the reduction by Odds Ratio was over 80%, while IDF reduced it by ≈18%.

As we can observe from the F_2 scores obtained with feature selection, Odds Ratio somewhat outperforms the results obtained with IDF filtering. In general, the performance of configuration #2 still outperforms those of configuration #1. The best model is composed of the feature configuration #2 (Bag-Of-Words + MeSH terms), the LMT classifier, a training set with 40% of *included* documents, and filtering by Odds Ratio. This model (in bold in Table V) achieved 0.445 in precision, 0.882 in recall, 0.591 in F_1 and 0.737 in F_2 score. We call this model *HM2*. Although *HM2* did not outperform *HM1* (in which no filtering was applied), *HM2* has very similar results to *HM1*. The major difference between the two is that *HM1* has a feature space size of 14,459; while the feature space size of *HM2* is 2,411. By having a more concise feature space, *HM2* requires less computational resources and time for the learning phase. Thus, *HM2* can be a suitable choice when resources are limited.

TABLE IV
SUMMARY OF EXPERIMENT RESULTS USING IDF FOR FEATURE SELECTION

Feature Configuration	Balance	Classifier	P	R	$\mathbf{F_1}$	$\mathbf{F_2}$
#1: Bag-Of-Words	90% - 10%	LMT	0.600	0.545	0.571	0.555
#1: Bag-Of-Words	90% - 10%	NB	0.214	0.845	0.342	0.532
#1: Bag-Of-Words	90% - 10%	SVM	0.688	0.400	0.506	0.437
#1: Bag-Of-Words	80% - 20%	LMT	0.529	0.673	0.592	0.64
#1: Bag-Of-Words	80% - 20%	NB	0.209	0.891	0.339	0.54
#1: Bag-Of-Words	80% - 20%	SVM	0.646	0.564	0.602	0.58
#1: Bag-Of-Words	70% - 30%	LMT	0.462	0.827	0.593	0.714
#1: Bag-Of-Words	70% - 30%	NB	0.209	0.909	0.340	0.544
#1: Bag-Of-Words	70% - 30%	SVM	0.518	0.655	0.578	0.622
#1: Bag-Of-Words	60% - 40%	LMT	0.438	0.836	0.575	0.707
#1: Bag-Of-Words	60% - 40%	NB	0.195	0.864	0.318	0.512
#1: Bag-Of-Words	60% - 40%	SVM	0.479	0.727	0.578	0.659
#1: Bag-Of-Words	50% - 50%	LMT	0.394	0.864	0.541	0.698
#1: Bag-Of-Words	50% - 50%	NB	0.199	0.927	0.328	0.535
#1: Bag-Of-Words	50% - 50%	SVM	0.386	0.800	0.521	0.659
#2: Bag-Of-Words + MeSH	90% - 10%	LMT	0.567	0.655	0.608	0.635
#2: Bag-Of-Words + MeSH	90% - 10%	NB	0.217	0.845	0.345	0.535
#2: Bag-Of-Words + MeSH	90% - 10%	SVM	0.688	0.400	0.506	0.437
#2: Bag-Of-Words + MeSH	80% - 20%	LMT	0.492	0.800	0.609	0.71
#2: Bag-Of-Words + MeSH	80% - 20%	NB	0.21	0.891	0.34	0.54
#2: Bag-Of-Words + MeSH	80% - 20%	SVM	0.656	0.555	0.601	0.57
#2: Bag-Of-Words + MeSH	70% - 30%	LMT	0.462	0.827	0.593	0.714
#2: Bag-Of-Words + MeSH	70% - 30%	NB	0.159	0.909	0.271	0.468
#2: Bag-Of-Words + MeSH	70% - 30%	SVM	0.526	0.645	0.58	0.612
#2: Bag-Of-Words + MeSH	60% - 40%	LMT	0.438	0.836	0.575	0.707
#2: Bag-Of-Words + MeSH	60% - 40%	NB	0.159	0.882	0.269	0.462
#2: Bag-Of-Words + MeSH	60% - 40%	SVM	0.462	0.727	0.565	0.652
#2: Bag-Of-Words + MeSH	50% - 50%	LMT	0.394	0.864	0.541	0.698
#2: Bag-Of-Words + MeSH	50% - 50%	NB	0.173	0.927	0.291	0.495
#2: Bag-Of-Words + MeSH	50% - 50%	SVM	0.350	0.845	0.495	0.659

The F_1 and F_2 results of feature selection methods applied to feature configuration #1 are presented in Figures 4 and 5, while the results obtained by feature selection applied to feature configuration #2 are presented in Figures 6 and 7. In both configurations we present a comparison between the models using no feature selection, the models in which IDF was applied, and the models in which Odds Ratio was applied.

V. DISCUSSION

The best models identified during our experiments, *HM1* and *HM2*, both made use of the LMT classifier and the feature configuration composed of the Bag-Of-Words and MeSH terms, using a training set with 40% of *included* documents. We discuss here our observations on these three parameters.

A. Imbalanced data

As demonstrated by [6] on biomedical literature classification, results obtained with a more balanced training corpus outperform the models based on training corpora that have similar distributions to the original task of literature screening. Among all different class distributions, the balance that yielded better results contained 40% of *included* and 60% of *excluded* documents. This distribution allows the more balanced model to still maintain the underlying characteristic of the data, while removing extra noise that would be introduced by additional *excluded* documents. We observed that models with such

TABLE V
SUMMARY OF EXPERIMENT RESULTS USING ODDS RATIO FOR FEATURE SELECTION

Feature Configuration	Balance	Classifier	P	R	F_1	F_2
#1: Bag-Of-Words	90% - 10%	LMT	0.588	0.609	0.598	0.605
#1: Bag-Of-Words	90% - 10%	NB	0.228	0.864	0.361	0.555
#1: Bag-Of-Words	90% - 10%	SVM	0.697	0.564	0.623	0.586
#1: Bag-Of-Words	80% - 20%	LMT	0.532	0.682	0.598	0.650
#1: Bag-Of-Words	80% - 20%	NB	0.222	0.873	0.354	0.550
#1: Bag-Of-Words	80% - 20%	SVM	0.583	0.736	0.651	0.700
#1: Bag-Of-Words	70% - 30%	LMT	0.481	0.800	0.601	0.706
#1: Bag-Of-Words	70% - 30%	NB	0.220	0.900	0.353	0.556
#1: Bag-Of-Words	70% - 30%	SVM	0.497	0.827	0.621	0.730
#1: Bag-Of-Words	60% - 40%	LMT	0.445	0.882	0.591	0.737
#1: Bag-Of-Words	60% - 40%	NB	0.213	0.873	0.343	0.539
#1: Bag-Of-Words	60% - 40%	SVM	0.430	0.873	0.577	0.724
#1: Bag-Of-Words	50% - 50%	LMT	0.392	0.909	0.548	0.719
#1: Bag-Of-Words	50% - 50%	NB	0.212	0.882	0.342	0.540
#1: Bag-Of-Words	50% - 50%	SVM	0.388	0.918	0.546	0.721
#2: Bag-Of-Words + MeSH	90% - 10%	LMT	0.593	0.609	0.601	0.606
#2: Bag-Of-Words + MeSH	90% - 10%	LMT	0.228	0.864	0.361	0.555
#2: Bag-Of-Words + MeSH	90% - 10%	LMT	0.755	0.336	0.465	0.378
#2: Bag-Of-Words + MeSH	80% - 20%	LMT	0.532	0.682	0.598	0.650
#2: Bag-Of-Words + MeSH	80% - 20%	LMT	0.221	0.873	0.352	0.550
#2: Bag-Of-Words + MeSH	80% - 20%	LMT	0.085	0.800	0.153	0.300
#2: Bag-Of-Words + MeSH	70% - 30%	LMT	0.481	0.800	0.601	0.706
#2: Bag-Of-Words + MeSH	70% - 30%	LMT	0.220	0.900	0.353	0.556
#2: Bag-Of-Words + MeSH	70% - 30%	LMT	0.497	0.827	0.621	0.730
#2: Bag-Of-Words + MeSH	**60% - 40%**	**LMT**	**0.445**	**0.882**	**0.591**	**0.737**
#2: Bag-Of-Words + MeSH	60% - 40%	LMT	0.213	0.873	0.342	0.539
#2: Bag-Of-Words + MeSH	60% - 40%	LMT	0.157	0.927	0.269	0.468
#2: Bag-Of-Words + MeSH	50% - 50%	LMT	0.392	0.909	0.548	0.719
#2: Bag-Of-Words + MeSH	50% - 50%	LMT	0.212	0.882	0.342	0.540
#2: Bag-Of-Words + MeSH	50% - 50%	LMT	0.384	0.900	0.538	0.709

TABLE VI
FEATURE SPACE SIZE REDUCTION AFTER FILTERING BY ODDS RATIO AND IDF

Configuration	*Included* (%)	# of features	IDF	(%)	OR	(%)
#1: Bag-Of-Words	50%	9,913	9,826	0.88%	2,042	79.40%
#1: Bag-Of-Words	40%	11,183	11,092	0.81%	2,392	78.61%
#1: Bag-Of-Words	30%	12,935	12,844	0.70%	2,828	78.14%
#1: Bag-Of-Words	20%	15,709	15,616	0.59%	3,229	79.44%
#1: Bag-Of-Words	10%	22,060	21,944	0.53%	4,040	81.69%
#2: Bag-Of-Words + MeSH	50%	12,688	10,511	17.16%	2,047	83.87%
#2: Bag-Of-Words + MeSH	40%	14,459	11,869	17.91%	2,411	83.33%
#2: Bag-Of-Words + MeSH	30%	16,794	13,766	18.03%	2,828	83.16%
#2: Bag-Of-Words + MeSH	20%	20,350	16,650	18.18%	3,514	82.73%
#2: Bag-Of-Words + MeSH	10%	28,506	23,223	18.53%	4,061	85.75%

balance can better classify documents on the test set composed of the same class distribution as the original task (10%-90%).

B. Feature configurations

As we can observe in Figures 2 and 3, the configuration containing only keywords is less discriminative compared to Bag-Of-Words and MeSH terms. We attribute this result to the size of the feature set. Feature configuration #1 has 9,913 to 22,060 features (both considering the most balanced, and the largest and most imbalanced training set, respectively) and feature configuration #2 has 12,688 to 28,506 features. On the other hand, configuration #3 (the keywords) contains a fixed set of 573 features, therefore ≈95% smaller than the smallest feature sets extracted by the other configurations. Configuration #2 generally demonstrated the best performance, and can be recommended as the most suitable feature set for this task. It is a combination of Bag-Of-Words and MeSH terms, resulting in a higher number of features, and therefore providing more information to build the decision boundary.

C. Feature selection

The models using IDF and Odds Ratio as feature selection achieved comparable results. However, the reduction in the feature space size provided by Odds Ratio is significant, while maintaining similar performance to the models with no feature selection. By using this selection method, the features that are kept are the most discriminating of *included* documents. This approach contributes to generate a feature subset that is better tailored to recognize the most relevant documents, while removing attributes that are not discriminative for these documents.

VI. CONCLUSION

We developed a supervised learning method to support the HIV literature screening. Similarly to other classification tasks using biomedical data, the *SHARE* dataset used in our work presented an imbalanced class distribution. Only a small proportion of the document collection represented the task target. Since this negatively affects the performance of classification algorithms, data undersampling and feature selection were analyzed. After experimenting with 105 classification models, we identified the two best models that seem to best support HIV literature screening. The first model, which we call *HM1*, is composed of a training set containing 40% of *included* and 60% of *excluded* documents, and uses a Bag-Of-Words and MeSH terms as features. *HM1* reached a recall of 0.9 for the *included* class, which indicates that 90% of the *included* documents were correctly classified. After applying feature selection, the best performing model, which we call *HM2* yielded a recall of 0.88 for the *included* class. *HM2* has a similar composition as *HM1*, but the set of features was filtered using Odds Ratio. While *HM2* achieved similar results to *HM1*, the set of features in *HM2* is ≈83% smaller than in *HM1*, which makes it a better model when computational resources is a concern.

The use of an automatic approach to support literature screening can greatly benefit experts working in HIV systematic reviews. Our results indicate that, by utilizing classification models, the great majority of documents to be potentially *included* in reviews by researchers can be precisely labelled. With the support of our system, experts can expect a decrease in the amount of time and effort needed to collect HIV systematic reviews.

Reproducibility. Our prototype can be re-used to support different literature screening tasks beyond the one described here. The prototype was implemented in Java and is composed of several modules that allow the use of other datasets, other undersampling methods, other features and other feature selection methods. The software prototype is available under the MIT License at https://github.com/TsangLab.

ACKNOWLEDGMENT

The authors acknowledge the contributions of the Ontario HIV Treatment Network (OHTN) and McMaster University, as well as the developers of *SHARE*. Part of this work was funded by MITACS, eHealth in Motion Ltd. and Dataparc.

REFERENCES

[1] E. W. Sayers, T. Barrett, D. A. Benson, E. Bolton, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, S. Federhen *et al.*, "Database Resources of the National Center for Biotechnology Information," *Nucleic acids research*, vol. 39, no. suppl 1, pp. D38–D51, 2011.

[2] T. B. Murdoch and A. S. Detsky, "The Inevitable Application of Big Data to Health Care," *JAMA*, vol. 309, no. 13, pp. 1351–1352, 2013.

[3] J. P. Higgins, S. Green *et al.*, *Cochrane Handbook for Systematic Reviews of Interventions*. Wiley Online Library, 2008, vol. 5.

[4] U. S. Mudunuri, M. Khouja, S. Repetski, G. Venkataraman, A. Che, B. T. Luke, F. P. Girard, and R. M. Stephens, "Knowledge and Theme Discovery across Very Large Biological Data Sets Using Distributed Queries: A Prototype Combining Unstructured and Structured Data," *PLOS ONE*, vol. 8, no. 12, p. e80503, 2013.

[5] C. Quan, M. Wang, and F. Ren, "An Unsupervised Text Mining Method for Relation Extraction from Biomedical Literature," *PLOS ONE*, vol. 9, no. 7, p. e102039, 2014.

[6] H. Almeida, M.-J. Meurs, L. Kosseim, G. Butler, and A. Tsang, "Machine Learning for Biomedical Literature Triage," *PLOS ONE*, vol. 9, no. 12, p. e115892, 2014.

[7] G. Tsafnat, P. Glasziou, M. K. Choong, A. Dunn, F. Galgani, and E. Coiera, "Systematic Review Automation Technologies," *BMC Systematic Reviews*, vol. 3, no. 1, p. 74, 2014.

[8] O. Alison, J. Thomas, J. McNaught, M. Miwa, S. Ananiadou *et al.*, "Using Text Mining for Study Identification in Systematic Reviews: A Systematic Review of Current Approaches," *Systematic Reviews*, vol. 4, no. 1, p. 5, 2015.

[9] T. Bekhuis and D. Demner-Fushman, "Screening Nonrandomized Studies for Medical Systematic Reviews: A Comparative Study of Classifiers," *Artificial intelligence in medicine*, vol. 55, no. 3, pp. 197–207, 2012.

[10] M. Wang, W. Zhang, W. Ding, D. Dai, H. Zhang, H. Xie, L. Chen, Y. Guo, and J. Xie, "Parallel Clustering Algorithm for Large-Scale Biological Data Sets," *PLOS ONE*, vol. 9, no. 4, p. e91315, 2014.

[11] B. C. Wallace, T. A. Trikalinos, J. Lau, C. Brodley, and C. H. Schmid, "Semi-automated Screening of Biomedical Citations for Systematic Reviews," *BMC bioinformatics*, vol. 11, no. 1, p. 55, 2010.

[12] C. N. Arighi, C. H. Wu, K. B. Cohen, L. Hirschman, M. Krallinger, A. Valencia, Z. Lu, J. W. Wilbur, and T. C. Wiegers, "BioCreative-IV Virtual Issue," *Database*, vol. 2014, p. bau039, 2014.

[13] F. Leitner, M. Krallinger, C. Rodriguez-Penagos, J. Hakenberg, C. Plake, C.-J. Kuo, C.-N. Hsu, R. Tsai, H.-C. Hung, W. W. Lau *et al.*, "Introducing Meta-services for Biomedical Information Extraction," *Genome Biol*, vol. 9, no. Suppl 2, p. S6, 2008.

[14] L. Hirschman, G. A. C. Burns, M. Krallinger, C. Arighi, K. B. Cohen, A. Valencia, C. H. Wu, A. Chatr-Aryamontri, K. G. Dowell, E. Huala *et al.*, "Text Mining for the Biocuration Workflow," *Database*, vol. 2012, p. bas020, 2012.

[15] L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia, "Overview of BioCreative: Critical Assessment of Information Extraction for Biology," *BMC bioinformatics*, vol. 6, no. Suppl 1, p. S1, 2005.

[16] S. Matis-Mitchell, P. Roberts, C. O. Tudor, and C. Arighi IV, "BioCreative IV interactive task," *BioCreative IV Proceedings*, vol. 1, 2013.

[17] C. N. Arighi, B. Carterette, K. B. Cohen, M. Krallinger, W. J. Wilbur, P. Fey, R. Dodson, L. Cooper, C. E. Van Slyke, W. Dahdul *et al.*, "An overview of the BioCreative 2012 Workshop Track III: Interactive text mining task," *Database*, vol. 2013, p. bas056, 2013.

[18] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[19] G. M. Weiss and F. Provost, "The Effect of Class Distribution on Classifier Learning: An Empirical Study," *Technical Report ML-TR-44, August 2, Department of Computer Science, Rutgers University*, 2001.

[20] Y. Liu, N. V. Chawla, M. P. Harper, E. Shriberg, and A. Stolcke, "A Study in Machine Learning from Imbalanced Data for Sentence Boundary Detection in Speech," *Computer Speech & Language*, vol. 20, no. 4, pp. 468–494, 2006.

[21] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler, "Learning from Imbalanced Data in Surveillance of Nosocomial Infection," *Artificial Intelligence in Medicine*, vol. 37, no. 1, pp. 7–18, 2006.

[22] R. J. Bolton and D. J. Hand, "Statistical Fraud Detection: A Review," *Statistical Science*, pp. 235–249, 2002.

[23] P. Soda, "A Multi-objective Optimisation Approach for Class Imbalance Learning," *Pattern Recognition*, vol. 44, no. 8, pp. 1801–1810, 2011.

[24] N. Garca-Pedrajas, J. Prez-Rodrguez, M. Garca-Pedrajas, D. Ortiz-Boyer, and Colin, "Class Imbalance Methods for Translation Initiation Site Recognition in DNA Sequences," *Knowledge-Based Systems*, vol. 25, no. 1, pp. 22 – 34, 2012, special Issue on New Trends in Data Mining.

[25] M. A. Maloof, "Learning when Data Sets Are Imbalanced and when Costs Are Unequal and Unknown," in *ICML-2003 workshop on learning from imbalanced data sets II, Washington DC*, vol. 2, 2003.

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling TEchnique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 341–378, 2002.

[27] L. Borrajo, R. Romero, E. L. Iglesias, and C. R. Marey, "Improving Imbalanced Scientific Text Classification Using Sampling Strategies and Dictionaries," *Journal of integrative bioinformatics*, vol. 8, p. 176, 2011.

[28] G. M. Weiss, K. McCarthy, and B. Zabar, "Cost-sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs?" in *DMIN-International Conference on Data Mining*, 2007, pp. 35–41.

[29] C. Drummond and R. C. Holte, "C4.5, Class Imbalance, and Cost Sensitivity: Why Under-sampling beats Over-sampling," in *Workshop on Learning from Imbalanced Datasets II at the International Conference on Machine Learning (ICML-2003), Washington DC*, 2003, pp. 1–8.

[30] J. Luengo, A. Fernández, S. García, and F. Herrera, "Addressing Data Complexity for Imbalanced Data Sets: Analysis of SMOTE-based Oversampling and Evolutionary Undersampling," *Soft Computing*, vol. 15, no. 10, pp. 1909–1936, 2011.

[31] O. Loyola-González, M. García-Borroto, M. A. Medina-Pérez, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and G. De Ita, "An Empirical Study of Oversampling and Undersampling Methods for LCMine an Emerging Pattern Based Classifier," in *Pattern Recognition*. Springer, 2013, pp. 264–273.

[32] T. A. Almeida, J. Almeida, and A. Yamakami, "Spam Filtering: How the Dimensionality Reduction Affects the Accuracy of Naive Bayes Classifiers," *Journal of Internet Services and Applications*, vol. 1, no. 3, pp. 183–200, 2011.

[33] T. Basu and C. Murthy, "Effective Text Classification by a Supervised Feature Selection Approach," in *Proceedings of the IEEE 12th International Conference on Data Mining Workshops (ICDMW), December 10, Brussels, Belgium*. IEEE, 2012, pp. 918–925.

[34] M. Szumilas, "Explaining odds ratios," *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, vol. 19, no. 3, p. 227, 2010.

[35] K. Sparck Jones, "A Statistical Interpretation of Term Specificity and its Application in Retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[36] C. E. Lipscomb, "Medical Subject Headings (MeSH)," *Bulletin of the Medical Library Association*, vol. 88, no. 3, p. 265, 2000.

[37] N. Landwehr, M. Hall, and E. Frank, "Logistic Model Trees," *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.

[38] E. Charton, M.-J. Meurs, L. Jean-Louis, and M. Gagnon, "Using Collaborative Tagging for Text Classification," *Informatics 2014*, pp. 32–51, 2013.

[39] V. N. Vapnik, "The Nature of Statistical Learning Theory," 1995.

[40] R. Akbani, S. Kwek, and N. Japkowicz, "Applying Support Vector Machines to Imbalanced Datasets," in *Machine Learning: ECML 2004*. Springer, 2004, pp. 39–50.

[41] A. Mountassir, H. Benbrahim, and I. Berrada, "An Empirical Study to Address the Problem of Unbalanced Data Sets in Sentiment Classification," *IEEE Systems, Man, Cybernetics*, pp. 3298–3303, 2012.

[42] S. Marsland, *Machine Learning: An Algorithm Perspective*, 1st ed. Chapman and Hall, 2009.