Analysis of the Influence of Preprocessing Techniques on Text Classification Accuracy: An Investigation with the Naive Bayes Model and the Reuters-21578 Dataset

Jose Manuel Arengas Acosta
Department of Multidisciplinary Studies
University of Guanajuato
Yuriria, México.
jm.arengasacosta@ugto.mx

Rafael Guzmán Cabrera
Department of Multidisciplinary Studies
University of Guanajuato
Yuriria, México.
guzmanc@ugto.mx

Misael López Ramírez
Department of Multidisciplinary Studies
University of Guanajuato
Yuriria, México.
lopez.misael@ugto.mx

***Abstract*— In the domain of Natural Language Processing (NLP), the role of automatic text classification stands paramount. This investigation delves deeply into optimizing classification precision by leveraging advanced preprocessing techniques on the renowned Reuters-21578 corpus, paired with the Naive Bayes classification schema. The research emphasizes the salient role of preprocessing and its consequent impact on model robustness, achieving a commendable validation accuracy of 90%. A meticulous examination of diverse scenarios reveals that meticulous preprocessing methodologies substantially bolster the model's operational efficacy. Additionally, the study accentuates consistent alignment across various evaluation metrics and meticulously scrutinizes the interplay between data volume and model prowess. These insights amplify the indispensable nature of preprocessing within text classification and illuminate prospective avenues for ensuing scholarly endeavors within machine learning and NLP realms.**

Keywords— Preprocessing Techniques; Naive Bayes; Reuters-21578 Automatic Text Classification; Machine Learning

I. INTRODUCTION

The digital revolution has triggered a rapid growth in the generation of digital texts, presenting a significant challenge for the management and classification of this overwhelming amount of information[1]–[3] Given the infeasibility of manually addressing such a volume of data, the scientific and technological community has directed efforts towards the creation of automatic classification systems. These systems, grounded in artificial intelligence and machine learning, promise to efficiently manage and categorize various types of texts[4]–[10].

These systems categorize texts that span across structured, unstructured, and semi-structured categories, each with its inherent challenges and characteristics[3]. A cornerstone of these systems' effectiveness is data preprocessing. This step is essential for improving both the quality of input data and the accuracy of the resulting model [11], [12] indicate, for optimal operation, these systems employ supervised machine learning algorithms. These algorithms are trained with previously labeled datasets. Once trained, they can classify unlabeled documents with impressive accuracy.

Among the algorithms, the Naive Bayes, grounded in Bayes' theorem, stands out as a robust tool for text classification. Especially when combined with representations like "bag of words" or TF-IDF, it can discern and categorize texts based on both semantic content and the frequency of specific terms[13].

With this context, the presented study focuses on analyzing the impact of various preprocessing techniques on the accuracy metric of automatic text classification. Using the Reuters 21578 dataset and the supervised learning algorithm Naïve Bayes, the aim is to determine how interventions in the preprocessing stage—from tokenization to the removal of unwanted characters—can influence performance metrics, specifically accuracy and the F score. This analysis offers clear insights into how the preprocessing stages interact with the final model's effectiveness.

II. THEORETICAL REFERENCE

A. *Literature Review*

Automated text classification has become a fundamental tool for analyzing large volumes of data. One of the primary approaches in this realm has been the use of supervised learning, which has proven to be effective in adapting and evolving based on the specific needs of the context and language, offering accurate and valuable results [10]. In turn, preprocessing techniques have played an essential role in enhancing the accuracy of such classification.

Within the research conducted in this field, the Reuters-21578 corpus has been widely used as a benchmark. [14] focused on improving the Multinomial Naive Bayes classifier by evaluating boosting algorithms like AdaBoost on this corpus, obtaining promising results due to Bayesian optimization [14] This effort closely relates to the work of [15], who introduced a rule-based technique with document embedding in the doc2vec format for text classification, comparing the performance of their approach on the Reuters-21758 corpus [15]

On the other hand, preprocessing, such as the removal of stopwords and lemmatization, has shown to have a direct impact on classification accuracy. [16] investigated preprocessing techniques to enhance the efficiency of text classification, particularly in the Vietnamese linguistic context. Similarly, an unidentified study [17] explored preprocessing algorithms related to stemming and lemmatizing texts, highlighting the importance of these techniques in preparing data for natural language processing applications.

It is evident that, while text classification has advanced significantly, preprocessing techniques remain a crucial part to ensure optimal classification accuracy. With the continuous increase in the quantity and diversity of textual data, it is imperative that research and optimization of these techniques continue to adapt to the ever-changing requirements of natural language processing.

B. *Preprocessing Techniques*

In automated text classification, preprocessing serves as a critical initial step that supports subsequent stages of machine learning and text analysis. The primary aim of text preprocessing is to refine, standardize, and structure textual data, enhancing the efficiency of later classification algorithms. The importance and applications of preprocessing methods in automated text classification are underscored, according to studies [11], [17], [18].

The essence of preprocessing is to illuminate the most informative patterns and features within text datasets. This not only facilitates a more coherent and interpretable text but also renders the data more manageable for algorithms. Moreover, preprocessing techniques help in establishing a "Baseline" or benchmark, essential for contrasting results across diverse scenarios and studies [11]. Let us delve into the key preprocessing techniques.

Baseline Selection [19], [20]: A foundational step that involves data preparation by extracting text from sources based on specific characteristics related to the desired classification.

Tokenization: A cornerstone in natural language processing, tokenization splits the text into smaller units or tokens. These tokens can represent words, phrases, sentences, or even individual symbols, aiding in streamlining and structuring the data for subsequent analyses.

Converting to Lowercase: By transforming all textual content into lowercase, this technique standardizes the text, minimizing potential duplications arising from case differences and streamlining subsequent processing tasks.

Stopwords Removal: This process eliminates commonly used words like "y", "de", "la", which typically lack substantive intrinsic meaning. By doing so, the text is rid of unnecessary noise, enabling algorithms to focus on semantically significant words, enhancing data extraction and comprehension.

Elimination of Punctuation, Numbers, and Special Characters: Stripping the text of elements like punctuation marks, numbers, and other special characters becomes imperative. Though these elements are vital for human interpretation, they can sometimes become redundant or even problematic for computational analyses. This purification allows algorithms to concentrate on the core content: words and their contextual relevance.

Elimination of low-frequency terms.: This technique reduces the text dataset's dimensionality by excluding words with minimal appearances across the corpus. Such an approach ensures the removal of infrequent terms, which might be irrelevant for text classification or could introduce unwarranted noise.

In conclusion, preprocessing techniques are pivotal in ensuring the accuracy of text classification algorithms. By refining the data and accentuating its most salient features, preprocessing sets the stage for effective and accurate analyses, especially when using algorithms like Naive Bayes on datasets such as Reuters-21578. Insights derived from such processed data can yield more precise and actionable results, underscoring the foundational significance of this step in automated text classification.

C. *Dataset*

The Reuters-21578 dataset is a standardized collection comprising 21,578 text files containing articles from the renowned news agency, Reuters. These files are structured in XML format, an open language that employs specific tags to define the content and meaning of the information, facilitating its organization into a hierarchical tree-like structure [20]. Although the copyright of this collection belongs to Reuters Ltd., it has been made available to the academic community under certain conditions, such as acknowledging its use and appropriately citing when publishing results based on these data [21] In the context of supervised learning and preprocessing technique research, this dataset is pivotal due to its properly labeled structure, allowing for more effective and accurate classification. In the current analysis, the "Reuters 21578" dataset is used to assess the influence of various preprocessing techniques on classification accuracy using the Naive Bayes algorithm.

D. Algoritmo Naïve Bayes

The Naïve Bayes (NB) algorithm is a probabilistic classifier rooted in Bayes' theorem. Known for its robustness and efficiency, it excels in text classification tasks. By evaluating the presence or absence of specific terms in a document, the algorithm gauges the probability of the document being associated with a certain category. It heavily relies on pre-labeled training datasets and is frequently integrated with representational methods like "bag of words" or TF-IDF. These models treat every term as an individual feature, which aids NB in distinguishing and categorizing texts, not just based on the semantic content, but also on the frequency of specific terms [13] A distinguishing trait of NB is its foundational assumption of feature independence. This allows it to outperform many classifiers in real-time datasets in terms of accuracy, even with limited training data. Especially beneficial for high-dimensional datasets, the algorithm estimates the probability of each attribute independently, predicting the class of a test instance using the most likely posterior [22].

E. Evaluation Metrics in Text Classification

Evaluation metrics play a pivotal role in quantifying the performance and quality of machine learning models, especially in the domain of text classification. They provide an objective measure, allowing researchers and practitioners to ascertain how well a model's predictions match the actual labels in a dataset. In the context of preprocessing techniques and their influence on text classification accuracy with the Reuters-21578 dataset, the following metrics are of paramount importance:

Precision: This metric captures the proportion of positive instances that the model correctly classifies. It reflects the model's capability to prevent misclassifications, especially avoiding the pitfall of wrongly classifying negative instances as positive. Essentially, precision is the ratio of instances that are correctly predicted as positive to all the instances predicted as positive by the [22]–[24]

$$precision = \frac{TP}{TP+FP}$$

(1)

Where:
TP: True Positives
FP: False Positives

Recall (Sensitivity): Recall, often termed as sensitivity, quantifies the model's ability to identify and retrieve all relevant instances. It indicates the percentage of genuine positives that the model successfully detected. It is a measure of the model's ability to capture all potential positives in the dataset [22]–[24].

$$Recall = \frac{TP}{TP+FN} \qquad \textbf{(2)}$$

Where:
TP: True Positives
FN: False Negatives

F1 Score: Combining the strengths of both precision and recall, the F1 Score provides a harmonized measure, especially vital when there is a need to strike a balance between the two metrics. It offers a comprehensive view of a model's classification performance, ensuring neither precision nor recall is unduly prioritized [22]–[24]

$$\text{F1-score} = 2 \text{ x } \frac{\text{PrecisiónxRecall}}{\text{Precisión+Recall}} \qquad \textbf{(3)}$$

These metrics, when applied to the Reuters-21578 dataset in conjunction with the Naive Bayes classifier, provide insights into the efficacy of preprocessing techniques on text classification accuracy. As the analysis progresses, they will be instrumental in validating and understanding the impact of preprocessing strategies on the dataset's classification results.

F. General Training of Machine Learning Models

Initialization: The model initializes certain parameters. These parameters are the values that the model will adjust during training to enhance its predictions.

Feed Data: The model ingests a data sample from the training set and makes a prediction based on its current parameters.

Compute Error: Once the model has made a prediction, it compares this prediction to the actual value (the label or target) to compute the error. The aim of the training is to minimize this error.

Adjust Parameters: The model employs an algorithm (such as gradient descent in regression models) to adjust its parameters with the aim of reducing the error.

Iterate: The model repeats steps 2-4 numerous times, adjusting its parameters in each iteration to reduce the error.

Termination: The process concludes when the error reaches a suitably low value, or after a predefined number of iterations, or if the error ceases to improve significantly (this might indicate that the model has converged).

III. METODOLOGY

The methodology employed in this study was systematically structured to ensure a comprehensive and organized analysis of automated text classification. It is segmented into the following phases: initiation, data selection, preprocessing, processing, evaluation, and conclusion, as illustrated in Fig.1.

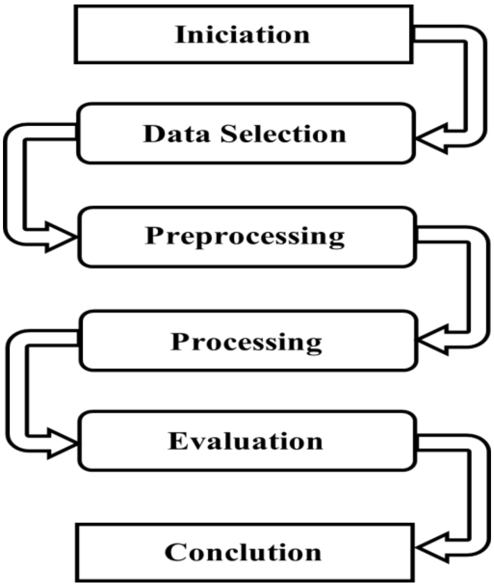


Fig. 1. Proposed Methodology Diagram.

A. *Initiation*

Begin the text classification process

B. *Data selection*

In the context of this study, automatic document classification was conducted using the Reuters-21578 Corpus. This corpus originally encompassed documents distributed across 120 categories.

TABLE I. NUMBER OF DOCUMENTS DISPLAYED BY THE DATABASE

Category name	*# documents by category*
acq	99
coffee	99
crude	99
earn	99
gold	99
interest	99
money-fx	99
ship	99
sugar	99
trade	99
Total	990

- Gather the Reuters-21578 dataset.
- Randomly select eight categories with more than 99 documents.

presented in Table I.

C. *Preprocessing*

Four classification scenarios were implemented using data preprocessing techniques, as described, and Fig.2 illustrated in the diagram below.

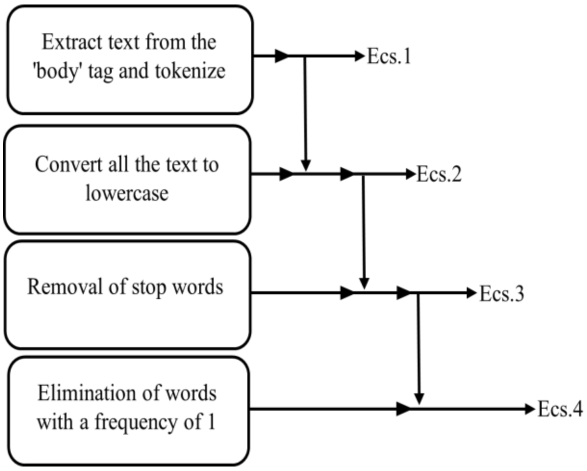

Extract text from the 'body' tag and tokenize
Convert all the text to lowercase
Removal of stop words
Elimination of words with a frequency of 1
Ecs.1
Ecs.2
Ecs.3
Ecs.4

Fig. 2. Classification Scenarios Diagram

D. *Processing*

Training the Naïve Bayes Model with Preprocessed Data:

At this juncture, the refined textual data will serve as the foundation to train our Naïve Bayes model. Rooted in its inherent probabilistic principles, the Naïve Bayes algorithm assimilates patterns from the curated data to facilitate document predictions and classifications. This training phase encompasses feeding the model with labeled data, thereby associating each document with its respective category or class. Through this process, the model discerns features within the text, acquainting itself with correlations between specific terms and their corresponding categories.

Upon successful training of the Naïve Bayes model, it is poised to categorize documents into predetermined classifications autonomously. The model evaluates each document's content against the learned features from the training phase. The ensuing classification demarcates the documents into their perceived categories, thereby shedding light on the model's predictive accuracy and the overall efficacy of the text classification endeavor

E. *Results and Analysis*

For the assessment of this research, key evaluation metrics have been computed, including accuracy, recall, and the F1 score, to gauge the efficacy of the model. The confusion matrix is also provided. The values obtained from both the model's training and its predictions are presented. A comprehensive analysis of these metrics offers insight into the overall performance and robustness of our text classification approach.

IV. RESULTS

In this section, the results stemming from the text classification process of this study are showcased. The first segment displays the outcomes from the training, while the second segment details the results from the model's validation. It should be noted that the selected data is split 70-30; that is, 70 percent is allocated for the training of the Naive Bayes model and 30 percent for its validation.

A. *Training Results*

The model is trained using 70% of the initial data, which equates to 693 documents. The results obtained during the system's training phase are presented below.

1. Results for Scenario 1.

In this scenario, the texts were subjected solely to the preprocessing technique of tokenization. The confusion matrix can be found in Fig.3, and the other evaluation metrics are detailed in Table II.

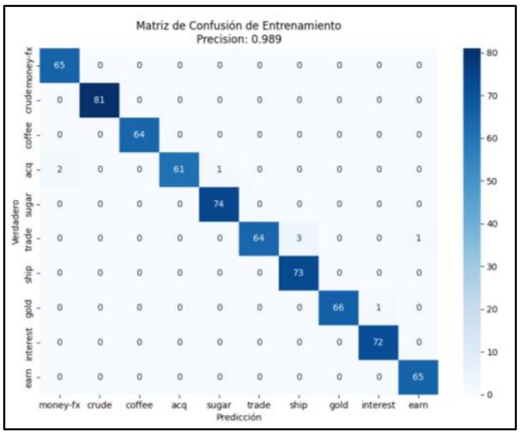


Fig. 3. Confusion Matrix for Training, Scenario 1

2. Results for Scenario 2.

In this scenario, the texts were subjected to the preprocessing technique of tokenization and were also converted to lowercase. The confusion matrix is presented in Fig.4, and the additional evaluation metrics are detailed in Table II.

3. Results for Scenario 3.

In this scenario, the texts were subjected to preprocessing techniques including tokenization, conversion to lowercase, and the removal of stop-words. The confusion matrix is illustrated in Fig.5, and the other evaluation metrics are detailed in Table II.

4. Results for Scenario 4, During Training:

In this scenario, the texts underwent multiple preprocessing techniques: tokenization, conversion to lowercase, stop-word removal, and the elimination of words with a frequency of 1 in the documents. The confusion matrix is presented in Fig.6, and the additional evaluation metrics are detailed in Table II.

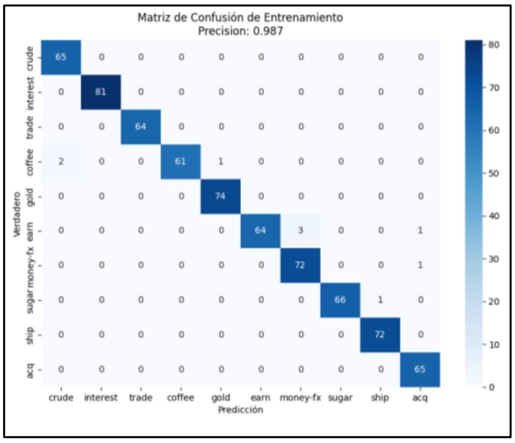


Fig. 4. Confusion Matrix for Training, Scenario 2

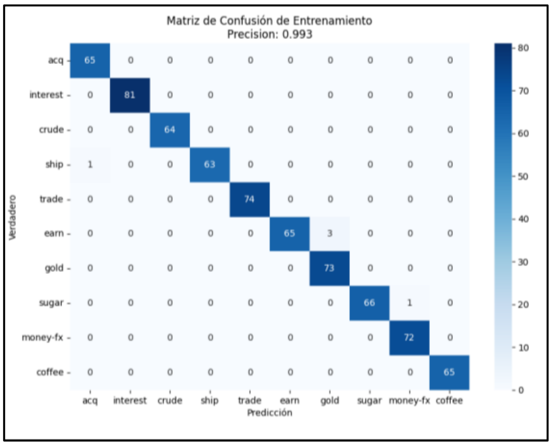


Fig. 5. Confusion Matrix for Training, Scenario 3

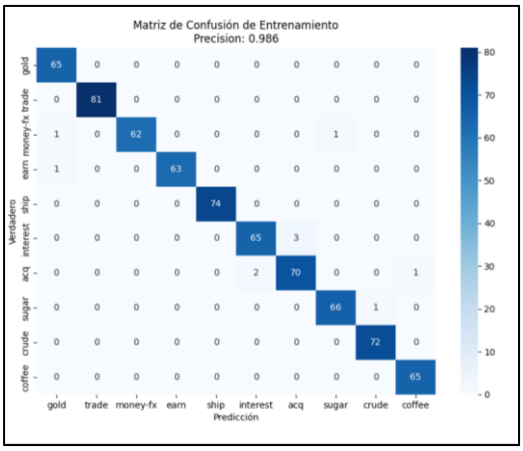


Fig. 6. Confusion Matrix for Training, Scenario 4

TABLE II. EVALUATION METRICS FOR TRAINING

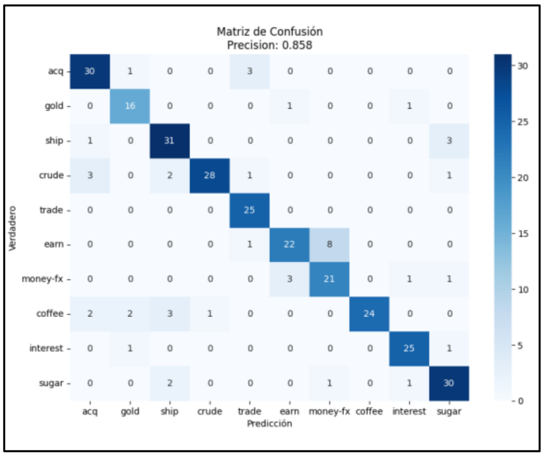
Scenario	Precision	Recall	F1_score
1	0.989	0.988	0.988
2	0.987	0.985	0.984
3	0.993	0.993	0.993
4	0.986	0.982	0.984

In Table II, the results obtained during the training phase for the evaluation metrics of accuracy, recall, and f1-score are displayed for each of the four scenarios of this study. Notably, Scenario 1 starts with an accuracy of 98.9%, while in Scenario 3, an accuracy of 99.3% is reached.

B. *Model Validation Results:*

Upon training the model, we proceed to validate it. For this purpose, the 30% of documents initially set aside—unfamiliar to the model—are used for validation. The objective is to observe the model's accuracy in predicting the category names of new documents (those in the test set), which are unknown to the model as they were not used during training.

1. Scenario 1 Results:
Within this scenario, only the tokenization preprocessing technique was applied to the texts. Refer to Fig.7 for the confusion matrix and Table III for additional evaluation metrics.



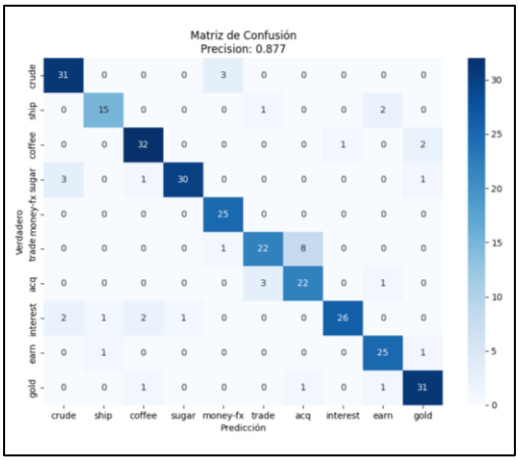
	acq	gold	ship	crude	trade	earn	money-fx	coffee	interest	sugar
acq	30	1	0	0	3	0	0	0	0	0
gold	0	16	0	0	0	1	0	0	1	0
ship	1	0	31	0	0	0	0	0	0	3
crude	3	0	2	28	1	0	0	0	0	1
trade	0	0	0	0	25	0	0	0	0	0
earn	0	0	0	0	1	22	8	0	0	0
money-fx	0	0	0	0	0	3	21	0	1	1
coffee	2	2	3	1	0	0	0	24	0	0
interest	0	1	0	0	0	0	0	0	25	1
sugar	0	0	2	0	0	0	1	0	1	30

Fig. 7. Confusion Matrix for Validation, Scenario 1

2. Scenario 2 Results:
For this scenario, the texts underwent two preprocessing techniques: tokenization and conversion to lowercase. The confusion matrix is depicted in Fig.8, while Table III provides a detailed account of other evaluation metrics.

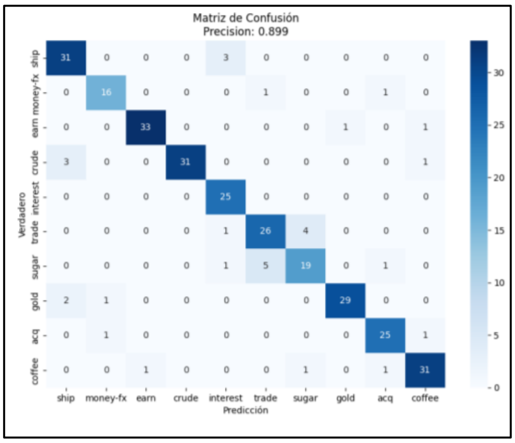
3. Scenario 3 Results:
In this context, the texts experienced a series of preprocessing actions, which included tokenization, transformation to lowercase, and stop-word removal. Fig.9 illustrates the confusion matrix, and further evaluation metrics can be consulted in Table III.

4. Scenario 4 Results:
This scenario saw the texts being processed through a comprehensive set of techniques: tokenization, lowercase conversion, stop-word elimination, and the removal of words appearing only once across the documents. The resulting confusion matrix is showcased in Fig.10, with additional evaluation metrics enumerated in Table III.



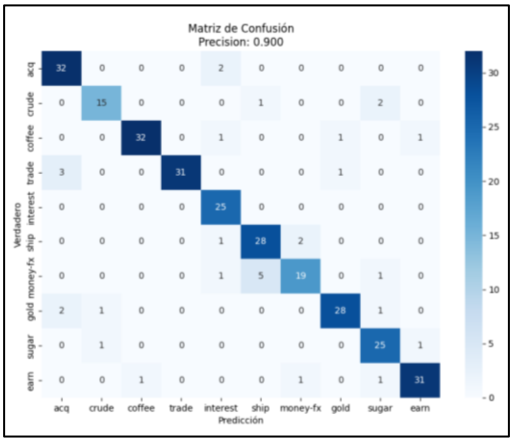
	crude	ship	coffee	sugar	money-fx	trade	acq	interest	earn	gold
crude	31	0	0	0	3	0	0	0	0	0
ship	0	15	0	0	0	1	0	0	2	0
coffee	0	0	32	0	0	0	0	1	0	2
sugar	3	0	1	30	0	0	0	0	0	1
money-fx	0	0	0	0	25	0	0	0	0	0
trade	0	0	0	0	1	22	8	0	0	0
acq	0	0	0	0	0	3	22	0	1	0
interest	2	1	2	1	0	0	0	26	0	0
earn	0	1	0	0	0	0	0	0	25	1
gold	0	0	1	0	0	0	1	0	1	31

Fig. 8. Confusion Matrix for Validation, Scenario 2



	ship	money-fx	earn	crude	interest	trade	sugar	gold	acq	coffee
ship	31	0	0	0	3	0	0	0	0	0
money-fx	0	16	0	0	0	1	0	0	1	0
earn	0	0	33	0	0	0	0	1	0	1
crude	3	0	0	31	0	0	0	0	0	1
interest	0	0	0	0	25	0	0	0	0	0
trade	0	0	0	0	1	26	4	0	0	0
sugar	0	0	0	0	1	5	19	0	1	0
gold	2	1	0	0	0	0	0	29	0	0
acq	0	1	0	0	0	0	0	0	25	1
coffee	0	0	1	0	0	0	1	0	1	31

Fig. 9. Confusion Matrix for Validation, Scenario 3



	acq	crude	coffee	trade	interest	ship	money-fx	gold	sugar	earn
acq	32	0	0	0	2	0	0	0	0	0
crude	0	15	0	0	0	1	0	0	2	0
coffee	0	0	32	0	1	0	0	1	0	1
trade	3	0	0	31	0	0	0	1	0	0
interest	0	0	0	0	25	0	0	0	0	0
ship	0	0	0	0	1	28	2	0	0	0
money-fx	0	0	0	0	1	5	19	0	1	0
gold	2	1	0	0	0	0	0	28	1	0
sugar	0	1	0	0	0	0	0	0	25	1
earn	0	0	1	0	0	0	1	0	1	31

Fig. 10. Confusion Matrix for Validation, Scenario 4

TABLE III. EVALUATION METRICS FOR VALIDATION

Scenario	Precision	Recall	F1_score
1	0.858	0.854	0.849
2	0.877	0.872	0.87
3	0.899	0.895	0.894
4	0.9	0.892	0.891

In Table III, the results obtained during validation for the evaluation metrics of accuracy, recall, and f1-score are presented for each of the four scenarios of this study. As can be observed, Scenario 1 starts with an accuracy of 85.8%, and by Scenario 4, an accuracy of 90% is achieved.

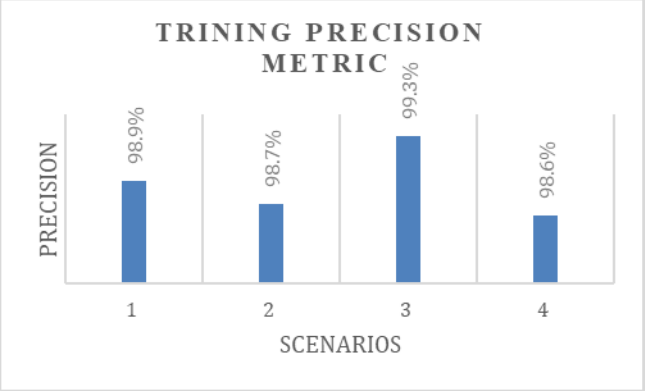


Fig. 11. Trining Precision Metric

In Fig. 11, the precision metric is displayed for each scenario during the model's training phase. The precision values are notably high. This elevated precision is attributed to the fact that, during training, the supervised machine learning algorithm —in this instance, Naive Bayes— employs the same data subset for both its training and validation. Specifically, when trained with 70% of the entire dataset, the algorithm is already familiar with the documents used during this phase, resulting in a high rate of accurate predictions.

However, the true indicator of the algorithm's learning capability is observed in the validation phase. At this stage, entirely new and unknown documents are introduced to the model for category prediction. The outcomes of this validation can be seen in Fig. 12.

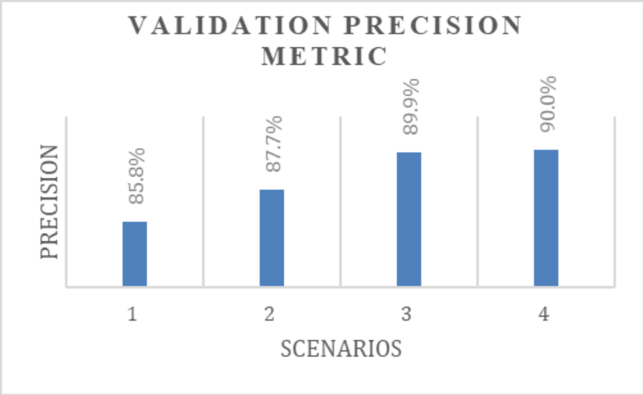


Fig. 12. Validation Precision Metric

In Fig. 12, the results pertaining to the precision metric during the training of the automatic text classification model for various proposed scenarios are presented. The first scenario focuses on a text that is solely tokenized; the second involves the tokenized text that is subsequently converted to lowercase; in the third scenario, in addition to tokenizing and converting the text to lowercase, stopwords are removed. Lastly, the fourth scenario encompasses tokenization, conversion to lowercase, removal of stopwords, and elimination of words that appear only once in the documents. The latter aren't pivotal for categorization, but their removal aids in reducing the dimensionality of the vectorization matrix. This, in turn, streamlines the computational resource requirements for the classification task, leading to enhanced outcomes.

It is worth emphasizing that Fig.12 vividly demonstrates the significant influence of text preprocessing on the precision metric. A rising trend in precision is observed as the preprocessing level intensifies. The results commence with a precision of 85.8%, followed by 87.7%, then 89.9%, and ultimately peaking at 90%.

It is worth noting that other evaluation metrics, such as Recall and F1-score, exhibit behavior analogous to that of precision, as depicted in Table III. Hence, the decision was made to emphasize solely the precision metric in Fig. 11 and Fig.12.

It is imperative to point out that in the deployment of machine learning algorithms, the volume of training data can either positively or adversely influence the outcomes. For this study, we utilized 99 documents per category, as the Reuters 21578 dataset allowed for this volume for an analysis involving 10 categories.

In conclusion, preprocessing techniques play a pivotal role in machine learning. These methodologies focus on cleaning, transforming, and tailoring the data, retaining only the most salient information, thereby enabling the learning algorithm to discern the defining features of each category more effectively.

V. CONCLUTIONS

In the domain of Natural Language Processing (NLP), automatic text classification stands as a cornerstone, having undergone profound advancements over recent decades. Central to these developments are the methodologies and techniques designed to bolster classification accuracy, offering immense potential for diverse applications in our digital era. The present study delved into this intricate challenge, incorporating a gamut of preprocessing techniques to the esteemed Reuters-21578 corpus and leveraging the capabilities of the Naive Bayes classification paradigm. Through a rigorously defined methodological framework, the investigation not only sought to amplify the precision of text categorization but also to accentuate the pivotal role of preprocessing within machine learning paradigms. The salient conclusions derived from this inquiry include:

Framework Integrity: The meticulously crafted methodology, spanning from inception to culmination, ensured a comprehensive assessment of automated text classification dynamics.

Corpus Relevance: Harnessing the Reuters-21578 Corpus, a gold standard in NLP literature, endowed the study with robust analytical credibility.

Preprocessing Vitality: The study elucidated the paramountcy of preprocessing in optimizing text classification outcomes. Its multifaceted preprocessing approach, ranging from rudimentary tokenization to nuanced strategies like stop-word elimination and frequency filtration, showcased a graduated enhancement in classification precision.

Model Proficiency: The Naive Bayes paradigm emerged as a judicious selection for the given classification endeavor, exhibiting stellar efficacy with training accuracy peaking at 99.3%. While the validation metrics trailed the training ones, they still presented an impressive accuracy of up to 90%.

Analytical Scenarios: The delineated scenarios manifest that refined preprocessing strategies correspond to augmented model efficacy. Particularly, Scenario 3, amalgamating tokenization, lowercasing, and stop-word omission, achieved a zenith of 99.3% training accuracy and 89.9% during validation, a benchmark marginally surpassed in Scenario 4, hinting at the law of diminishing returns tied to over-processing.

Metrics Concordance: A harmony observed amongst diverse evaluation metrics (Precision, Recall, and F1-score) across scenarios underscored the model's unwavering robustness, augmenting the confidence in the deduced outcomes.

Influence of Data Volume: The strategic selection of 99 documents per category struck an optimal balance for both training and validation phases. The investigation hinted at the intricate interplay between data volume and model proficiency, illuminating the perpetual equilibrium sought between data scale and performance optimization.

Forward-Looking Considerations: The inquiry underscores the weightage of preprocessing within machine learning and NLP terrains. Subsequent research can venture into more intricate preprocessing methodologies or juxtapose alternative machine learning paradigms to gauge relative efficiencies.

Concluding Insights: The study reaffirms the dual importance of machine learning algorithms and data preprocessing. Harmonizing these elements is the key to unlocking unparalleled text classification prowess.

References

[1] R. E. Lopez condori and J. L. Tejada Cárcamo, “Método de Clasificación Automática de Textos basado en Palabras Claves utilizando Información Semántica: Aplicación a Historias Clínicas,” Universidad Nacional de San Agustín, 2014. Accessed: Oct. 18, 2022. [Online]. Available: https://roquelopez.com/resource/publications/Lopez_UndergraduateThesis.pdf

[2] 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

[3] J. M. Duarte and L. Berton, “A review of semi-supervised learning for text classification,” *Artif Intell Rev*, vol. 56, no. 9, pp. 9401–9469, Sep. 2023, doi: 10.1007/s10462-023-10393-8.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] B. Zhang, “News Text Classification Algorithm Based on Machine Learning Technology,” in *2022 International Conference on Education, Network and Information Technology (ICENIT)*, IEEE, Sep. 2022, pp. 182–186. doi: 10.1109/ICENIT57306.2022.00047.

[9] R. Venegas, “Clasificación de textos académicos en función de su contenido léxico-semántico,” *Revista signos*, vol. 40, no. 63, pp. 239–271, 2007, doi: 10.4067/S0718-09342007000100012.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] A. M. Aubaid and A. Mishra, “A rule-based approach to embedding techniques for text document classification,” *Applied Sciences (Switzerland)*, vol. 10, no. 11, Jun. 2020, doi: 10.3390/app10114009.

[16] H.-T. Duong and T.-A. Nguyen-Thi, “A review: preprocessing techniques and data augmentation for sentiment analysis,” *Comput Soc Netw*, vol. 8, no. 1, p. 1, Dec. 2021, doi: 10.1186/s40649-020-00080-x.

[17] K. Smelyakov, D. Karachevtsev, D. Kulemza, Y. Samoilenko, O. Patlan, and A. Chupryna, “Effectiveness of Preprocessing Algorithms for Natural Language Processing Applications,” in *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*, IEEE, Oct. 2020, pp. 187–191. doi: 10.1109/PICST51311.2020.9467919.

[18] 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.

[19] J. J. Paniagua and R. Guzman-Cabrera, “Clasificación automática de documentos utilizando aprendizaje automático y Reuters-21578,” YURIRIA, Jan. 2022. [Online]. Available: https://www.researchgate.net/publication/357860387

[20] 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.

[21] 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

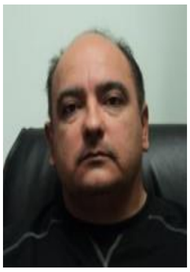
[22] R. Mosquera, O. D. Castrillón, and L. Parra, “Support vector machines, Naïve Bayes classifier and genetic algorithms for the prediction of psychosocial risks in teachers of Colombian public schools.,” *Informacion Tecnologica*, vol. 29, no. 6, pp. 153–162, Dec. 2018, doi: 10.4067/S0718-07642018000600153.

[23] M. M. Hijazi, A. Zeki, and A. Ismail, “A Review Study on Arabic Text Classification,” in *2022 International Arab Conference on Information Technology (ACIT)*, IEEE, Nov. 2022, pp. 1–13. doi: 10.1109/ACIT57182.2022.9994124.

[24] 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.

BIOGRAFIAS

JOSE ARENGAS ACOSTA Student of the Master's in Technology Management in the Department of Multidisciplinary Studies of Yuriria from the Engineering Division of the Irapuato-Salamanca Campus of the University of Guanajuato, Mexico. Electrical Engineer graduated from the Industrial University of Santander, Colombia.



RAFAEL GUZMÁN CABRERA Full Professor in the Department of Electrical Engineering of the Engineering Division at the Irapuato-Salamanca Campus of the University of Guanajuato for 23 years, Ph.D. in Pattern Recognition and Artificial Intelligence from the Polytechnic University of Valencia, Spain. Member of the Mexican Academy of Sciences, SNI-1. Member of the academic body of applied physics and advanced technologies.

Misael López Ramírez He received his Doctorate and Master's degree in electrical engineering from the University of Guanajuato, Guanajuato, Mexico, in 2017 and 2013, respectively, and completed a postdoctoral stay at the Technological Institute of Aguascalientes, Mexico, in 2019. He joined the Engineering Division of the Irapuato Salamanca Campus of the University of Guanajuato at the end of 2019. He is recognized as a Level I member of the National System of Researchers (SNI-I) by the National Council of Researchers (CONAHCyT). His current research and interests include digital image and signal processing, power quality, intelligent systems, machine learning, and digital systems applied to industry