UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Posgrados

Detección de comentarios acusatorios con indicios de corrupción en los procesos de compras públicas de Ecuador utilizando Procesamiento de Lenguaje Natural

Proyecto de Titulación

Francisco Roh López

Felipe Grijalva, Ph.D.

Director de Trabajo de Titulación

Trabajo de titulación de posgrado presentado como requisito para la obtención del título de Magíster en Inteligencia Artificial

Quito, 01 de diciembre de 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE POSGRADOS

HOJA DE APROBACIÓN DE TRABAJO DE TITULACIÓN

Detección de comentarios acusatorios con indicios de corrupción en los procesos de compras públicas de Ecuador utilizando Procesamiento de Lenguaje Natural

Francisco Roh López

Nombre del Director del Programa:	Felipe Grijalva
Título académico:	Ph.D. en Ingeniería Eléctrica
Director del programa de:	Inteligencia Artificial

Nombre del Decano del colegio Académico:	Eduardo Alba
Título académico:	Doctor en Ciencias Matemáticas
Decano del Colegio:	Ciencias e Ingenierías

Nombre del Decano del Colegio de Posgrados:	Dario Niebieskikwiat
Título académico:	Doctor en Física

Quito, diciembre 2024

© DERECHOS DE AUTOR

Nombre del estudiante: Francisco Roh López

Código de estudiante: 00338904

C.I.: 050171816-7

Lugar y fecha: Quito, 01 de Diciembre de 2024.

ACLARACIÓN PARA PUBLICACIÓN

UNPUBLISHED DOCUMENT

DEDICATORIA

A mi madre, por su apoyo constante e incondicional

AGRADECIMIENTOS

Quiero expresar mi sincero agradecimiento a los profesores de este posgrado, quienes participaron activamente en el desarrollo de este trabajo, aportando su conocimiento y experiencia de manera invaluable.

Mi especial gratitud al tutor de este trabajo de titulación, Felipe Grijalva, quien desde el inicio de este camino ha estado siempre presente, brindando una guía oportuna y mostrando un constante compromiso por el avance de este proyecto.

Agradezco también al proyecto Kapak, que ha permitido el acceso al portal de indicadores e información relacionada, que facilita la detección de riesgos de corrupción en los procesos de compras públicas a partir de los datos transparentados por el Servicio Nacional de Contratación Pública (SERCOP) mediante el Sistema Oficial de Contratación Pública del Ecuador (SOCE).

Finalmente, agradezco a los Poligrants 24261 y 24270, cuyo apoyo ha sido esencial para la ejecución de este trabajo.

RESUMEN

En los procesos de contratación pública, toda la información relacionada es de importancia para valorar su validez y transparencia: actores involucrados, montos, relaciones, cantidades, especificaciones técnicas, entre otros. Dentro de ésta, los comentarios escritos por los participantes durante los procesos representan una fuente invaluable y complementaria de información para identificar posibles indicios de corrupción. Sin embargo, la complejidad del lenguaje natural, la interpretación de comentarios no estructurados que incluyen preguntas, opiniones, quejas, acusaciones entre otros, y la necesidad de procesar grandes volúmenes de datos, hacen indispensable el uso de técnicas de Inteligencia Artificial y Procesamiento del Lenguaje Natural (PLN) para identificar de manera oportuna y eficiente posibles anomalías dentro de esta gestión pública.

Este documento analiza los comentarios de los procesos de contratación pública en el Sistema Oficial de Contratación Pública del Ecuador (SOCE) utilizando modelos de aprendizaje supervisado tradicionales (Regresión Logística, Naive Bayes, Clasificador de Bosque Aleatorio y SVC) junto con representaciones de texto convencionales (BoW y TF-IDF) y nuevos métodos de representación: embeddings entrenados como Doc2Vec, embeddings preentrenados como FastText y los más avanzados como SBERT y el reciente text-embedding-3-large. También incorpora modelos supervisados profundos, incluidos transformers (BERT y RoBERTa) y modelos RNN (LSTM y GRU), con el objetivo de detectar si los comentarios sugieren indicios de prácticas corruptas o riesgos de corrupción, permitiendo así a los especialistas realizar análisis dirigidos hacia contratos de alto riesgo y desarrollar indicadores pertinentes.

El estudio presenta una comparativa de múltiples esquemas de aumento de datos, balanceo de datos y preprocesamiento en modelos tradicionales y profundos, priorizando la eficiencia en recursos para implementar alertas tempranas de forma efectiva, escalable y con bajos costos operativos. Los resultados obtenidos resaltan la importancia del data augmentation para mejorar el rendimiento de los modelos en datasets desbalanceados, el rendimiento balanceado de los modelos basados en redes neuronales, se enfatiza la importancia de las representaciones semánticas avanzadas (SBERT y text-embedding-3-large) para detectar patrones complejos mediante modelos tradicionales, y se identifican combinaciones prometedoras de métodos convencionales que ofrecen buenos resultados (como Random Forest Classifier con TF-IDF).

Este trabajo sienta las bases para el desarrollo de sistemas de monitoreo predictivo en tiempo real, que combinen eficiencia en el uso de recursos con un alto rendimiento predictivo. Al integrar técnicas de Inteligencia Artificial con herramientas de PLN, se abre la posibilidad de fomentar la transparencia en los procesos de contratación pública, identificar riesgos y patrones anómalos de manera proactiva, y fortalecer la supervisión ciudadana inteligente como parte de los esfuerzos para combatir la corrupción en la gestión pública.

Palabras clave: textos, lenguaje, comentarios, clasificación, aprendizaje automático, aprendizaje profundo, modelos supervisados, aumento de datos, corrupción

ABSTRACT

In public procurement processes, all related information is critical for evaluating their validity and transparency: involved stakeholders, amounts, relationships, quantities, technical specifications, among others. Among these, the comments written by participants during the processes represent an invaluable and complementary source of information for identifying potential signs of corruption. However, the complexity of natural language, the interpretation of unstructured comments—including questions, opinions, complaints, accusations, and more—and the need to process large volumes of data make it essential to use Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques to identify potential anomalies in public management both efficiently and promptly.

This paper analyzes comments from public procurement processes within Ecuador's Official Public Procurement System (SOCE) using traditional supervised learning models (Logistic Regression, Naive Bayes, Random Forest Classifier, and SVC) in combination with conventional text representations (BoW and TF-IDF) and newer representation methods: trained embeddings such as Doc2Vec, pre-trained embeddings such as FastText, and more advanced ones like SBERT and the recent text-embedding-3-large. It also incorporates deep supervised models, including transformers (BERT and RoBERTa) and RNN models (LSTM and GRU), aiming to detect whether the comments indicate corrupt practices or corruption risks, thereby enabling specialists to focus their analyses on high-risk contracts and develop pertinent indicators.

The study presents a comparative analysis of multiple data augmentation schemes, data balancing techniques, and preprocessing strategies in traditional and deep models, prioritizing resource efficiency to implement scalable, effective early-warning systems with low operational costs. The results highlight the importance of data augmentation in improving model performance on imbalanced datasets, the balanced performance of neural network-based models, the relevance of advanced semantic representations (SBERT and text-embedding-3-large) in detecting complex patterns using traditional models, and the identification of promising combinations of conventional methods that deliver good results (such as Random Forest Classifier with TF-IDF).

This work lays the groundwork for developing real-time predictive monitoring systems that combine resource efficiency with high predictive performance. By integrating AI techniques with NLP tools, it becomes possible to promote transparency in public procurement processes, proactively identify risks and anomalous patterns, and strengthen smart citizen oversight as part of the broader efforts to combat corruption in public administration.

Key words: texts, language, comments, classification, machine learning, deep learning, supervised models, data augmentation, corruption

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Detection of Accusatory Comments in Ecuadorian Public Procurement Processes using Natural Language Processing.

Francisco Roh, *Member, IEEE,* Felipe Grijalva, *Senior Member, IEEE*

***Abstract*—In public procurement processes, all related information is critical for evaluating their validity and transparency: involved stakeholders, amounts, relationships, quantities, technical specifications, among others. Among these, the comments written by participants during the processes represent an invaluable and complementary source of information for identifying potential signs of corruption. However, the complexity of natural language, the interpretation of unstructured comments—including questions, opinions, complaints, accusations, and more—and the need to process large volumes of data make it essential to use Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques to identify potential anomalies in public management both efficiently and promptly.**

This paper analyzes comments from public procurement processes within Ecuador's Official Public Procurement System (SOCE) using traditional supervised learning models (Logistic Regression, Naive Bayes, Random Forest Classifier, and SVC) in combination with conventional text representations (BoW and TF-IDF) and newer representation methods: trained embeddings such as Doc2Vec, pre-trained embeddings such as FastText, and more advanced ones like SBERT and the recent text-embedding-3-large. It also incorporates deep supervised models, including transformers (BERT and RoBERTa) and RNN models (LSTM and GRU), aiming to detect whether the comments indicate corrupt practices or corruption risks, thereby enabling specialists to focus their analyses on high-risk contracts and develop pertinent indicators.

The study presents a comparative analysis of multiple data augmentation schemes, data balancing techniques, and preprocessing strategies in traditional and deep models, prioritizing resource efficiency to implement scalable, effective early-warning systems with low operational costs. The results highlight the importance of data augmentation in improving model performance on imbalanced datasets, the balanced performance of neural network-based models, the relevance of advanced semantic representations (SBERT and text-embedding-3-large) in detecting complex patterns using traditional models, and the identification of promising combinations of conventional methods that deliver good results (such as Random Forest Classifier with TF-IDF).

This work lays the groundwork for developing real-time predictive monitoring systems that combine resource efficiency with high predictive performance. By integrating AI techniques with NLP tools, it becomes possible to promote transparency in public procurement processes, proactively identify risks and anomalous patterns, and strengthen smart citizen oversight as part of the broader efforts to combat corruption in public administration.

***Index Terms*—texts, language, comments, classification, machine learning, deep learning, supervised models, data augmentation, corruption**

I. Introduction

Currently, Universidad San Francisco de Quito has developed Kapak, a platform aimed at improving transparency in public procurement in Ecuador. The project, led by the College of Jurisprudence and the Polytechnic College of USFQ.[1]

Kapak is a software application that leverages data science and artificial intelligence to combat corruption in Ecuador's public procurement processes[2]. Operating independently of the government, Kapak extracts data directly from the state's official procurement system (SOCE) rather than from public portals. This approach ensures a reliable and sustainable technological solution for monitoring corruption risks over time.[1], [3], [2]

The development of Kapak was supported by the technical assistance of the German Cooperation GIZ's Ecuador SinCero program, which aims to create conditions for preventing corruption in line with international standards. The project began with a diagnostic phase that involved reviewing the SOCE to identify available data variables for Electronic Reverse Auction (SIE) and Specific Business Turnaround (GEN) processes. This evaluation highlighted that the SOCE does not follow best practices in publishing open procurement data.[1], [3], [2]

To address these shortcomings, the team researched national and international platforms recognized for transparency in open data to model Kapak accordingly. They also reviewed literature on risk indicators in public procurement for potential corruption cases, which informed the scope of the project and the choice of technological solutions.[1], [3], [2]

The final steps involved formulating corruption risk indicators and engaging in collaborative activities to create new ones. Indicators were selected based on their calculability using available SOCE data. An information collector—a

F. Roh and F. Grijalva are with Universidad San Francisco de Quito USFQ

web crawler employing data scraping techniques—was developed to extract information from the system, a strategy approved by SOCE officials. The collected data is stored in a centralized database and undergoes preprocessing for homogenization and cleaning[4].

Kapak calculates individual corruption risk indicators at three levels: procurement process, contracting entity, and supplier, specifically for the SIE and GEN modalities. For GEN procedures, the types of procurement considered include public companies, mercantile or subsidiary companies, and publication processes. From these individual indicators, six composite corruption risk indices are generated: SIE-process, SIE-entity, SIE-supplier, GEN-process, GEN-entity, and GEN-supplier. The results are presented through a public web portal that displays methodologies, results, and visualizations of the composite indices. By detecting suspicious procedures, evaluating contracting entities and suppliers, and raising awareness about corruption risks, Kapak empowers citizens and enhances transparency in a historically under-supervised area of government. The ultimate goal is to integrate advanced algorithms to improve monitoring and provide early warnings of potential irregularities.[1], [3], [2]

This study focuses on the detection of accusatory comments added by participants in Ecuador's public procurement processes by utilizing comment data available in the Kapak procurement database and its labeling help. The main objective is to identify signs of corruption through the application of various traditional and deep learning models. We employ advanced natural language processing (NLP) techniques and text representations and data augmentation methods to build effective models capable of providing early warnings of potential irregularities.

mds

December 1, 2024

II. Prior works

The topic of transparency in public procurement in Ecuador has been a significant issue both nationally and internationally, dating back to times when computer science did not hold the relevance it does today. The classification of comments using Recurrent Neural Networks (RNNs) and transformers has been extensively researched due to their wide applications in Natural Language Processing (NLP). Early work in this area focused on RNNs and their variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) [5]. These architectures are capable of capturing long-term dependencies in text sequences, which is useful for tasks like sentiment classification, spam detection, and content moderation. However, RNNs face difficulties with long sequences and can be computationally intensive, limiting their scalability and efficiency.

The advent of text embeddings has profoundly transformed NLP by enabling the conversion of textual data into continuous vector representations that capture semantic and syntactic nuances. Early models like Word2Vec [6] and FastText [7] introduced efficient methods for generating word embeddings, allowing algorithms to comprehend semantic relationships between words through unsupervised learning on large corpora. However, these models were limited in capturing context-dependent meanings of words.

To address this limitation, contextualized word embedding models like ELMo [8] emerged, generating word representations that vary according to their contextual usage. The introduction of the Transformer architecture [9] further revolutionized the field by employing self-attention mechanisms, leading to models such as BERT [10] and RoBERTa [11]. These models produce deep bidirectional representations and have set new benchmarks across various NLP tasks by understanding context more effectively. The introduction of transformers marked a significant advance in comment classification, as they process sequences in parallel and capture long-term relationships more efficiently than RNNs. Moreover, transformers can be pre-trained on large amounts of data and then fine-tuned for specific tasks, significantly improving their performance in classification tasks. Current research focuses on optimizing these models and combining them with techniques such as data augmentation and transfer learning to improve precision and efficiency in comment classification.

More recently, large-scale pre-trained language models like GPT-3 [12] have been leveraged to generate high-quality text embeddings. OpenAI's development of the text-embedding-ada-002 model [13] represents a significant advancement, achieving excellent precision in capturing complex semantic relationships. This model facilitates a wide range of applications—from semantic search and sentiment analysis to text classification—by providing highly accurate and context-aware embeddings.

OpenAI has introduced more advanced embedding models, such as text-embedding-3-large, which offer significant improvements in performance and the ability to capture complex semantic relationships. These models enable more precise and efficient applications in NLP tasks, including comment classification. [14]

III. Materials and Methods

Methodology used is composed of three main stages [15], [10], [6]:

1) Dataset Exploration
2) Data Balancing and Data Augmentation
3) Machine Learning Models

These are explored below:

A. Dataset Exploration

A public procurement system comments database with 5,005 human-labeled binary records was used, resulting in an imbalanced dataset: 97% without signs of corruption

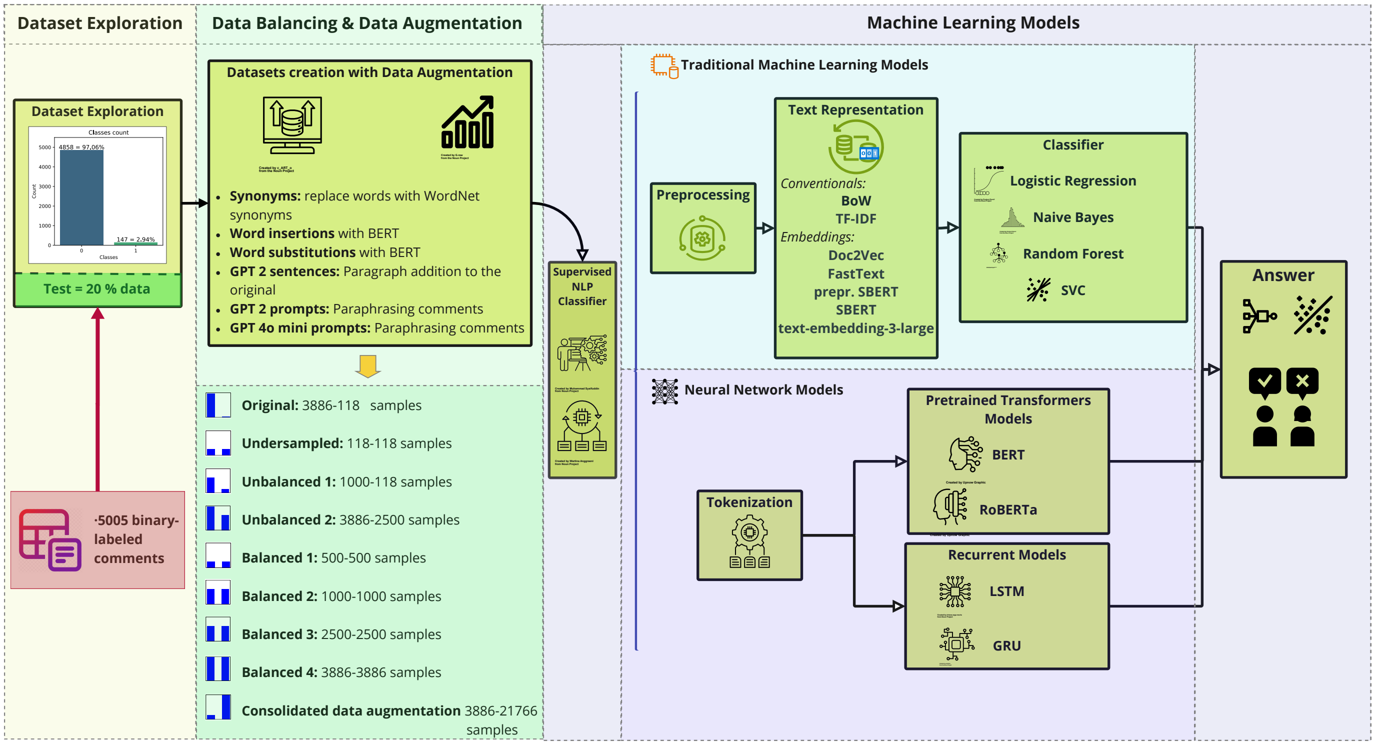


Figure 1. Block diagram of the proposed approach.

and 3% with examples of comments indicating corruption, according to classification specialists.

Currently, there is a large volume of comments available, but labeling capacity is limited, relying solely on the dataset described for this exercise [4].

In this stage, before any data manipulation, 20% of the data was randomly and stratified extracted due to its condition to collect the test dataset that would provide the analysis metrics for this exercise, considering that the data available is limited.

B. Data Balancing and Data Augmentation

Various data augmentation techniques were applied to increase the diversity of training data [16], [17].

For certain methods, auxiliary libraries such as **nlpaug** were used to ensure that generated comments maintained the original context [18]:

Synonyms: For each word, using WordNet, a synonym is searched. Words are replaced with their synonyms, ensuring that the grammatical structure of the text is preserved [19].

BERT: Insertion and substitution of words using BERT transformer. The original text is tokenized using the pre-trained model tokenizer to obtain contextual embeddings [10]. Positions where words can be inserted or substituted are identified, applied, and the modified text is reconstructed as output. Insertion and substitution are different methods that generate distinct but suitable results for this purpose [17].

Additionally, low (or no) cost LLMs were used for generating new data to facilitate model training [15]:

GPT-2: Via API, text was generated based on the context provided by the original comment using the open-source GPT-2 model, allowing the model to complement the comment with the aim of introducing noise while maintaining a relative meaning [20]. A specific prompt was also used to request a new comment similar to the original but employing different words and phrasing without redundancy, maintaining its meaning [15].

GPT-4o mini: The latest version from OpenAI, was used via API. Using a low-cost solution, it allowed the generation of new comments based on the context provided by a prompt with the required specifications and the original comment [14].

With the previously described techniques that ensure generated comments maintain the original context and thus increase model effectiveness, we created the set of datasets to be tested in machine learning models as follows:

Without using the described techniques and by random sampling, we obtained the following datasets:

Table I
Dataset Size by Class without Data Augmentation

Dataset Name	Size by Class
Original	3886-118 samples
Undersampled	118-118 samples
Unbalanced 1	1000-118 samples

Using all the described techniques, the following datasets were created for each of them:

Table II
Dataset Size by Class with Data Augmentation

Dataset Name	Size by Class
Unbalanced 2	3886-2500 samples
Balanced 1	500-500 samples
Balanced 2	1000-1000 samples
Balanced 3	2500-2500 samples
Balanced 4	3886-3886 samples

Finally, a last dataset was obtained that aggregated all data augmentation techniques along with the initial data. Considering that for GPT-4o mini, two versions were tested, all data augmentation methods were consolidated into the following dataset:

Table III
Dataset Size by Class with Consolidated Data Augmentation

Dataset Name	Size by Class
Consolidated Data Augmentation	3886-21766 samples

The application of these techniques and combinations allowed us to create 36 datasets with different data characteristics and balancing, which will be tested with 4 deep models, each with different parameters and functions, and 4 traditional machine learning models using 7 different text representation methods. This leads to testing 1,658 models.

C. Machine Learning Models

In this stage, various machine learning models will be applied to the datasets obtained in the previous step. Two types of models will be used:

1) Traditional Machine Learning models
2) Neural network-based models

1) Traditional Machine Learning Models: Before processing the comments with traditional machine learning models for classification, a preprocessing step was conducted. This step involved converting all text to lowercase, removing non-alphabetic characters, eliminating stopwords, and performing tokenization and lemmatization.

In natural language processing (NLP), effective text representation is crucial for the performance of machine learning models. Below is a detailed overview of both conventional and advanced text representation methods used:

Conventional Methods:

- *Bag of Words (BoW):* This approach represents a document as an unordered collection of words, disregarding grammar and word order. Each word is associated with its frequency of occurrence, forming a vector that describes the document. While simple and effective for basic tasks, BoW fails to capture semantic relationships between words. [21]

- *Term Frequency-Inverse Document Frequency (TF-IDF):* TF-IDF enhances BoW by weighting the frequency of words in a document relative to their frequency across the entire corpus. This highlights distinctive terms and reduces the influence of common words, providing a more informative text representation. [22]

Trained Embeddings:

- *Doc2Vec:* An extension of Word2Vec, Doc2Vec generates vectors representing entire documents instead of individual words. It captures the semantics of a document by considering the context of words, making it useful for tasks such as document classification and information retrieval [23].

Pre-trained Embeddings:

- *FastText:* Developed by Facebook, FastText improves upon Word2Vec by considering subword information, allowing it to handle rare or out-of-vocabulary words and capture morphological relationships. This is particularly beneficial for languages with rich morphology[7], [24]. We use a mean vector of paragraphs of 300 dimensions.

- *SBERT:* Sentence-BERT adapts BERT to produce sentence embeddings, enabling efficient comparison and retrieval of sentences. It is useful in tasks such as semantic search and sentence classification. SBERT generates embeddings of 768 dimensions, facilitating efficient comparison of sentences in lower-dimensional vector spaces [25]. Due to the characteristics of this text representation method, it was used with both the original comments (SBERT) and the preprocessed comments (prepSBERT).

- *text-embedding-3-large:* Developed by OpenAI, this model generates high-quality embeddings for text via API, facilitating tasks such as semantic search, classification, and sentiment analysis. It offers a dense and contextualized representation of text. text-embedding-3-large produces embeddings with up to 3,072 dimensions, allowing for a more detailed and nuanced representation of text compared to previous models [14]. It was used with original comments.

The traditional machine learning models tested for classifying comments, using each produced dataset and each form of text representation, are described in Table IV with their respective variable parameters that were optimized during training.

The models were trained on an A-100 machine with an 80GB GPU using the following parameters:

- **Cross-validation strategy:** StratifiedKFold combined with GridSearchCV.
- **Optimization metric:** F1-Score, while also calculating precision, accuracy, and recall.
- **Cross-validation splits:** 3 splits.
- **Parallel processing:** 8 jobs.

Additionally, confusion matrices, learning curves (train/validation) and ROC curves were generated to evaluate

Table IV
Traditional Machine Learning Models and Parameters for Optimization

Model	Parameters to Optimize
Logistic Regression	C: Regularization strength (values: 0.01, 1, 100) Penalty: Type of regularization (options: L2, ElasticNet) Solver: Optimization algorithm (option: saga)
Naive Bayes	*var_smoothing*: Portion of the largest variance of all features added to variances for stability (values: logarithmic space from 10^0 to 10^{-9})
Random Forest Classifier	*n_estimators*: Number of trees in the forest (values: 80, 200) *max_depth*: Maximum depth of the tree (values: None, 10, 30) *min_samples_leaf*: Minimum number of samples required to be at a leaf node (values: 1, 4)
Support Vector Machine (SVC)	C: Regularization parameter (values: 1, 100)

the performance of the models, along with the processing time taken to train each model.

2) Neural Network-based Models: Tested neural network-based models, along with their respective parameters and features, are detailed below. These consist of recurrent neural network models and pretrained transformers. Tokenization was prioritized using the specific methods provided by each model; if not available, standard methods were used to generate the required vocabulary (RNNs).

3) RNN – Bidirectional GRU (Gated Recurrent Unit): GRU networks are effective in sequence tasks and offer a simpler structure than LSTMs [26]. The parameters used were:

- **Number of classes:** 2
- **Tokenization:** *nltk.word_tokenize* for tokenizing inputs and building the vocabulary.
- **Embeddings:** *nn.Embedding(...)*
- **Embedding dimension:** 512
- **Hidden dimension:** 128
- **Number of layers:** 2
- **Dropout:** 0.3

4) RNN – Bidirectional LSTM (Long Short-Term Memory): LSTM networks specialize in capturing long-term dependencies and are useful for textual data where order is important [5]. The parameters used were:

- **Number of classes:** 2
- **Tokenization:** *nltk.word_tokenize* for tokenizing inputs and building the vocabulary.
- **Embeddings:** *nn.Embedding(...)*
- **Embedding dimension:** 512
- **Hidden dimension:** 128
- **Number of layers:** 2
- **Dropout:** 0.3

5) Transformers: BERT (Bidirectional Encoder Representations from Transformers): BERT is excellent for text classification tasks due to its deep understanding of language [10]. A fully connected layer was added for classification purposes.

- **Pre-trained model:** *dccuchile/ bert-base-spanish-wwm-cased*
- **Size:** Base model (12 layers, 768 hidden units, 12 attention heads)
- **Number of classes:** 2
- **Tokenization:** *BertTokenizer* for input tokenization to ensure proper word segmentation.
- **Embeddings:** Pre-trained embeddings integrated within BERT
- **Hidden dimension:** 768

Auto_BERT: The BERT model was also tested using *AutoModelForSequenceClassification.from_pretrained()*, a method for loading pre-trained models from Hugging Face optimized for sequence classification.

6) Transformers: RoBERTa (Robustly Optimized BERT Pretraining Approach): RoBERTa is similar to BERT but optimized in pretraining, benefiting from a larger corpus and specific adjustments [11].

- **Pre-trained model:** *PlanTL-GOB-ES/roberta-base-bne*
- **Size:** Base model (12 layers, 768 hidden units, 12 attention heads)
- **Number of classes:** 2
- **Tokenization:** *RobertaTokenizer*
- **Embeddings:** Pre-trained embeddings integrated within RoBERTa
- **Hidden dimension:** 768

Auto_RoBERTa: RoBERTa model was also tested using *AutoModelForSequenceClassification.from_pretrained()*, optimized for sequence classification.

The models were trained on an A-100 machine with an 80GB GPU using the following parameters:

- **Loss function:** Cross Entropy
- **Epochs:** 30
- **Batch size:** 32
- **Workers:** 4
- **Warmup:** One-sixth of the training steps
- **Optimization:**

  - **Transformers:** AdamW with a learning rate of 2×10^{-5} [27]
  - **RNNs:** Adam with a learning rate of 2×10^{-3}
- **Dropout:** 0.3 for RNNs
- **Metrics:** Precision, Recall, F1 Score, and Accuracy

Additionally, confusion matrices, learning curves (train/validation) and ROC curves were generated to evaluate model performance along with processing time for training each model.

BERT and RoBERTa models were trained both unfrozen and with 8 and 10 layers frozen. Unfrozen models and recurrent networks were trained with early stopping patience of 5 and 10, while models with frozen layers used a patience of 5.

A notable aspect of this exercise was structuring it using PyTorch Lightning for Neural Network-based Models, employing a single *Dataset* class, a *DataModule*, and a *LightningModule*, which facilitated iterative training across datasets and models, as well as evaluation and result storage. And, for Traditional Machine Learning Models, using Customized General Functions that train and evaluate all models and parameters. Code, files and instructions can be found on GitHub at: *<Github code site link>*.

IV. Results and Discussion

A model comparison was conducted under similar conditions, considering that when analyzing one dimension of the obtained results, the others are "balanced" or comparable along the same axis. The main findings include:

A. Patience in NN-Based Models (Figure 2)

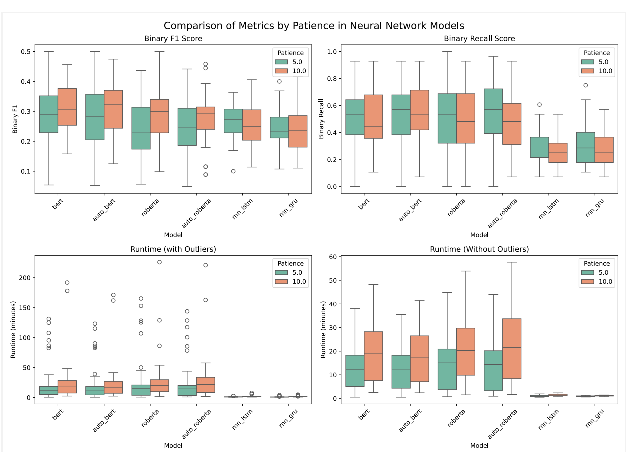


Figure 2. Metrics by patience in NN-models.

- *BERT* and *RoBERTa* exhibit higher runtime distributions compared to other models, suggesting greater computational complexity. Specifically, RoBERTa has the highest runtime under both patience conditions: 5 and 10 [28], [11].
- *LSTM and GRU architectures* consistently show lower runtimes, making them attractive for implementations where time and computational resources are limited [29]. However, it is crucial to evaluate if this efficiency compromises performance in terms of accuracy and other metrics.
- *BERT and Auto-BERT* show *superior performance* in both F1 and Recall, making them suitable for tasks requiring a balance between precision and true positive detection [28].
- *RoBERTa and Auto-RoBERTa models* have performance competitive with *BERT and Auto-BERT*, with slight variations.
- *RNN-based models (LSTM and GRU)* perform *lower* compared to Transformer-based models, though they could be useful in scenarios emphasizing simplicity and reduced runtime.
- Models with *patience of 10.0* (in orange) have slightly higher runtimes on average compared to those with a patience of 5.0 (in green). This is expected as a higher patience value allows more iterations before training stops [15].

To compare patience levels 5 and 10 in F1 Score, Recall, and Runtime metrics, a Shapiro-Wilk test was conducted to verify the normality of the differences for each patience setting across models. Results showed that these differences do not follow a normal distribution in any case, leading to Wilcoxon tests for patience 5 vs. 10, indicating that there is no significant difference in F1 score and recall; only runtime shows significant differences. Thus, using a patience of 5 is recommended to identify the best models in terms of the number of frozen layers to adjust for BERT and RoBERTa (Table V).

Table V
Shapiro-Wilk and Wilcoxon Test Results for F1 Score, Recall, and Runtime Differences of Patience 5 and 10 Neural Network-based Models

	Shapiro-Wilk Test			Wilcoxon Test		
	Statistic	p-value	Conclusion	w-statistic	p-value	Conclusion
F1 Score	0.9824	0.0102	Non-normal	10134.0	0.3984	No significant diff.
Recall	0.9141	7.36E-10	Non-normal	8865.0	0.7841	No significant diff.
Runtime	0.4985	2.21E-24	Non-normal	1544.0	1.88E-28	Significant diff.

Traditional machine learning models show shorter runtimes compared to neural network-based models, except for SVC models which induce high computational load, and, to a lesser extent, logistic regression models with BoW or TF-IDF due to their high dimensionality [29].

B. Performance of Pretrained Models with Frozen Layers (Figure 3)

- **BERT and RoBERTa** are sensitive to layer freezing, particularly in F1 Score and Recall. Leaving all layers trainable results in better performance for these models [28], [11].
- Freezing layers in these models degrades performance, especially in terms of F1 Score. To maximize perfor-

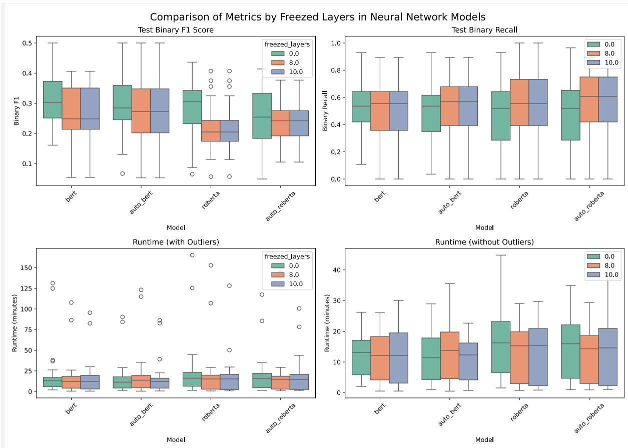


Figure 3. Metrics by freezed layers in Neural network models.

mance, not freezing or minimally freezing layers is recommended, depending on the specific model used.

To compare the effects of freezing 0, 8, and 10 layers on F1 Score, Recall, and Runtime, the Shapiro-Wilk test was conducted to verify normality. The paired t-test was used when differences followed a normal distribution, and the Wilcoxon test was applied otherwise. The tests show significant differences across all cases (Tables VI and VII).

Table VI
COMPARISON BETWEEN FREEZED LAYERS 0 AND 8 FOR F1 SCORE, RECALL, AND RUNTIME

	Shapiro-Wilk Test			t-Test / Wilcoxon Test		
	Statistic	**p-value**	**Conclusion**	**Statistic**	**p-value**	**Conclusion**
F1 Score	0.9863	0.2022	Normal	4.0250 (t)	9.51E-05	Significant diff.
Recall	0.9649	0.00095	Non-normal	2993.0 (W)	0.0198	Significant diff.
Runtime	0.7338	7.49E-15	Non-normal	3715.0 (W)	0.0027	Significant diff.

Table VII
COMPARISON BETWEEN FREEZED LAYERS 0 AND 10 FOR F1 SCORE, RECALL, AND RUNTIME

	Shapiro-Wilk Test			t-Test / Wilcoxon Test		
	Statistic	**p-value**	**Conclusion**	**Statistic**	**p-value**	**Conclusion**
F1 Score	0.9863	0.2022	Normal	4.0250 (t)	9.51E-05	Significant diff.
Recall	0.9649	0.00095	Non-normal	2993.0 (W)	0.0198	Significant diff.
Runtime	0.6072	5.27E-18	Non-normal	2944.0 (W)	5.65E-06	Significant diff.

C. Performance with Different Text Representations (Figure 4)

Text embeddings such as *text-embedding-3- large* and *SBERT* yield superior results compared to simpler representations like BoW, TF-IDF demostrates a great performance for its type, suggesting that these techniques better capture the semantic characteristics of the text [25], [30].

Deep neural network models like BERT and RoBERTa, even without using advanced text representations, generally show stable and superior performance compared to traditional models like Logistic Regression and Naive Bayes that rely on text representation.

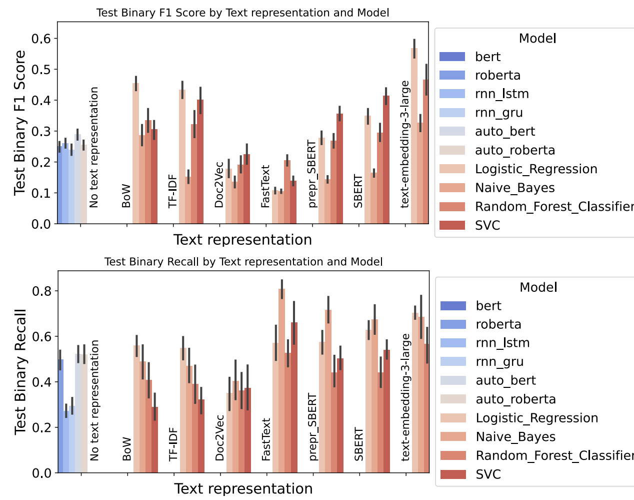


Figure 4. Test Binary F1 Score, Test Binary Recall by text representations and models.

D. Area Under the ROC Curve (AUC-ROC) with Different Text Representations (Figure 5)

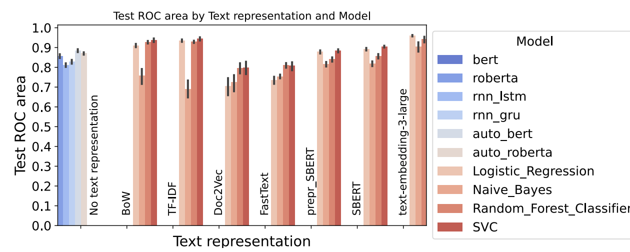


Figure 5. Test ROC area under curve by text representations and models.

The AUC-ROC metric reflects a model's ability to distinguish between classes, where a value of 1.0 indicates perfect classification and 0.5 is equivalent to random guessing [31]. Higher values indicate better classification performance.

The use of text embeddings (e.g., *text-embedding-3-large* and *SBERT*) enhances performance across most models, underscoring the importance of choosing representations that capture semantic meaning [25], [30].

Simpler text representations like BoW and TF-IDF have good performance with certain models, but with others like Naive Bayes perform worse, compared to embeddings like *text-embedding-3-large* and *SBERT*, which enable higher and more stable AUC-ROC scores.

Doc2Vec and FastText offer low-intermediate performance, with AUC-ROC varying depending on the applied model.

E. Performance Across Dataset Sizes (Figure 6)

Advanced augmentation techniques generated by GPT-4o mini-prompts and consolidated data augmentation show better F1 and Recall compared to simpler methods like original data or undersampling [15].

It was observed that combining various augmentation techniques, both standard and advanced, improves model

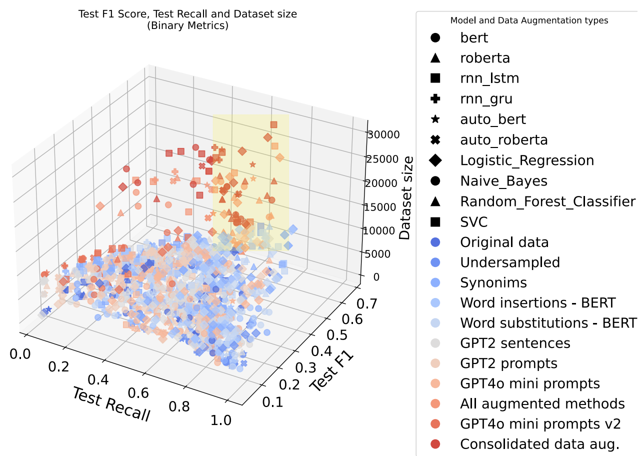


Figure 6. Test Binary F1 Score, Test Binary Recall and datasize of models by data augmentation.

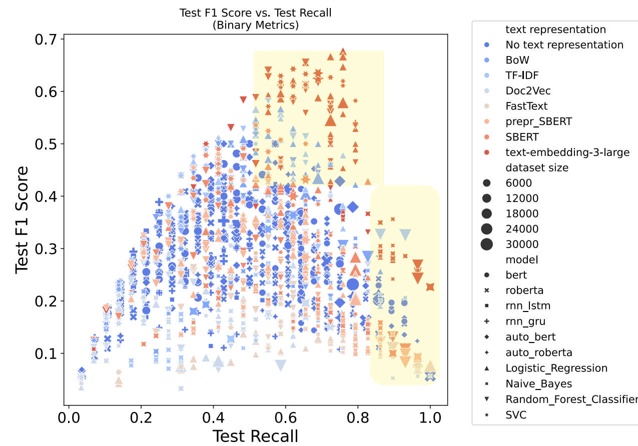


Figure 8. Test Macro Recall vs. Test Macro F1 Score of models.

performance. Additionally, as dataset size increases, models, particularly deep neural networks like BERT and RoBERTa, benefit more from larger datasets and data augmentation compared to simpler models [28], [11], [29].

RNN-based models (LSTM and GRU) do not benefit from dataset size in the same way that Transformer models do (e.g., BERT and RoBERTa).

F. Balancing F1 Score and Recall (Figures 7 and 8)

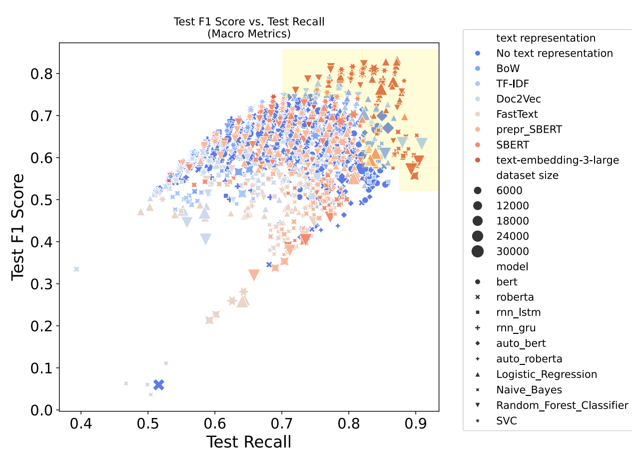


Figure 7. Test Binary Recall vs. Test Binary F1 Score of models.

Recall is the proportion of true positives detected by the model relative to all actual positives, measuring the model's ability to find all positive examples. It is critical when minimizing false negatives is essential, such as in detecting accusatory comments.

F1 Score, the harmonic mean of precision and recall, penalizes extreme values of either. It is particularly useful when balancing both metrics is desired instead of optimizing only one.

High *Recall* indicates that most positive examples are detected but may lack precision, leading to many false positives. On the other hand, high precision may result in low recall, omitting many positives. *F1 Score* helps balance these aspects, providing a comprehensive view of model performance.

Using both metrics allows for a more holistic assessment of a model's performance. *Recall* alone can lead to a model that detects many positives but with low precision, whereas *F1 Score* ensures the model detects relevant classes with acceptable precision and confidence.

Binary Metrics are primarily used in binary classification tasks, focusing on the model's performance in a specific detection task, such as recognizing an important class. These metrics provide a detailed analysis of the performance in a single relevant class (the "positive" class), facilitating understanding of how the model handles true positives and false negatives.

Macro Metrics average the performance of each class individually, giving equal weight to each class regardless of its frequency. They are useful for multi-class classification problems where evaluating the model's performance across all classes is important, ensuring that the model does not favor more frequent classes to the detriment of less represented ones. This is particularly important for imbalanced datasets.

With binary metrics, overall model performance is more varied. Models such as BERT and RNN (LSTM) show consistent scores across different parts of the graph.

Analyzing and reviewing the results, models like BERT, RoBERTa, and Auto-RoBERTa achieve balanced results across both metrics, but do not stand out with recall-oriented detection.

Text representations such as *text-embedding-3 - large* and *SBERT* show good performance for detecting the positive class.

Classic text representations like BoW and TF-IDF show a broader range of performance and generally lower compared to advanced embeddings.

The importance of having large, robust datasets and using data augmentation techniques is confirmed, as traditional models like Random Forest Classifier achieve good results under these conditions.

In lower-performing extremes (lower left), there is a variety of models and text representations that do not achieve high F1 scores or notable recall, indicating their inadequacy given the models and parameters used.

G. True Positives vs. False Positives (Figure 9)

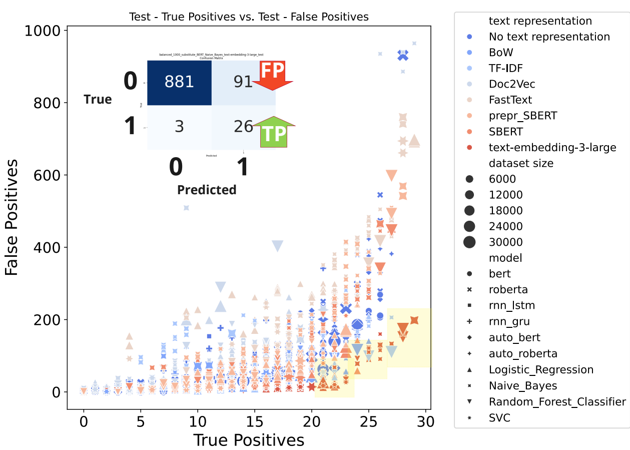


Figure 9. Test True positives vs. Test False positives by models and main features.

This analysis complements the observations from the *F1 Score* and *Recall graphs*, providing a more detailed view of how models handle false positives (FP) and true positives (TP), reinforcing the importance of utilizing both performance metrics and proper model configuration for binary classification applications.

The relationship between true positives (TP) and false positives (FP) in a binary classification context is fundamental for understanding the balance between a model's ability to correctly detect the positive class and the risk of generating false alarms. The combination of TP and FP helps evaluate the predictive power of the model. An ideal model would have a high number of TP and a low number of FP, indicating high sensitivity (recall) and high precision [31]. This relationship allows for model comparison and helps identify which models strike the best balance between accurately detecting positives without excessively increasing FP.

In the highlighted region (yellow quadrants), models that achieve a high number of TP without significantly increasing FP are primarily those utilizing modern and complex text representations, indicating that these embeddings provide better detection capabilities.

The representations of *BERT*, *RoBERTa*, and *Auto-RoBERTa* are prominently positioned in these areas of good performance but not in the best ranking.

The results suggest that models using advanced embeddings such as *text-embedding-3-large* and *SBERT* are more effective in terms of maximizing TP and minimizing FP, which translates into a good balance between *Recall* and precision [25], [30].

Larger points indicate that models trained on bigger datasets tend to perform better in terms of TP, suggesting that data volume significantly contributes to model performance [29].

H. General Model Review (Figure 10)

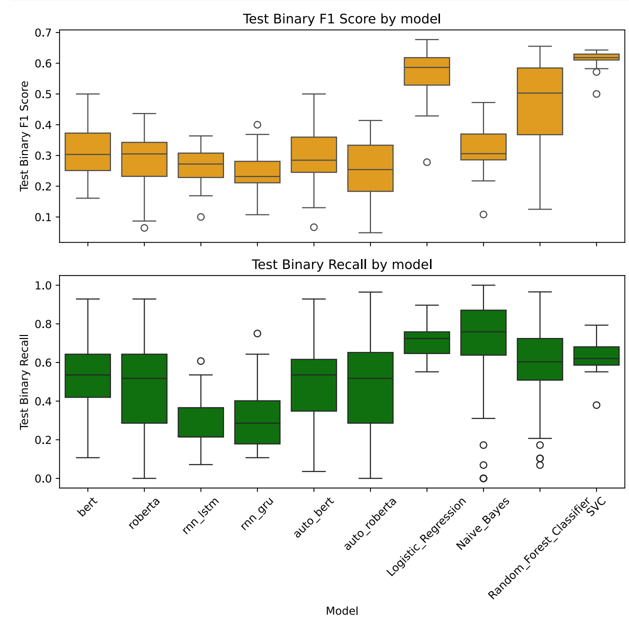


Figure 10. Models performance summary.

In summary, this figure shows that neural network-based models, particularly *BERT* and *RoBERTa*, tend to have acceptable performance in terms of *F1 Score* and *Recall* even without advanced text representations, compared to traditional models. This highlights the capacity of neural network models to work directly with text and effectively extract features [28], [11].

Neural network models like *BERT* and *RoBERTa*, whether with manual classifiers or using their automatic classification scheme (e.g., *Auto-BERT* and *Auto-RoBERTa*), do not generate the exact same results but yield fairly similar outcomes, suggesting their use depending on specific ease-of-use or requirements.

Traditional models show a higher dependence on the text representation technique used (BoW, TF-IDF, Doc2Vec, and more advanced embeddings such as *text-embedding-3-large* and *SBERT*) and display more variable performance. This difference is critical when considering model selection for text classification tasks where balancing precision and recall is essential for optimal performance.

V. Conclusions

1) **Impact of Data Augmentation on Imbalanced Datasets:** The use of data augmentation techniques

was shown to significantly improve model performance on imbalanced datasets. Models trained with augmented data exhibited greater detection capabilities and a better balance between *Recall* and *F1 Score*. This finding emphasizes the importance of expanding labeled data for training more robust models. Additionally, incorporating reinforcement learning methods could further enhance performance [15].

2) **Advantages of Advanced Embeddings:** Advanced embedding generation techniques, such as *text-embedding-3-large*, demonstrated superior performance when combined with traditional supervised models like Random Forest Classifier, logistic regression, and even Naive Bayes. While some of these techniques entail economic costs, they are scalable and manageable when processing large volumes of data compared to more expensive techniques, yielding considerable improvements in detection, precision, and consistency [25], [30]. With this alternatives, transfer learning is a good approach for good results with traditional models with low computational costs.
3) **Performance of Specific Models:** *BERT* model showed balance and stability between *Recall* and *F1 Score*, standing out as an effective option for binary classification tasks requiring an equilibrium between positive class detection and general precision. However, more traditional models like Random Forest, Naive Bayes and Logistic Regression, when used with appropriate text representations (e.g., *TF-IDF*, *SBERT* or other advanced embeddings), can also offer competitive or even superior performance, especially in environments where simplicity and ease of adjustment are sought [28], [11].
4) **Considerations for Model Selection:** Model choice should be guided by the specific needs and capabilities of the application. For real-time monitoring applications, where specialist review and analysis are essential, assessing the capability to analyze false positives without penalizing true positives is crucial; or, for building indicators, selecting a model that facilitates appropriate correction of results is important.
5) **Effective Detection of Corruption Cases:** Advanced natural language processing (NLP) techniques, combined with appropriate data preprocessing and data augmentation strategies, proved effective in detecting patterns that may be associated with corruption cases. This has positive implications for transparency, as it enables the automated and accurate identification of suspicious behavior in large text volumes [29].

Finally, this study reinforces the importance of strategic model selection and text processing techniques based on specific objectives and highlights how advanced NLP techniques can be leveraged to promote transparency and improve detection in sensitive applications such as corruption identification.

Appendices include results of Top 15 - Best Detection Models and Instances obtained from all tested models. Traditional models with transfer learning of text representation give best results regarding adequate detection.

Additionally, in the following link: *<Results link>*, you can review all models and instances tested (1658) with their respective test metrics, curves and confussion matrices.

VI. Final Recommendations

- Expand the availability of labeled data through active or semi-supervised labeling techniques to maximize model potential.
- Consider implementing hybrid models, combining traditional and advanced techniques to optimize both cost and performance across different applications.
- Develop customized solutions that allow for model adjustment based on whether precision (general indicators) or detection capability (real-time monitoring) is prioritized.

Acknowledgment

We would like to thank the Kapak project, which has allowed access to the portal of indicators and related information, which facilitates the detection of corruption risks in public procurement processes based on data made transparent by the National Public Procurement Service (SERCOP) through the Official Public Procurement System of Ecuador (SOCE).

And we would like to thank Poligrants 24261 and 24270, whose support has been essential for the execution of this work.

References

[1] USFQ, "Kapak: transparencia en compras públicas," 2023, accedido: 13 de noviembre de 2024. [Online]. Available: https://noticias.usfq.edu.ec/2023/11/kapak-transparencia-en-compras-publicas.html

[2] P. for Humanity, "Kapak," 2023, accedido: 13 de noviembre de 2024. [Online]. Available: https://www.prototypesforhumanity.com/project/kapak/

[3] USFQ, "Kapak - transparencia en compras públicas," accedido: 13 de noviembre de 2024. [Online]. Available: https://kapak.usfq.edu.ec/#/inicio

[4] M. Fortuny, R. Sandoval, D. Riofrío, F. Simon, M. Baldeon-Calisto, R. Flores-Moyano, and D. Benítez, "Building a graph database for electronic inverse auction: Ecuador's case study," in *2023 IEEE Seventh Ecuador Technical Chapters Meeting (ETCM)*, IEEE. IEEE, 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10308986

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1734–1781, 1997.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[7] F. A. Research, "fasttext," 2016, accedido: 2024-12-01. [Online]. Available: https://fasttext.cc/

[8] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of NAACL-HLT*, 2018, pp. 2226–2238.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5997–6009.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4170–4187.

[11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1902.

[13] OpenAI, "Introducing text and code embeddings in the openai api," https://openai.com/blog/introducing-text-and-code-embeddings, 2022.

[14] ——, "New embedding models and api updates," https://openai.com/index/new-embedding-models-and-api-updates/, 2024.

[15] T. B. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners," pp. 1876–1901, 2020.

[16] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–49, 2019.

[17] S. Kobayashi, "Contextual augmentation: Data augmentation by words with paradigmatic relations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 451–458.

[18] Q. Ma, "Nlp augmentation library," 2019, available: https://github.com/makcedward/nlpaug.

[19] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[20] A. Radford, J. Wu, R. Child *et al.*, "Language models are unsupervised multitask learners," in *Proceedings of NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2019.

[21] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 145–162, 1954.

[22] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–524, 1988.

[23] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning*, 2014, pp. 1188–1196.

[24] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," in *Transactions of the Association for Computational Linguistics*, vol. 5, 2017, pp. 134–147.

[25] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 3981–3993.

[26] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1735.

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

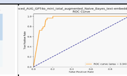
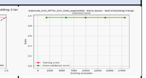
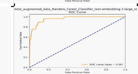
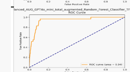
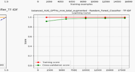
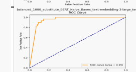
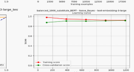
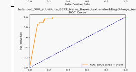
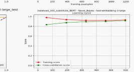
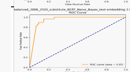
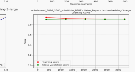
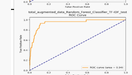
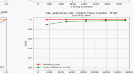
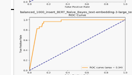
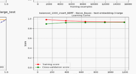
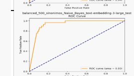
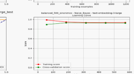
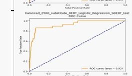
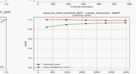
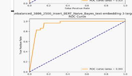
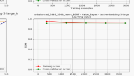
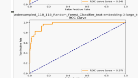
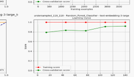
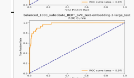
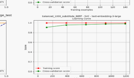
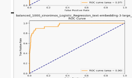
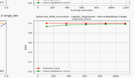
[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[29] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge university press, 2008.

[30] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar *et al.*, "Universal sentence encoder," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018, pp. 168–175.

[31] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 860–875, 2006.

Appendix A
Top 15: Metrics of Best Detection Models and Instances

Dataset Balance Type	Data Augmentation Type	Model	Text Representation Type	Runtime (minutes)	Test AUC ROC	Test Binary Recall	Test Binary F1 Score	Test Binary Precision	Test Macro Recall	Test Macro F1 Score	Test TP	Test FN	Test FP	Test TN	Best Traditional Params	Test Confusion Matrix	Test ROC Curve	F1 Score Learning Curve
Consolidated Data Augmentation	Consolidated Data Augmentation	Naive Bayes	text-embedding-3-large	0.020	0.940	1.000	0.227	0.128	0.898	0.557	29	0	198	774	{'var_smoothing': 0.35111917342151305}	774 198 0 29		
Consolidated Data Augmentation without GPT4o mini v2	Consolidated Data Augmentation without GPT4o mini v2	Random Forest Classifier	text-embedding-3-large	2.808	0.960	0.966	0.268	0.156	0.905	0.591	28	1	152	820	{'max_depth': None. 'min_samples_leaf': 1. 'n_estimators': 200}	820 152 1 28		
Consolidated Data Augmentation	Consolidated Data Augmentation	Random Forest Classifier	TF-IDF	1.558	0.940	0.931	0.325	0.197	0.909	0.632	27	2	110	862	{'max_depth': None. 'min_samples_leaf': 1. 'n_estimators': 200}	862 110 2 27		
Balanced 2	BERT - substitution	Naive Bayes	text-embedding-3-large	0.001	0.950	0.897	0.356	0.222	0.901	0.653	26	3	91	881	{'var_smoothing': 0.006579332246575682 }	881 91 3 26		
Balanced 1	BERT - substitution	Naive Bayes	text-embedding-3-large	0.000	0.940	0.897	0.306	0.184	0.889	0.621	26	3	115	857	{'var_smoothing': 0.012328467394420659 }	857 115 3 26		
Unbalanced 2	BERT - substitution	Naive Bayes	text-embedding-3-large	0.004	0.950	0.862	0.350	0.219	0.885	0.650	25	4	89	883	{'var_smoothing': 0.005336699231206307 }	883 89 4 25		
Consolidated Data Augmentation without GPT4o mini v2	Consolidated Data Augmentation without GPT4o mini v2	Random Forest Classifier	TF-IDF	1.259	0.940	0.862	0.327	0.202	0.880	0.636	25	4	99	873	{'max_depth': None. 'min_samples_leaf': 1. 'n_estimators': 200}	873 99 4 25		
Balanced 2	BERT - insertion	Naive Bayes	text-embedding-3-large	0.001	0.940	0.862	0.305	0.185	0.874	0.621	25	4	110	862	{'var_smoothing': 0.012328467394420659 }	862 110 4 25		
Balanced 1	Synonyms	Naive Bayes	text-embedding-3-large	0.000	0.930	0.862	0.303	0.184	0.874	0.620	25	4	111	861	{'var_smoothing': 0.012328467394420659 }	861 111 4 25		
Balanced 3	BERT - substitution	Logistic Regression	SBERT	0.040	0.930	0.828	0.429	0.289	0.883	0.697	24	5	59	913	{'C': 100. 'penalty': 'l2'. 'solver': 'saga'}	913 59 5 24		
Unbalanced 2	BERT - insertion	Naive Bayes	text-embedding-3-large	0.006	0.940	0.828	0.366	0.235	0.874	0.661	24	5	78	894	{'var_smoothing': 0.003511191734215131 }	894 78 5 24		
Undersampled	Undersampled	Random Forest Classifier	text-embedding-3-large	0.002	0.970	0.828	0.331	0.207	0.866	0.639	24	5	92	880	{'max_depth': 30. 'min_samples_leaf': 4. 'n_estimators': 200}	880 92 5 24		
Balanced 2	BERT - substitution	SVC	text-embedding-3-large	0.124	0.970	0.793	0.582	0.460	0.883	0.783	23	6	27	945	{'C': 100}	945 27 6 23		
Balanced 2	Synonyms	Logistic Regression	text-embedding-3-large	0.054	0.960	0.793	0.541	0.411	0.880	0.760	23	6	33	939	{'C': 100. 'penalty': 'l2'. 'solver': 'saga'}	939 33 6 23		
Balanced 4	Synonyms	Logistic Regression	text-embedding-3-large	0.170	0.940	0.759	0.677	0.611	0.872	0.833	22	7	14	958	{'C': 100. 'penalty': 'l2'. 'solver': 'saga'}	958 14 7 22