ISSN: 1692-7257 - Volume 1 – Number 43 - 2024

RCTA
Revista Colombiana de Tecnologías de Avanzada
UNIPAMPLONA

Digital Object Identifier: 10.24054/rcta.v1i43.2506

Impact of preprocessing on automatic text classification using supervised learning and reuters 21578

Impacto del preprocesamiento en la clasificación automática de textos usando aprendizaje supervisado y reuters 21578

Ing. José Manuel Arengas Acosta[1], PhD. Rafael Guzmán Cabrera[1]
PhD. Misael López Ramírez[1]

[1] ***Universidad de Guanajuato,*** *Campus Irapuato-Salamanca, Departamento de Estudios Multidisciplinario, Yuriria, Guanajuato, Mexico.*

Correspondencia: *jm.arengasacosta@ugto.mx*

Received: *October 15, 2023.* ***Accepted:*** *December 17, 2023.* ***Published:*** *March 31, 2024.*

How to cite: *J. M. Arengas Acosta, M. Lopez Ramirez, and R. Guzman Cabrera, "Impact of preprocessing on automatic text classification using supervised learning and reuters 21578", RCTA, vol. 1, no. 43, pp. 110–118, Mar. 2024.*
Retrieved from https://ojs.unipamplona.edu.co/index.php/rcta/article/view/2506

Abstract: Faced with the increasing generation of digital data, challenges emerge in its management and categorization. This study emphasizes automatic text classification, placing special emphasis on the impact of preprocessing. By using the Reuters 21578 dataset and applying supervised learning algorithms such as Random Forest, k-Nearest Neighbors, and Naïve Bayes, we examined how techniques like tokenization and the removal of stop words influence classification accuracy. The findings underscore the added value of preprocessing, singling out "Random Forest" as the optimal algorithm, achieving a precision of 92.2%. This research illustrates the potential of combining preprocessing techniques and machine learning algorithms to enhance text categorization in the digital age.

Keywords: Automatic text classification, Preprocessing, Reuters 21578, machine learning.

Resumen: Ante la creciente generación de datos digitales, surgen retos en su gestión y categorización. Este estudio enfatiza en la clasificación automática de textos, poniendo especial énfasis en el impacto del preprocesamiento. Al emplear el conjunto de datos Reuters 21578 y aplicar algoritmos de aprendizaje supervisado como Random Forest, k-Vecinos Más Cercanos y Naïve Bayes, se analizó cómo técnicas como la tokenización y eliminación de palabras vacías influencian la precisión clasificatoria. Los hallazgos resaltan el valor agregado del preprocesamiento, destacando a "Random Forest" como el algoritmo óptimo, alcanzando una precisión del 92.2%. Este trabajo ilustra la potencialidad de combinar técnicas de preprocesamiento y algoritmos para mejorar la categorización de textos en la era digital.

Palabras clave: Clasificación automática de texto, Preprocesamiento, Reuters 21578, aprendizaje automático.

1. INTRODUCTION

Every second, everyone connected to the internet generates a vast amount of information in digital formats such as image, audio, video, and text. An inevitable question arises: what to do with all this volume of information? Faced with this reality, companies from various sectors are compelled to develop tools to analyze and manage such information.

In the field of communication, for example, news agencies need to appropriately categorize their content, assigning it to sections such as sports, economy, politics, or entertainment [1] In turn, readers wish to filter or find news focused on specific topics or characteristics, aiming to streamline the process and access information of interest quickly. However, manually labeling these contents requires considerable human effort and time. Here is where contemporary technology becomes relevant, driving the development of tools and programs that, through artificial intelligence, automate this process [2], [3], [4], [5], [6].

Thus, automatic text classification systems emerge, created to emulate the human task of categorization. To develop a program that automatically classifies textual documents, one approach is to have a set of already labeled data and use supervised machine learning algorithms [7]. During the training process, these algorithms learn the distinctive features associated with each category based on the provided data. Thus, once the model is trained, when an unexamined document is introduced, the system has the capability to classify it with a significant degree of accuracy. The implementation of this training process encompasses various stages, including data selection, preprocessing, choosing relevant features, the training itself, and, finally, evaluating the obtained results [8].

This study conducts a comparative analysis with the objective of observing the impact of various preprocessing techniques on automatic text classification. These techniques include tokenization, conversion to lowercase, removal of low-frequency words, as well as the elimination of stop words, numbers, punctuation marks, and special characters. It aims to understand how these techniques affect precision metrics, the F score. To achieve this purpose, the Reuters 21578 dataset is used, and three supervised learning algorithms are implemented: Random Forest, k-Nearest Neighbors (k-NN), and Naïve Bayes.

2. STATE OF THE ART

Supervised learning has established itself as a key technique in text mining and automatic classification. The ability to process and categorize large volumes of data in real-time and efficiently is essential in the current era of digital information.

Zdrojewska in [9] delved into the world of reinforcement algorithms, highlighting the superiority of boosting algorithms over other traditional methods. Her research with the Reuters dataset shows that the evolution and improvement of classification techniques are a constant need. By transforming documents into vectors using TF-IDF weighting and utilizing specialized libraries like Scikit-learn, it is possible to achieve results that surpass previous studies.

On the other hand, [10] faced the challenge of categorizing texts in the Lao language, a linguistic domain with limited resources. Through the optimization of the KNN algorithm and the application of normalization techniques, they achieved a considerable accuracy of 69.2%. This research underscores the importance of adapting and optimizing supervised learning techniques for different languages and cultural contexts.

In [11], analyzing extensive volumes of unstructured text highlighted the invaluable information that textual data can provide, especially in business decision-making and in preventing unwanted emails. The need for automation is palpable, and this is where Natural Language Processing (NLP) takes center stage.

Similarly, [12] research emphasizes the significance of supervised machine learning in text categorization. The evolution from manual categorization to the use of advanced algorithms like Naïve Bayes and SVM shows the rapid progress and adaptability of supervised learning.

In conclusion, these studies demonstrate the growing importance of supervised learning in automatic text classification. The ability of these techniques to adapt, evolve, and offer precise results in different contexts and languages reinforces their relevance in the field of text mining and natural language processing.

3. STAGES IN AUTOMATIC TEXT CLASSIFICATION

Automatic text classification involves analyzing and grouping text documents based on their features and content. It consists of several stages, outlined as follows:

3.1. Data Set

Refers to the initial group of documents to be classified. In supervised learning, it is crucial that these documents are properly labeled. This labeling facilitates the algorithm's task, allowing it to learn and distinguish the different classification categories.

For this study, the "Reuters 21578" dataset [13] is employed his is a standardized collection of texts, widely used in classification tests, comprising files containing news reports and articles from the Reuters news agency.

3.2. Preprocessing

ext preprocessing is an essential stage in automatic text classification. Its purpose is to prepare and structure the information, simplifying the text to facilitate later tasks, such as training machine learning algorithms [14]. This process makes the patterns and relevant features of each category more prominent, resulting in more coherent, interpretable, and manageable text for the algorithms. During this stage, a "Baseline" or reference point is established, allowing for the comparison of results obtained in different studied scenarios [8].

Preprocessing techniques used in this study include:

3.2.1. Baseline Selection

This technique involves preparing the data for the study. It includes extracting text from each file based on specific characteristics related to the intended classification [15].

3.2.2. Tokenization

This technique involves dividing text into smaller units called "tokens". Although these tokens are commonly words, they can also be phrases, sentences, or symbols. Tokenization simplifies and structures the text, facilitating its subsequent processing and analysis.

3.2.3. Conversion to Lowercase

This technique refers to the process of transforming all the letters in a text to lowercase. The main goal is to homogenize the text to avoid duplications and reduce variability, eliminating differences caused using uppercase and lowercase letters. For instance, the words "Book", "BOOK", and "book" would be treated as different entities without this conversion. By converting everything to lowercase, it ensures that the words are recognized and processed as the same entity, facilitating subsequent classification tasks.

3.2.4 Removal of Stop Words

This technique involves removing "stop words" or words that, despite being common in a language, lack intrinsic critical meaning. By discarding these terms, noise in the text is minimized, allowing algorithms to focus on semantically more relevant words. This facilitates the analysis and improves the acquisition and understanding of the information.

3.2.5. Removal of Punctuation Marks, Numbers, and Special Characters

During text processing, elements such as punctuation marks, numbers, and special characters, while crucial for human interpretation, can be superfluous or problematic for algorithmic analyses. Removing them purifies the text, minimizing ambiguities and potential confusions in its automatic interpretation. This cleaning process allows algorithms to focus their attention on the core of the content: the words and their contextual meaning.

3.2.6. Elimination of Low-Frequency Words by Category

This technique focuses on omitting terms that appear rarely in each category. By dispensing with them, a more refined data representation is achieved, optimizing the focus of machine learning algorithms on the most significant and predominant features of the category.

3.3. Supervised Learning Algorithms

These algorithms are part of machine learning and function by interpreting relationships in labeled datasets [5], [12], [16]. The training requires supplying the algorithm with a set where each sample is linked to a specific label or class. Once the model is optimized, it can predict labels for new

samples based on the previously identified structures and relationships.

In this study, the following supervised learning algorithms are used: k-Nearest Neighbors (k-NN), Random Forest (RF), and Naïve Bayes (NB).

3.3.1. k-Nearest Neighbors (k-NN):

This algorithm is one of the most basic and essential in supervised learning. It operates under a simple concept: similarity. By introducing a new instance for classification, k-NN searches the training set for the 'k' samples closest to that instance and subsequently assigns the new instance the predominant label among those neighbors. The value of 'k' is crucial for the algorithm's performance: a small 'k' can result in a model susceptible to noise, while a large 'k' can lead to more generalized decisions. Although k-NN is efficient for datasets with low dimensionality, its performance tends to decrease in high-dimensional spaces. [17].

3.3.2. Random Forest (RF)

Is an ensemble learning algorithm using the "bagging" approach, it generates multiple decision trees during training and, when receiving an input, offers a classification based on the mode or an average prediction of these trees, Random Forest addresses the challenge of high data dimensionality by randomly selecting subsets of features for each tree, encouraging diversity, and reducing overfitting. When classifying a new text, each tree in the ensemble contributes its classification, and the final decision is made through a majority vote among all [18], [19].

3.3.3. Naïve Bayes (NB)

Is an algorithm based on Bayes' theorem, known for its robustness and efficiency in classification text, It operates by estimating the probability that a document belongs to a certain category, based on the presence or absence of specific terms in the document. To determine these probabilities, it relies on already labeled training data. This algorithm is commonly used in conjunction with representations like "bag of words" or TF-IDF. In [20] these models, each term or word in the document is turned into a unique feature, enabling the algorithm to discern and categorize texts based on both the semantic content and the frequency of certain terms.

3.4. Experimental Scenarios

Refer to the specific configurations or parameters under which algorithms are trained and evaluated. They include aspects such as the preprocessing techniques applied and strategies for partitioning the dataset, like dividing between training and validation data. The purpose of defining multiple experimental scenarios is to examine each algorithm's performance under different conditions and thereby identify the most effective strategy for the problem at hand. These scenarios enable a systematic comparison of the algorithms, facilitating the identification of inherent advantages and disadvantages of each method in relation to the specific classification problem.

3.5. Evaluation Metrics

In [21] Described as tools to quantify the effectiveness and performance of models and algorithms in machine learning. They provide an objective perspective on a model's capability, assessing its accuracy, reliability, and adaptability to real-world contexts. Using them for validation ensures the precision of predictions and the accuracy of obtained results. The metrics used in this study include precision, recall, and F score.

3.5.1. Precision

t is a metric commonly used in classification tasks to evaluate the quality of predictions. It measures the proportion of cases that the model classified into a category and that belong to it.

It is defined as:

$$\text{Precisión} = \frac{VP}{VP+FP} \quad \textbf{(1)}$$

Where:

VP: True Positives, correspond to the cases that the model classifies into a category and that truly belong to it.
FP: False Positives, correspond to the cases that the model classifies into a category, but, do not belong to it.

3.5.2. Recall (Sensibilidad)

It is a metric that measures the proportion of true positives correctly identified in relation to the total number of actual positive cases. In other words, it indicates what percentage of the real positives was detected by the model. It is mathematically defined as:

$$\text{Recall} = \frac{VP}{VP+FN} \quad (2)$$

Where:

FN: False Negatives, correspond to the cases of a category that the model classifies into another category.

3.5.3. score- F1

Also known as F1-score, it is a measure that combines precision and recall into a single number. It is particularly useful when one wants to balance these two aspects in problems where one may be more relevant than the other. The F1-score is calculated as the harmonic mean between Precision and Recall, providing a balance between precision and recall. It is mathematically defined as:

$$\text{F1-score} = 2 \times \frac{\text{PrecisiónxRecall}}{\text{Precisión+Recall}} \quad (3)$$

4. METHODOLOGY

Within the framework of this research, experiments were conducted using various preprocessing techniques to analyze their influence on evaluation metrics: Precision, Recall, and F1-score. For classification, three algorithms were used: Random Forest, Naïve Bayes, and K Nearest Neighbors, utilizing the Reuters-21578 Corpus as a foundation. The adopted methodology is structured in four essential phases, which are illustrated in Fig. 1. and are described in detail below.

4.1. Data Selection

Within the context of this study, a balanced dataset was generated from articles belonging to specific categories of the Reuters 21578 collection. To achieve this, a detailed filtering of the Reuters 21578 set was carried out. Initially, a distinction was made between labeled and unlabeled documents, resulting in 11,367 labeled documents distributed across 120 categories. Of these, only 10 have more than 100 documents, as shown in Table 1.

From these 10, five categories were randomly selected: Coffee, Crude, Earn, Gold, and Sugar. Subsequently, 80 documents were randomly extracted from each category, leading to a balanced dataset.

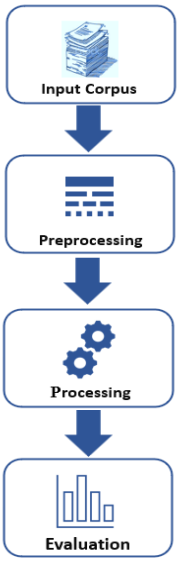

Fig. 1. *Implemented Methodology Scheme*
Source: *Own elaboration.*

Table 1: *Categories with More Than 100 Documents*

Category	# Documents per Category
earn	3735
acq	2125
crude	355
trade	333
money	259
interest	211
ship	156
sugar	135
coffee	114
gold	100

Source: *Own elaboration.*

Table 2 presents the set consisting of 5 categories and a total of 400 documents.

Table 2: *Balanced Dataset*

Category	# Documents per Category
Coffee	80
Crude	80
Earn	80
Gold	80
Sugar	80
∑	**400**

Source: *Own elaboration.*

4.2. Preprocessing

Preprocessing involves a series of techniques aimed at cleaning, preparing, and transforming raw text into a format more suitable for the subsequent classification process. The preprocessing techniques used in this study are shown in Table 3.

Table 3: *Preprocessing Techniques*

#	Technique
I	Tokenization.
II	Conversion to lowercase.
III	Removal of stop words, punctuation marks, and numbers.
IV	Removal of words with a frequency of one.

Source: *Own elaboration.*

The techniques detailed in Table 3 form the basis for establishing the experimental scenarios of this study. These scenarios are configured through combinations of said techniques, as illustrated in Fig. 2.

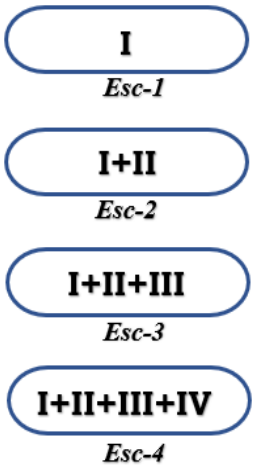

Fig. 2. *Experimental Scenarios*
Source: *Own elaboration.*

The experimental scenarios detailed in Fig. 2 are as follows:

Scenario 1 (Esc.1): Tokenization of the data.

Scenario 2 (Esc.2): Tokenization of the data and conversion of words to lowercase.

Scenario 3 (Esc.3): Tokenization of the data, conversion of words to lowercase, and removal of stop words, punctuation marks, and numbers.

Scenario 4 (Esc.4): Tokenization, conversion to lowercase, removal of stop words, punctuation marks, numbers, and words with a frequency of appearance of one.

4.3. Processing

During processing, each algorithm is trained according to different scenarios, following the structure outlined in Figure 3.

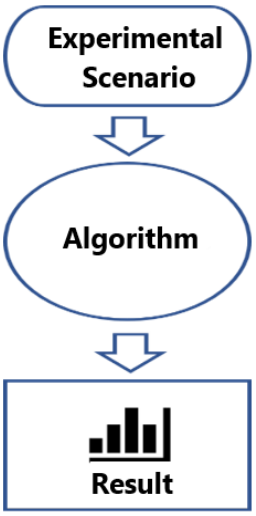

Fig. 3. *Training Scheme*
Source: *Own elaboration.*

The data undergo a preprocessing stage following the specific configuration of techniques inherent to each scenario. Once the information is preprocessed, it is processed using supervised learning algorithms: k-Nearest Neighbors (k-NN), Random Forest (RF), and Naïve Bayes (NB). Finally, the results are presented in terms of metrics: Precision, recall, and F1 score, corresponding to each case.

5. RESULTS

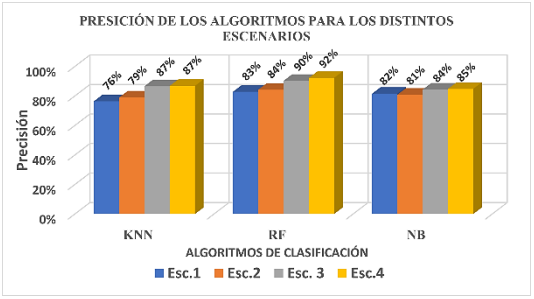

Fig. 4. *Precision Metric of the algorithms in the different scenarios*
Source: *Own elaboration.*

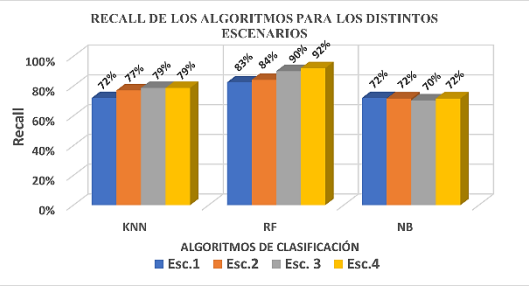

Fig. 5. *Recall Metric of the algorithms in the different scenarios*
Source: *Own elaboration.*

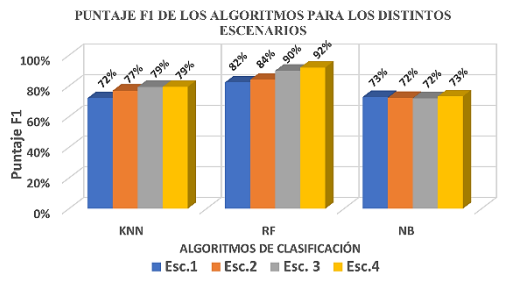

Fig. 6. *F1-score Metric of the algorithms in the different Scenarios*
Source: *Own elaboration.*

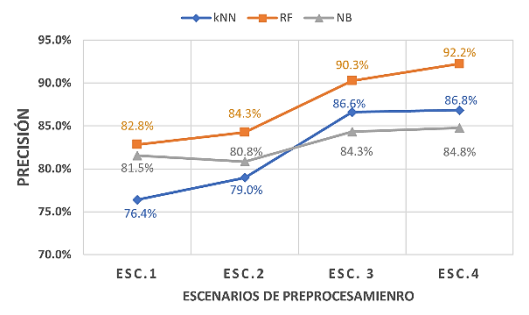

Fig. 7. *Impact of preprocessing techniques on the algorithms in each scenario*
Source: *Own elaboration.*

6. DISCUSSION

Figure 4 displays the precision achieved in each scenario for the different classification algorithms.

For the k-Nearest Neighbors algorithm, a precision of 76% is attained in scenario 1, whereas a precision of 87% is recorded in scenario 4. This indicates that preprocessing techniques increase the precision metric by 11%.

In the case of the Random Forest algorithm, precision is 83% in scenario 1 and rises to 92% in scenario 4, reflecting an improvement of 11% after applying the preprocessing techniques.
Lastly, with the Naïve Bayes algorithm, a precision of 82% is observed in scenario 1 and 85% in scenario 4, representing a 3% improvement due to preprocessing techniques.

Figures 5 and 6 present the results of the evaluation metrics, Recall and F1 score, respectively, for each scenario and classification algorithm.

It is noteworthy that the Naïve Bayes algorithm shows the lowest values in both metrics and does not appear to be significantly affected by preprocessing techniques. On the other hand, in the Random Forest and k-Nearest Neighbors algorithms, preprocessing techniques do seem to have a positive impact on the metrics. Specifically, in Random Forest, there is a 9% increase, while in k-Nearest Neighbors, the increase is 7%.

Figure 7 illustrates the precision of each algorithm per scenario. Initially, the "k-Nearest Neighbors" algorithm shows the lowest precision, while "Random Forest" stands out with the best performance. With the progressive application of preprocessing techniques, a positive effect on the precision of the algorithms is evident. After completing preprocessing, "Random Forest" leads with a precision of 92.2%. In contrast, the "Naïve Bayes" algorithm reports the lowest performance, although still with a respectable 84.8% precision. It is worth mentioning that these percentages are quite satisfactory for automatic text classification.

In all algorithms employed, an improvement in precision is recorded, showing an upward trend in this metric.

7. CONCLUSIONS

Preprocessing techniques have proven to be fundamental in improving the accuracy of automatic text classification. Their systematic application led to significant improvements in evaluation metrics, particularly precision.

Among the evaluated algorithms, "Random Forest" stood out with the highest precision, reaching 92.2%. Although "Naïve Bayes" had the lowest performance, it still achieved a respectable 84.8%, demonstrating that, in this context, all selected algorithms are suitable for the task.

As future work, it would be beneficial to explore other advanced preprocessing techniques or consider the inclusion of unsupervised or more recent classification algorithms to evaluate if they can outperform the current performance.

ACKNOWLEDGEMENTS

I would like to thank the Consejo de Ciencia y Tecnología (CONACYT) for their support through scholarship 1192218 during my graduate studies in Technology Management at the University of Guanajuato, Mexico.

REFERENCES

[1] C. Guardiola González, "Clasificador de textos mediante técnicas de aprendizaje automático," 2020. Accessed: Sep. 27, 2023. [Online]. Available: https://riunet.upv.es:443/handle/10251/133840

[2] Y. Li, "Automatic Classification of Chinese Long Texts Based on Deep Transfer Learning Algorithm," in *2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, IEEE, Nov. 2021, pp. 17–20. doi: 10.1109/ICAICE54393.2021.00011.

[3] D. Onita, "Active Learning Based on Transfer Learning Techniques for Text Classification," *IEEE Access*, vol. 11, pp. 28751–28761, 2023, doi: 10.1109/ACCESS.2023.3260771.

[4] M. A. Tayal, V. Bajaj, A. Gore, P. Yadav, and V. Chouhan, "Automatic Domain Classification of Text using Machine Learning," in *2023 International Conference on Communication, Circuits, and Systems (IC3S)*, IEEE, May 2023, pp. 1–5. doi: 10.1109/IC3S57698.2023.10169470.

[5] L. Zhang, B. Sun, F. Shu, and Y. Huang, "Comparing paper level classifications across different methods and systems: an investigation of Nature publications," *Scientometrics*, 2022, doi: 10.1007/s11192-022-04352-3.

[6] C. Liu, Y. Sheng, Z. Wei, and Y.-Q. Yang, "Research of Text Classification Based on Improved TF-IDF Algorithm," in *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, IEEE, Aug. 2018, pp. 218–222. doi: 10.1109/IRCE.2018.8492945.

[7] A. Rusli, A. Suryadibrata, S. B. Nusantara, and J. C. Young, "A Comparison of Traditional Machine Learning Approaches for Supervised Feedback Classification in Bahasa Indonesia," vol. VII, no. 1, 2020.

[8] D. Ji-Zhaxi, C. Zhi-Jie, C. Rang-Zhuoma, S. Maocuo, and B. Mabao, "A Corpus Preprocessing Method for Syllable-Level Tibetan Text Classification," in *2021 3rd International Conference on Natural Language Processing (ICNLP)*, IEEE, Mar. 2021, pp. 33–36. doi: 10.1109/ICNLP52887.2021.00011.

[9] A. Zdrojewska, J. Dutkiewicz, C. Jędrzejek, and M. Olejnik, "Comparison of the novel classification methods on the reuters-21578 corpus," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2019, pp. 290–299. doi: 10.1007/978-3-319-98678-4_30.

[10] Z. Chen, L. J. Zhou, X. Da Li, J. N. Zhang, and W. J. Huo, "The Lao text classification method based on KNN," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 523–528. doi: 10.1016/j.procs.2020.02.053.

[11] M. Nasr, A. karam, M. Atef, K. Boles, K. Samir, and M. Raouf, "Natural Language Processing: Text Categorization and Classifications," *Advanced Networking and Applications*, vol. 12, no. 02, pp. 4542–4548, 2020.

[12] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artif Intell Rev*, vol. 52, no. 1, pp. 273–292, Jun. 2019, doi: 10.1007/s10462-018-09677-1.

[13] D. D. Lewis, "Machine Learning Repository," Documents came from Reuters newswire in 1987. Accessed: Oct. 18, 2022. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[14] C. L. Hernández and J. E. Rodríguez, "Preprocesamiento de datos estructurados Structured Data Preprocessing," *Investigacion y desarrollo*, vol. 4, no. 2, pp. 27–48, 2013, doi: 10.14483/2322939X.4123.

[15] J. J. Paniagua Medina, E. Vargas Rodriguez, and R. Guzman Cabrera, "Machine Learning And The Reuters Collection-21578 In Document Classification," *Revista Colombiana De Tecnologias De Avanzada (RCTA)*, vol. 2, no. 40, Jul. 2023, doi: 10.24054/rcta.v2i40.2344.

[16] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information (Switzerland)*, vol. 10, no. 4. MDPI AG, 2019. doi: 10.3390/info10040150.

[17] L. A. Calvo-Valverde and J. A. Mena-Arias, "Evaluación de distintas técnicas de

representación de texto y medidas de distancia de texto usando KNN para clasificación de documentos," *Revista Tecnología en Marcha*, Feb. 2020, doi: 10.18845/tm.v33i1.5022.

[18] T. Salles, M. Gonçalves, V. Rodrigues, and L. Rocha, "Improving random forests by neighborhood projection for effective text classification," *Inf Syst*, vol. 77, pp. 1–21, Sep. 2018, doi: 10.1016/j.is.2018.05.006.

[19] J. J. Espinosa Zúñiga, "Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito," *Ingeniería Investigación y Tecnología*, vol. 21, no. 3, pp. 1–16, Jul. 2020, doi: 10.22201/fi.25940732e.2020.21.3.022.

[20] M. Thangaraj and M. Sivakami, "Text classification techniques: A literature review," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 13, pp. 117–135, 2018, doi: 10.28945/4066.

[21] A. Bhavani and B. Santhosh Kumar, "A Review of State Art of Text Classification Algorithms," in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021, pp. 1484–1490. doi: 10.1109/ICCMC51019.2021.9418262.