International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 1, February 2024, pp. 589~599
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i1.pp589-599

Text classification supervised algorithms with term frequency inverse document frequency and global vectors for word representation: a comparative study

Zakia Labd[1], Said Bahassine[2], Khalid Housni[1], Fatima Zahrae Ait Hamou Aadi[1], Khalid Benabbes[1]
[1]Laboratory of Research in Informatics L@RI, Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco
[2]Laboratory of Artificial Intelligence and Complex Systems Engineering, Department of Computer Science, National Higher School of Arts and Crafts, Hassan II University, Casablanca, Morocco

Article Info

Article history:

Received Sep 15, 2022
Revised Jul 6, 2023
Accepted Aug 7, 2023

Keywords:

Decision tries
Document classification
Global vectors
K-nearest neighbors
Natural language processing
Support vector machine
Survey

ABSTRACT

Over the course of the previous two decades, there has been a rise in the quantity of text documents stored digitally. The ability to organize and categorize those documents in an automated mechanism, is known as text categorization which is used to classify them into a set of predefined categories so they may be preserved and sorted more efficiently. Identifying appropriate structures, architectures, and methods for text classification presents a challenge for researchers. This is due to the significant impact this concept has on content management, contextual search, opinion mining, product review analysis, spam filtering, and text sentiment mining. This study analyzes the generic categorization strategy and examines supervised machine learning approaches and their ability to comprehend complex models and nonlinear data interactions. Among these methods are k-nearest neighbors (KNN), support vector machine (SVM), and ensemble learning algorithms employing various evaluation techniques. Thereafter, an evaluation is conducted on the constraints of every technique and how they can be applied to real-life situations.

Corresponding Author:

Zakia Labd
Laboratory of Research in Informatics L@RI, Department of Computer Science, Faculty of Sciences, Ibn Tofail University
Kenitra, Morocco
Email: zakia.labd@uit.ac.ma

1. INTRODUCTION

In numerous real-world applications, text classification challenges have been extensively investigated during the past few decades. Recent advances in natural language processing and text mining have piqued the interest of numerous researchers in the creation of applications that utilize text categorization algorithms. These advancements have not only enhanced the accuracy of text classification, but also expanded it is scope. Text classification models have produced impressive results in tasks such as sentiment analysis, machine translation, and document summarization by combining deep learning approaches and word embeddings such as global vectors for word representation (GloVe). As a result, the opportunities for leveraging text classification continue to grow, promising enhanced automation and information retrieval across a wide range of domains.

Classification of documents is a problem involving the construction of models that can categorize documents into predetermined categories. It is a complicated process that comprises training models, data

processing, transformation, and reduction. This remains a noteworthy research area, utilizing numerous strategies and their sophisticated algorithmic combinations. An initial classification of documents into distinct categories simplifies numerous document processing processes and improves the overall performance of document processing systems. The bulk of document classification algorithms now use text content or document structure to classify documents such as insurance papers, letters, and essays. This work addresses document classification challenges by considering the content of the document rather than the structure.

Selecting the optimal classifier is the most crucial step in the classification of text. We cannot choose the most effective model for a text categorization application until we have a thorough conceptual understanding of each approach. In the next section, the most common supervised text categorization approaches are discussed. First, we will cover non-parametric algorithms that have been explored and applied for classification problems, such as k-nearest neighbor (KNN) [1]. Support vector machine (SVM) [2], [3] is another well-known technique for document categorization that employs a discriminative classifier. This technique has been widely implemented in numerous data mining domains, including image and video processing, among others. In addition, researchers frequently utilize SVM as a benchmark to evaluate the efficacy of their proposed models and to demonstrate their original contributions.

Document classification has also been researched using tree-based classifiers such as decision tree (DT) and random forest (RF) [4]. Each of these tree-based algorithms will receive its own segment of discussion. The majority of these methods are applied for document summarization [5] and automated keyword extraction [6]. The purpose of this research is to conduct a comparative analysis of the efficiency and efficacy of various document classification strategies. Even though there are numerous comparison studies and experiments for document categorization, their tests are sometimes "incomplete," as their conclusions are inconsistent due to the use of diverse data sets. We explore the effectiveness, efficiency, and scalability of several document classification techniques.

The paper is structured as follows: in section 2, an overview of feature extraction and classification techniques is presented. Section 3 examines the main issues in text classification and provides a survey of current solutions. Section 4 outlines the generic strategy utilized in the survey, offering insights into the methodologies employed. Section 5, delves into the experimental phase and presents an evaluation of the utilized methods and approaches, discussing their effectiveness and performance. Finally, in section 6, the paper provides a comprehensive summary of the main points discussed throughout the study.

2. RELATED WORK

2.1. Feature extraction

Although the term "word embedding" has gained popularity because of the development of neural network techniques, the first attempts to create distributed representations were made in the context-counting field. The co-occurrence matrix must be manually allocated in memory, which is the main disadvantage of context-counting methods. Random indexing [7], [8] was proposed to address this limitation by creating nearly orthogonal random indexes for words and then iteratively removing the factorization. When dealing with large amounts of text data, however, neural methods such as word2vec and GloVe have proven to be more effective than rule-based inference. GloVe, a well-known embedding method, has been shown to outperform word2vec in a variety of tasks [9]. GloVe can learn word vectors that can be used to reconstruct the likelihood of co-occurrence between phrases based on their dot product. Both word2vec and GloVe have been used to create massive collections of embeddings that are publicly available.

Table 1 provides a comparison of three text representation models: term frequency-inverse document frequency (TF-IDF), Word2Vec, and GloVe (pre-trained). Although TF-IDF is simple to compute and use for document similarity, it lacks semantic understanding and can be slow with big vocabularies. Word2Vec can extract word order and semantics but not in-text word meaning or out-of-vocabulary phrases. GloVe (pre-trained) outperforms Word2Vec in terms of capturing word locations and meanings.

2.2. Classification techniques

Boser *et al.* [10] created supervised learning methods applicable to classification or regression, including the SVM. SVM was originally developed for binary classification but may be extended to higher-dimensional nonlinear situations [11], [12] and is based on structural risk reduction. An SVM-based method is presented in [13] that improves the performance of the SVM classifier by incremental learning, harmful unlearning, and boosting. Boosted SVM works particularly well on high-dimensional datasets, while other approaches have improved SVM performance by enhancing vectorization algorithms. The augmented naive Bayes vectorization algorithm outperforms the TF-IDF classifier, according to a study [14], [15]. Laplace smoothing improves naive Bayes-SVM classification performance beyond that of TF-IDF [15], hence the suggested approach for categorizing texts is very effective and accurate.

Table 1. Comparison of feature extraction methods

Model	Advantages	Limitation
TF-IDF	– Easily computed – Easy to use to calculate a similarity of two documents – Basic metrics for extracting the most descriptive terms	– The meaning between words (semantics) in the text is not included – It fails to grasp the significance of the text (semantics) – It calculates document resemblance directly in space, which can be slow for big vocabulary
Word2Vec	– It depicts the order in which the words appear in the text (syntactic) – It assesses the meaning of the words (semantics)	– It is unable to extract the meaning of a word from the body text – It is unable to extract out-of-vocabulary words from the corpus
Glove (Pre-Trained)	– It captures the position of the words in the text – It captures meaning in the words (semantics) – Trained on enormous corpora	– Unable to extract the meaning of a word from the body text – Memory consumption for storage – Unable to extract out-of-vocabulary words from the corpus

2.2.1. K-nearst neibghors (KNN)

K-NN is an efficient similarity-based learning algorithm for categorizing documents. It identifies the k nearest neighbors of a test document in the training set and assesses class candidates according to their classes. Iswarya and Radha [16] suggested an Ensemble learning strategy for the Improved KNN method for text categorization (EINNTC), which use one-pass clustering to reduce similarity calculation time and minimize noisy samples. In the first stage, a classification model is developed and updated, and in the second step, ensemble learning is used to determine the ideal value for the parameter K. In terms of F1 score, the results demonstrate that EINNTC surpasses SVM and conventional KNN.

2.2.2. Decision trees (DTs)

Decision trees are regarded as one of the most practical and simple approaches to classification. This technique is built through a hierarchical decomposition of the data space. D. Morgan proposed and J. R. Quinlan developed the decision tree as a classification task. The main concept is to create a tree of categorized data points based on the attribute. The classifier is a tree with internal nodes representing features, branches deviating from them representing a decision rule, and leaves and leaf nodes representing the outcome labels. A decision tree classifies a test document by recursively evaluating the labelling weights of internal nodes in the document vector until a leaf is reached. The primary problem, however, is defining which properties or characteristics belong at the parent level and which belong at the child level. The main properties are achieved by applying a metric known as Information Gain.

2.2.3. Random forests (RFs)

Random forests (RFs) are a type of tree predictor created by T. Kam Ho in 1995 as an ensemble learning method for text classification. In 2001, Breiman's description of random forests gained attention, influenced by Amit and Geman's similar "random trees" methods. Random forests are widely used due to their high predictive accuracy and have been successfully applied in various fields [17]–[22]. In 2018, a new variation called LazyNN RF was proposed for high-dimensional noisy classification applications. The model improves on typical random forests by using a "localized" training projection that filters out unnecessary data, avoiding overfitting caused by overly complex trees. LazyNN RF outperformed state-of-the-art classifiers in almost all reference datasets tested, demonstrating it is effectiveness and feasibility as a strategy [22].

2.2.4. Classification techniques comparison

In the context of large-scale search problems, as illustrated in the Table 2, the effectiveness of the KNN algorithm is constrained by data storage limitations. Moreover, the efficacy of KNN is highly dependent on the definition of a meaningful distance function, making it a highly data-dependent algorithm, as demonstrated by previous research [23], [24]. These observations highlight the critical considerations associated with the practical application of KNN in scenarios where storage resources and the definition of pertinent distance metrics play a pivotal role in determining the algorithm's success.

Since its introduction in the 1990s, SVM has been one of the most effective machine learning algorithms. However, they are hindered by the lack of transparency in their conclusions, which is a result of the numerous dimensions. Consequently, the company score cannot be displayed as a parametric function based on financial indicators or in any other functional form [25]. A variable financial ratio rate is a further limitation [26]. The decision tree is a rapid method for both learning and prediction, but it is particularly sensitive to small data changes and is easily overfit [27]. Prediction outside of the sample is also a difficulty with this method. Compared to other systems, random forests are extremely quick to train, but once trained, they are slow at making predictions [28]. SVM classifier gave the better results in terms of precision, recall and f-measure compared to DT [29], [30].

Table 2. Comparison of text categorization algorithms (SVM, KNN, DT, and RF)

Class/approach	Algorithms	Advantages	Disadvantages
Supervised learning	Support vector machine (SVM)	– SVM is capable of handling nonlinear decision boundaries – Robust against over fitting issues. – Can work with large size data	– Large number of dimensions – Difficulty in picking an efficient kernel function – Time and memory complexity is high
	K-nearest neighbor (KNN)	– Effectiveness in text classification – Non-parametric – Handles multi-class data sets	– Computationally expensive – Difficulties finding an optimal k value – Challenging to find a meaningful distance function
	Decision tree (DT)	– Handles categorical features easily – Divides hierarchically the data and works well with decision margins parallel to the feature axis – Fast in learning and prediction	– Overfit – Sensitive to perturbations in the data set – The noise handling is bad
Ensemble learning	Random forest (RF)	– With decision tree ensembles, training time is reduced compared to other approaches – There is less variance in trees – The input data does not need to be prepared or pre-processed	– Slow predictions – Large number of trees increases the difficulty of the prediction stage – Visually, it is not as straightforward – Overfitting is a common problem – Choosing the right number of trees for a forest is necessary

3. STATE OF THE ART TECHNIQUES

Table 3 (see in appendix) summarizes key aspects, including the used method, review element, key contribution, and corpus utilized by each methodology of four research articles addressing text classification techniques. The first article introduces a boosted SVM classifier using incremental learning and detrimental unlearning to address challenges related to SVM convergence and memory consumption in high-dimensional datasets. The second article discusses multi-class document classification using support vector machine based on an improved naïve Bayes vectorization technique, aiming to reduce the dimensionality of data while enhancing vectorization methods. The third article presents adaptive random forests for evolving data streams, proposing a technique that adapts random forests for dynamic data stream learning. The final article introduces a LazyNN RF classifier designed for high-dimensional noisy classification tasks and demonstrates its superior performance compared to state-of-the-art classifiers in various reference datasets. Each article contributes unique approaches to addressing specific challenges in text classification, and they utilize different datasets to validate their methods.

4. METHODOLOGY OF STUDY

We intend to provide an overview of text classification techniques in this article, along with an explanation of the relevant pre-processing processes and evaluation methods, following the workflow in Figure 1. First, we will begin with text preparation and go over the various techniques available, followed by a review of text representation, which is typically the most difficult issue in building a classifier. Phase 2 presents the document presentation and in the last part we review and evaluate the different methods of classification in the 4 different corpuses.

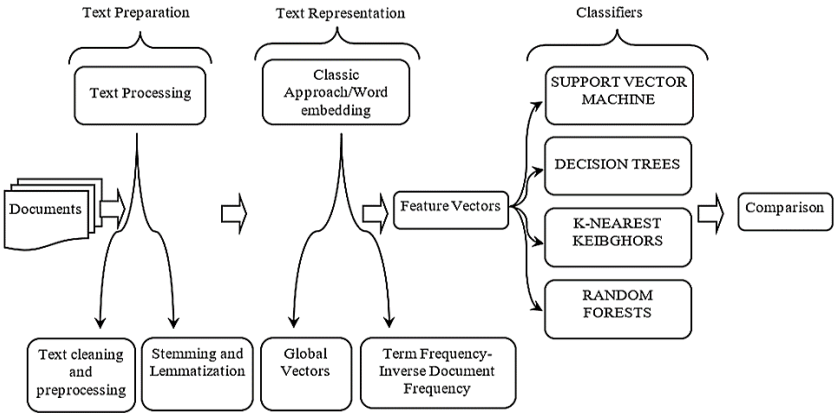

Figure 1. Methodology and workflow of present paper

4.1. Text preprocessing

Text cleaning and pre-processing are crucial steps for improving the performance of text categorization. This stage involves removing unnecessary and nonsensical terms from the data. In our evaluation, each dataset underwent the following procedures: elimination of punctuation and numerals, as well as the removal of stop words. Additionally, tokenization is another essential pre-processing approach, which breaks down a text into smaller units called tokens. Tokens can be words, sentences, or other significant parts of the text. The main goal here is to ensure that sentences are correctly processed. Text documents often contain common but uninformative words like "before," "the," "after," and "a." These words are typically removed from text documents to improve analysis accuracy. Finally, stemming and lemmatization are employed to handle different forms of words while preserving their semantic meaning. This technique helps in reducing the feature space by merging various word forms into a common representation, ultimately aiding in text classification.

4.2. Text representation

4.2.1. Term frequency-inverse document frequency

Jones [31] developed the inverted document frequency (IDF) technique to reduce the influence of frequently used words in a corpus in conjunction with term frequencies. Words that appear frequently or infrequently in a document are given more weight by IDF. When combined with term frequency (TF), this yields the document's TF-inverse frequency (TF-IDF). Although the IDF attempts to address the issue of common terminology in documents, this approach has limitations. Because each word is represented independently as an index, TF-IDF ignores word similarity within the document. In recent years, however, new methods with more complex models, such as word embedding, which can incorporate notions such as word similarity and speech recognition, have been introduced.

4.2.2. Word embedding: GloVe

Word embedding is a category of feature-learning algorithms that entails mapping each word or phrase in a lexicon into real-number vectors (N-dimension vector). Numerous word embedding approaches have been developed to turn unigrams into inputs appropriate for machine learning models. Word2Vec and GloVe are two of the most prevalent and successful deep learning approaches.

GloVe is a robust word embedding technique that has been used for text document classification [9]. In this method, words are also represented as high-dimensional vectors and trained using a large corpus of neighboring words. Pre-trained word embeddings are used in many works and are based on 400,000 trained words from Wikipedia 2014 and Gigaword 5. Word presentation is performed using 50 dimensions. GloVe also provides pre-trained word vectorizations with 100, 200, and 300 dimensions.

5. EXPEREMENT AND EVALUATION

In this section, we compare each of the strategies and algorithms. In addition, we investigate the flaws of current categorization strategies and evaluation methodologies. The purpose is to select an efficient technique of classification while understanding the similarities and variations between existing systems.

5.1. Dataset

Text categorization corpora are collections of texts that have been classified into distinct categories or subsets. Annotated datasets, which contain text document samples with labels, have expedited the expansion of this subject. We investigate the domain-specific characteristics of the four datasets included in this study. Table 2 provides a summary of datasets by category, average phrase length, dataset size, related publications, data sources, and expected applications. By evaluating these datasets, we gain a greater understanding of text categorization issues and opportunities. This can enhance classification techniques and tools for several applications.

- IMDB:25,000 IMDB film reviews, categorized by sentiment (positive/negative). Following pre-processing, each evaluation is encoded as a series of word indexes (integers). For instance, the number "3" represents the third most common term in the data.
- Reuters-21578: 11,228 newswires from reuters, categorized under 46 themes. It is a dataset with several classes and labels. It includes 90 total classes, 7,769 training documents, and 3,019 testing documents.
- 20 newsgroups: The 20-newsgroup dataset contains roughly 18,000 newsgroup posts on 20 themes, separated into training and testing subsets. The distinction between the train set and the test set is determined by communications posted before and after a given date.
- Web of science dataset: This dataset consists of 11,967 documents classified into 35 categories, including seven parent categories.

5.2. Extraction methods

After preprocessing the data, the TF-IDF extractor from the scikit-learn toolbox is used to vectorize the texts for input into classifiers [28]. In a similar fashion, a pre-trained GloVe [32], [33] model is utilized to construct the GloVe feature extractor by averaging the vectorized word representations of the words in the document. The GloVe model was trained on data from Wikipedia and Gigaword 5 [9], with 6 billion tokens and 400,000 concepts in its lexicon [28]. This technique includes both semantics and context without requiring N-grams to assess the input. This article aims to offer a thorough introduction to text categorization approaches, including preprocessing procedures, assessment methodologies, and a comparison of various algorithms and strategies. In addition, we explore the limits of current classification and assessment strategies and emphasize the difficulties in selecting an efficient classification system by comprehending the similarities and differences between existing systems throughout pipeline phases. Two tests were performed, each with a different feature extraction approach, and four ML classifiers were used. All tests were carried out on Intel Core i5-6500 CPUs with 16 GB of RAM.

5.2.1. Experiment 1

Prior to applying the ML algorithms, the first experiment was carried out using the TF-IDF feature extraction approach. Table 4 displays the accuracies obtained by several classifiers, with the best accuracy highlighted in bold. According to the findings of experiment 1, SVM, KNN, and RF yield high accuracy of more than 80%. Table 4 displays the classification scores when utilizing the TFIDF extraction technique and clearly indicates that the SVM classifier outperforms the TF-IDF extraction approach. When utilizing TF-IDF, the SVM classifier has four of the top assessed scores.

Table 4. The performance (precision, recall, f-measure ($F1$)) and accuracy of the different classification algorithms using TF-IDF vectorization techniques

Metric	Dataset	SVM	KNN	DTs	RFs
Accuracy	Reuters-21578	**0.90**	0.80	0.77	0.80
	20 newsgroups	0.85	0.66	0.55	0.76
	WOS	0.83	0.63	0.75	0.85
	IMDB	0.87	0.67	0.70	0.84
F1 score	Reuters-21578	**0.89**	0.79	0.77	0.77
	20 newsgroups	0.85	0.66	0.55	0.76
	WOS	0.82	0.62	0.74	0.85
	IMDB	0.87	0.67	0.70	0.84
Precision	Reuters-21578	**0.89**	0.81	0.77	0.78
	20 newsgroups	0.85	0.67	0.56	0.77
	WOS	0.82	0.63	0.75	0.86
	IMDB	0.87	0.68	0.70	0.84
Recall	Reuters-21578	**0.90**	0.80	0.77	0.80
	20 newsgroups	0.85	0.66	0.55	0.76
	WOS	0.82	0.63	0.75	0.85
	IMDB	0.87	0.67	0.70	0.84

5.2.2. Experiment 2

When using Glove [34] extraction approach on data, the SVM and KNN classifiers perform equally well, as shown in Table 5. It is notable, however, that when evaluating the IMDB datasets, the random forests classifier emerges as the top performer across all metrics evaluated. This observation highlights the dataset-specific nuances that can impact classifier effectiveness. While SVM and KNN remain competitive in the majority of instances, the IMDB dataset presents a unique challenge in which the random forests classifier consistently demonstrates it is efficacy across multiple evaluation criteria. This insight emphasizes the significance of selecting an appropriate embedding technique and classifier based on the specific characteristics of the dataset under consideration, as this decision can have a substantial impact on classification outcomes.

5.3. Discussion

Figures 2 and 3 demonstrate that the maximum accuracy for recognizing the reuters dataset is 90 percent, according to the best accuracy of each approach as indicated in the Figures 2. TF-IDF consistently beats Word Embedding in most models, according to our observations. This finding might be due to several factors. Word Embeddings is unable to generate links between new occurring words and use them for training due to a lack of vectors and associations in GloVe Word Embeddings. TF-IDF, on the other hand, builds vectors using the whole vocabulary available in the train data. Overfitting is also a common issue when using word embeddings. Because word embedding is a complex type of word representation (in

addition to the limited vocabulary), it is quite conceivable that the train data is over-fitted in our experiment. Another downside of using complex word representations is that they contain more hidden information, which is especially useless in our case, but we see in the results that word embeddings utilize links between words to get better precision in the case of random forests.

Table 5. The performance (precision, recall, f-measure (*F*1)) and accuracy of the different classification algorithms using Glove vectorization techniques

Metric	Dataset	SVM	KNN	DTs	RFs
Accuracy	Reuters-21578	**0.74**	**0.73**	0.61	0.72
	20 newsgroups	0.49	0.41	0.24	0.43
	WOS	0.56	0.49	0.25	0.52
	IMDB	0.61	0.56	0.53	**0.74**
F1 score	Reuters-21578	**0.70**	**0.70**	0.60	0.68
	20 newsgroups	0.47	0.41	0.24	0.41
	WOS	0.54	0.48	0.25	0.50
	IMDB	0.61	0.55	0.53	**0.74**
Precision	Reuters-21578	0.69	0.69	0.59	0.68
	20 newsgroups	0.47	0.42	0.24	0.42
	WOS	0.55	0.50	0.25	0.51
	IMDB	0.61	0.57	0.53	**0.74**
Recall	Reuters-21578	**0.74**	**0.73**	0.61	0.72
	20 newsgroups	0.49	0.41	0.24	0.42
	WOS	0.56	0.49	0.25	0.50
	IMDB	0.61	0.56	0.53	**0.74**

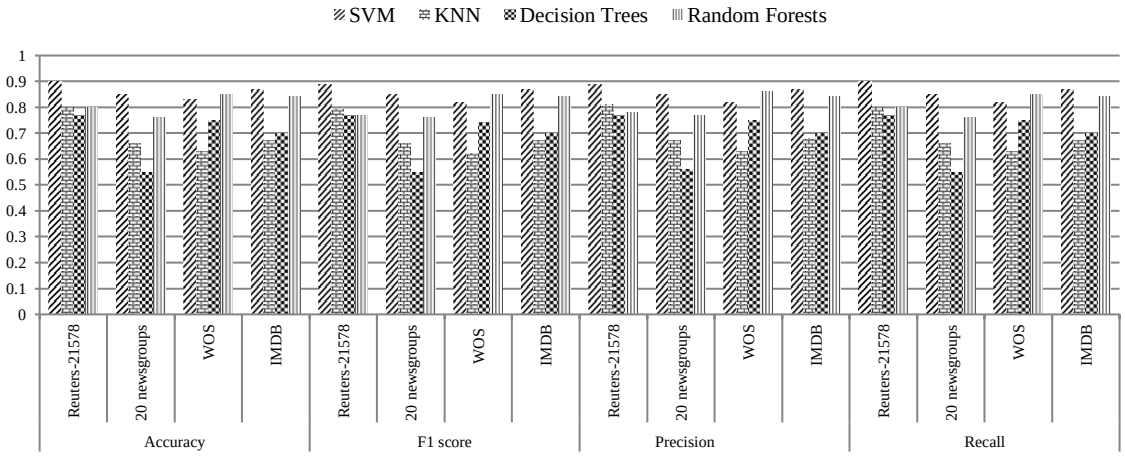

Figure 2. Results of classification methods with TF-IDF vectorization algorithm

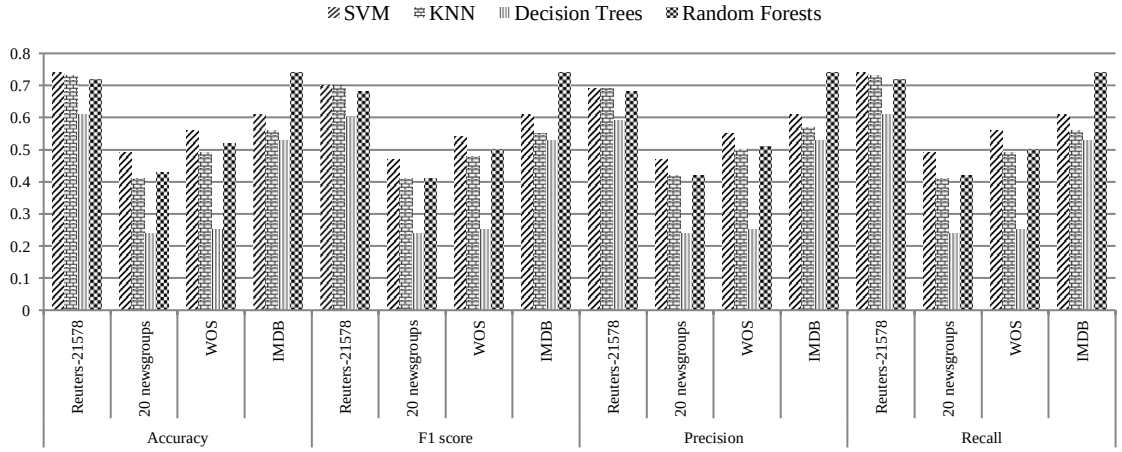

Figure 3. Results of classification methods with GloVe50 vectorization algorithm

6. CONCLUSION

In recent years, text classification has risen in prominence, resulting in the application of numerous data mining methods to the text domain. The performance of many of these methods is hindered by the presence of high-dimensional characteristics and hidden meanings in text data. All of the methods presented in the article have advantages and disadvantages, and selecting the optimal classifier for the task is essential for good classification performance. A combination of an adequate classifier selection and dimensionality reduction technique would surely improve the classification outcome.

Text categorization is a major challenge in machine learning, especially as text and document datasets grow. To address this issue, it is critical to create and disseminate supervised machine learning methods, particularly for text categorization. Existing algorithms must be evaluated to improve existing document classification systems. Nonetheless, improving existing text classification algorithms requires a better understanding of feature extraction methods and how to accurately evaluate them. Text classification approaches are currently classified primarily as follows: In both academic and commercial applications, TF-IDF, TF, and GloVe are extensively used feature extraction techniques. In this study, we discussed classic supervised techniques. In contrast, text and document cleaning can increase an application's correctness and robustness. We examined the essential pre-processing techniques for text. We also define Existing classification approaches such as the KNN, SVM, DTC, RF, and conditional random field are the primary focus of this study (CRF). Accuracy and precision evaluation methodologies were applied to measure performance. Using these metrics, the classification algorithm for text may be evaluated.

This article concludes with a summary of recent developments in supervised techniques and the evolution of text categorization algorithms. It highlights the continuous progress in harnessing machine learning methods to enhance the accuracy and efficiency of text classification tasks. In the upcoming article, our focus will shift toward deep learning algorithms, exploring their most recent developments in the field of natural language processing. Additionally, we will conduct a comparative analysis of these deep learning techniques, evaluating their performance when pair with traditional text representation methods like TF-IDF and GloVe.

APPENDIX

Table 3. Text categorization techniques comparison using the following criteria: strategy used, review element, key contribution (novelty), and corpus of each methodology

Issues Articles	Approach used	Review element	Main contribution	Dataset
A boosted SVM classifier trained by incremental learning and detrimental unlearning approach [13]	Incremental learning and detrimental unlearning approach	– Due to the availability of large data sets in high-dimensional settings, the SVM classifier suffers from low convergence and high memory needs. – These problems are readily apparent in the fields of document classification. – Boosting is a powerful method for enhancing the performance and accuracy of insufficient SVM classifiers.	– An innovative boosting method based on the ideas of incremental learning and detrimental unlearning. – The boosting technique has been applied to numerous fake and real-world datasets of differing sizes, dimensions, forms, and configurations. – Experiment findings demonstrate that the Boosting algorithm lowers training time and improves the performance of a weak SVM classifier.	Artificial dataset (Linear separable two-dimensional Gaussian dataset)
Multi-class document classification using support vector machine (svm) based on improved naïve bayes vectorization technique [35]	Improved Naïve Bayes Vectorization Technique.	– Currently, multiple vectorization strategies are employed to convert text data to a numerical format. – To handle vectorized data with enormous dimensions, a large number of features transformed from text data in a single document require time. – This work seeks to reduce the dimensionality of data.	– To minimize the number of dimensions, this study employs an enhanced Nave Bayes method to vectorize texts based on a probability distribution indicating the document's probable groups or classes. – This paper presents an enhanced Nave Bayes vectorization strategy that incorporates a smoothing technique to overcome the zero probability of unseen data and the use of the logarithmic function to avoid underflow error. It proposes an enhanced vectorization technique for text documents utilizing Naive Bayes as the vectorizer and the probability distribution, where the number of accessible categories in the classification task determines the dimension of the features.	– WebKB Dataset – Song Lyrics Dataset – News Headlines Dataset

Table 3. Text categorization techniques comparison using the following criteria: strategy used, review element, key contribution (novelty), and corpus of each methodology (*continue*)

Issues Articles	Approach used	Review element	Main contribution	Dataset
Adaptive random forests for evolving data stream Classification [36]	Adaptive random forests using an effective resampling mechanism and adaptive operators to deal with various forms of concept drifts without requiring extensive optimizations for various data sets.	– -Random forests are currently one of the most popular non-streaming (batch) machine learning methods. – -This choice is due to its great learning performance and low input preparation and hyper-parameter tuning requirements, yet in comparison to bagging and boosting-based algorithms, there is no random forests solution that can be regarded state-of-the-art in the demanding setting of developing data streams	– The adaptive random forests (ARF) technique was proposed in this paper, which allows the Random Forests algorithm to be used for dynamic data stream learning. – A series of parallel implementations of ARF[S] and ARF[M] have been provided, demonstrating that the parallel version can handle the same number of instances in an acceptable period of time without sacrificing classification performance. – The description of stream learning according to when labels are provided (immediate and delayed settings) is an additional contribution of this work.	– LEDa – LEDg – SEAa – SEAg – AGRa – AGRg – RTG – RBFm – RBF f – HYPER – AIRL – ELEC – COVT – GMSC – KDD99 – SPAM
Improving Random Forests by Neighborhood Projection for Effective Text Classification [22]	A lazy version of the traditional RF classifier (called LazyNN RF), designed specifically for high-dimensional noisy classification tasks	This article introduced a lazy version of the standard random forest classifier, which was specifically developed for sparse high-dimensional noisy classification applications.	– The LazyNN RF classifier, a lazy version of the traditional random forest classifier, was proposed in this article. – The LazyNN RF "localized" training projection is made up of examples that are more similar to the test example. – The experiments, which took into account both topic and sentiment classification, revealed that the LazyNN RF consistently outperforms the explored state-of-the-art classifiers, being the only classifier to achieve the best performance in almost all tested reference datasets. – This provides strong evidence in favor of the potential of exploring data neighborhood in RF models, in the form of a projected (and reduced) training set in the test.	– 20Newsgroups – 4 Universities – Reuters – ACM-DL – UniRCV1 – MEDLINE – Amazon – BBC – Debate – Digg – MySpace – NYT – Tweets – Twitter – Yelp – Youtube

REFERENCES

[1] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen, "Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics*, vol. 17, no. 12, pp. 1131–1142, Dec. 2001, doi: 10.1093/bioinformatics/17.12.1131.

[2] L. M. Manevitz and M. Yousef, "One-class SVMs for document classification," *Journal of machine Learning research*, vol. 2, pp. 139–154, 2001.

[3] E.-H. Han and G. Karypis, "Centroid-based document classification: Analysis and experimental results," in *European conference on principles of data mining and knowledge discovery*, 2000, pp. 424–431.

[4] B. Xu, X. Guo, Y. Ye, and J. Cheng, "An improved random forest classifier for text categorization," *Journal of Computers*, vol. 7, no. 12, Dec. 2012, doi: 10.4304/jcp.7.12.2913-2920.

[5] D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in *IJCAI*, 2007, pp. 2862–2867.

[6] C. Zhang, "Automatic keyword extraction from documents using conditional random fields," *Journal of Computational Information Systems*, vol. 4, no. 3, pp. 1169–1180, 2008.

[7] P. Kanerva, J. Kristoferson, and A. Holst, "Random indexing of text samples for latent semantic analysis," *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2000.

[8] M. Sahlgren, "An introduction to random indexing," 2005.

[9] J. Pennington, R. Socher, and C. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543, doi: 10.3115/v1/D14-1162.

[10] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.

[11] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: a survey," *Information*, vol. 10, no. 4, Apr. 2019, doi: 10.3390/info10040150.

[12] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting LSTM networks for semi-supervised text classification via mixed objective function," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 6940–6948, Jul. 2019, doi: 10.1609/aaai.v33i01.33016940.

[13] R. Kashef, "A boosted SVM classifier trained by incremental learning and decremental unlearning approach," *Expert Systems with Applications*, vol. 167, Apr. 2021, doi: 10.1016/j.eswa.2020.114154.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, Jan. 2013.
[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
[16] P. Iswarya and V. Radha, "Ensemble learning approach in improved k nearest neighbor algorithm for text categorization," in *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2015, pp. 1–5.
[17] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, no. 7, pp. 1545–1588, Oct. 1997, doi: 10.1162/neco.1997.9.7.1545.
[18] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995, vol. 1, pp. 278–282, doi: 10.1109/ICDAR.1995.598994.
[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–23, 2001, doi: 10.1023/A:1010950718922.
[20] W. G. Touw *et al.*, "Data mining in the life sciences with random forest: a walk in the park or lost in the jungle?," *Briefings in Bioinformatics*, vol. 14, no. 3, pp. 315–326, May 2013, doi: 10.1093/bib/bbs034.
[21] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining data with random forests: A survey and results of new tests," *Pattern Recognition*, vol. 44, no. 2, pp. 330–349, Feb. 2011, doi: 10.1016/j.patcog.2010.08.011.
[22] T. Salles, M. Gonçalves, V. Rodrigues, and L. Rocha, "Improving random forests by neighborhood projection for effective text classification," *Information Systems*, vol. 77, pp. 1–21, Sep. 2018, doi: 10.1016/j.is.2018.05.006.
[23] D. Sahgal and M. Parida, "Object recognition using gabor wavelet features with various classification techniques," in *Proceedings of the Third International Conference on Soft Computing for Problem Solving: SocProS 2013, Volume 1*, 2014, pp. 793–804.
[24] G. P. Sanjay, V. Nagori, G. P. Sanjay, and V. Nagori, "Comparing existing methods for predicting the detection of possibilities of blood cancer by analyzing health data," *IJIRST-International Journal for Innovative Research in Science and Technology*, vol. 4, pp. 10–14, 2018.
[25] S. Karamizadeh, S. M. Abdullah, M. Halimi, J. Shayan, and M. javad Rajabi, "Advantage and drawback of support vector machine functionality," in *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*, Sep. 2014, pp. 63–65, doi: 10.1109/I4CT.2014.6914146.
[26] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. Wiley, 2013.
[27] J. R. Quinlan, "Simplifying decision trees," *International Journal of Human-Computer Studies*, vol. 51, no. 2, pp. 497–510, Aug. 1999, doi: 10.1006/ijhc.1987.0321.
[28] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[29] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 2, pp. 225–231, Feb. 2020, doi: 10.1016/j.jksuci.2018.05.010.
[30] S. Bahassine, A. Madani, and M. Kissi, "Comparative study of arabic text categorization using feature selection techniques and four classifier models," in *Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications*, Sep. 2020, pp. 1–5, doi: 10.1145/3419604.3419778.
[31] K. S. Jones, "IDF term weighting and IR research lessons," *Journal of Documentation*, vol. 60, no. 5, pp. 521–523, Oct. 2004, doi: 10.1108/00220410410560591.
[32] H. K. Obayed, F. S. Al-Turaihi, and K. H. Alhussayni, "Sentiment classification of user's reviews on drugs based on global vectors for word representation and bidirectional long short-term memory recurrent neural network," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 23, no. 1, pp. 345–353, Jul. 2021, doi: 10.11591/ijeecs.v23.i1.pp345-353.
[33] R. Adipradana, B. P. Nayoga, R. Suryadi, and D. Suhartono, "Hoax analyzer for Indonesian news using RNNs with fasttext and glove embeddings," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2130–2136, Aug. 2021, doi: 10.11591/eei.v10i4.2956.
[34] Z. Iklima, T. M. Kadarina, and M. H. I. Hajar, "Sentiment classification of delta robot trajectory control using word embedding and convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 26, no. 1, pp. 211–220, Apr. 2022, doi: 10.11591/ijeecs.v26.i1.pp211-220.
[35] H. T. Sueno, "Multi-class document classification using support vector machine (SVM) based on improved naïve Bayes vectorization technique," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3937–3944, Jun. 2020, doi: 10.30534/ijatcse/2020/216932020.
[36] H. M. Gomes *et al.*, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9–10, pp. 1469–1495, Oct. 2017, doi: 10.1007/s10994-017-5642-8.

BIOGRAPHIES OF AUTHORS

Zakia Labd earned her master's degrees in software engineering and cloud computing from the Faculty of Science at Ibn Tofail University in Kenitra, Morocco, in 2019. She is a Ph.D. student in the Department of Computer Science at the University of Ibn Tofail in Kenitra, where she is also a member of the research in Informatics Laboratory (L@RI). Her areas of interest in research include natural language processing, machine learning, and text mining. She can be contacted at email: zakia.labd@uit.ac.ma.

Said Bahassine received his Ph.D. degree from Faculty of Sciences, Chouaib Doukkali University, El Jadida, Morocco in 2019. He is currently a Professor in Department of Computer Science, National Higher School of Arts and Crafts, Hassan II University, Casablanca, Morocco. Member of the Laboratory of Artificial Intelligence and Complex Systems Engineering (AICSE), his research interests include natural language processing, feature selection, machine learning and text mining. He is the author of many research papers published at conference proceedings and international journals. He can be contacted at email: said.bahassine@univh2c.ma.

Khalid Housni received the Master of Advanced Study degree in applied mathematics and computer science, and the Ph.D. degree in computer science from the Ibn Zohr University of Agadir, Morocco, in 2008 and 2012, respectively. He joined the Department of Computer Science, University Ibn Tofail of Kenitra, Morocco, in 2014, where he has been involved in several projects in video analysis and network's reliability. In 2019 he obtained his HDR degree (Habilitation à Diriger desRecherches: Qualification to supervise research) from Ibn Tofail University. He is a member of the research in Informatics Laboratory (L@RI) and head of the MISC team. His current research interests include image/video processing, computer vision, machine learning, artificial intelligence, pattern recognition, and networks reliability. He can be contacted at email: housni.khalid@uit.ac.ma.

Fatima Zahrae Ait Hamou Aadi earned her master's degrees in computer science research from the Faculty of Science at Ibn Tofail University in Kenitra, Morocco, in 2017. She is a Ph.D student in the Department of Computer Science at the University of Ibn Tofail in Kenitra, where she is also a member of the research in Computer Scence Laboratory. Her areas of interest in research include compter vision, machine learning, and artificial intelligence. She is the author of many research papers published at conference proceedings and international journals. She can be contacted at fatimazahrae.aithamouaadi@uit.ac.ma.

Khalid Benabbes is a Ph.D. student at the MISC Laboratory, Faculty of Sciences, Ibn Tofail University, Kénitra, Morocco. He is currently a Software Engineer at the Hassan II Institute of Agronomy and Veterinary Medicine in Rabat. He holds an engineering degree in Computer Sciences from ENSA, Agadir. His research interests include MOOC, recommender systems, machine learning, and data science. He can be contacted at email: khalid.benabbes@uit.ac.ma.