UAEMex Participation at TA1C-2025: Leveraging Lexical and Semantic Information for Clickbait Detection in Spanish

Jonathan Rojas-Simón[1,*,†], Christian Ruiz-Ugalde[1,†], Noé Torres-Pedroza[1,†], María del Carmen García-Galindo[1,†], Verónica Neri-Mendoza[1,2,†], Yulia Ledeneva[1,†], and René Arnulfo García-Hernández[1,†]

[1] *Autonomous University of the State of Mexico, Instituto Literario 100, Toluca 50000, Mexico*
[2] *Secretariat of Science, Humanities, Technology and Innovation, No. 1582, Insurgentes Sur Avenue, Credito Constructor, Benito Juarez, Mexico City, Mexico*

Abstract

In today's digital landscape, social media information has become a valuable source for capturing readers' attention quickly through clickbait. However, this information sometimes leads users to low-quality or misleading information. For this reason, the TA1C shared task in IberLEF 2025 focuses on detecting clickbait from social media posts in Spanish. This paper presents an NLP framework developed by the UAEMex team that exploits lexical and semantic information from texts to detect this kind of information. This framework was also complemented with traditional classifiers (KNN, LR, NaiveBayes, and MLP) to create models that best fit the task. According to the obtained results, semantic representations provided by BETO (Spanish version of BERT) provide better representations with LR and MLP classifiers, obtaining 0.7538 in F1-score, which is a competitive performance compared to state-of-the-art approaches and baselines in the task.

Keywords

NLPClassKit, ASCII, BERT, Logistic Regression (LR), and Multilayer Perceptron (MLP)

1. Introduction

According to Mordecki, clickbait is defined as "a technique for generating headlines and teasers that deliberately omit part of the information with the goal of raising the readers' curiosity, capturing their attention and enticing them to click" [1]. This definition emphasizes the intentional use of curiosity as an attention-grabbing mechanism. By exploiting this cognitive bias, media outlets capture the user's attention almost involuntarily, distracting them from more relevant content. Other researchers have defined clickbait as "a viral journalism strategy that seeks to provoke users into clicking a hyperlink by means of news selection and writing techniques that function as bait" [2] a definition that approaches the phenomenon more as an editorial technique than a psychological effect.

In practice, clickbait has detrimental effects on the quality of journalism and the formation of public opinion, as it displaces relevant information in favor of attractive but superficial or misleading content. Various authors agree that clickbait undermines the quality of public discourse and the healthy functioning of democratic societies. Palau-Sampio warns that leading newspapers such as El País have adopted sensationalist writing strategies that blur the boundaries between quality journalism and entertainment [3]. Mordecki similarly argues that when citizens receive incomplete or irrelevant information, their ability to make informed decisions is compromised [1].

IberLEF 2025 September 2025, Zaragoza, Spain
* Corresponding author.
† These authors contributed equally.
jrojass@uaemex.mx (J. Rojas-Simón); cruizu@uaemex.mx (C. Ruiz-Ugalde); ntorresp003@alumno.uaemex.mx (N. Torres-Pedroza); marycarmeng142@gmail.com (M.C. García-Galindo); veronica.nerimendoz@gmail.com (V. Neri-Mendoza); ynledeneva@uaemex.mx (Y. Ledeneva); reagarciah@uaemex.mx (R.A. García-Hernández)
0000-0002-0389-4201 (J. Rojas-Simón); 0000-0002-9582-9502 (V. Neri-Mendoza); 0000-0003-0766-542X (Y. Ledeneva); 0000-0001-7941-377X (R.A. García-Hernández)

Potthast et. al., also criticizes media outlets that prioritize maximizing clicks for economic gain at the expense of content quality [4].

To enable automatic clickbait detection using machine learning, some researchers have compiled datasets to train models that have proven effective for natural language processing (NLP) tasks. However, Mordecki and colleagues in [1] point out two major issues with existing datasets: (a) most datasets are in English language, and there is no consensus on a precise definition of clickbait, and (b) the dataset construction methods may introduce bias into what is considered clickbait. For instance, reputation-based methods label all headlines from sensationalist sources as clickbait, even if some are legitimate. Likewise, gatekeeper-based methods rely on expert judgments, which may reflect subjective biases rooted in experience, editorial values, or personal perceptions.

To address these problems, Mordecki and collaborators recently introduced the TA1C (Te Ahorré Un Click) dataset—the first manually annotated Spanish-language corpus for clickbait detection, consisting of 3,500 tweets from 18 media outlets [1]. Its main contribution lies in providing rigorous and less subjective annotations. The dataset was built using clear operational criteria to identify deliberate omission of information, considering the reader's context and prior knowledge, and recognizing when adjectives create informational gaps.

The dataset was publicly presented under the title Detecting and Spoiling Clickbait in Spanish-Language News [5] at IberLEF 2025 [6], within the Content Curation and Generation track. The authors of the present article participated in that shared task and report here on the results obtained and we will present a component-based framework designed to facilitate the development of classification models for clickbait classification projects, which will be detailed in the following sections.

This paper is organized as follows: Section 2 reviews related work on clickbait detection using various datasets and NLP methods. Section 3 describes the proposed system for clickbait classification. Section 4 discusses the results and hyperparameters for each method, and Section 5 outlines the main conclusions and feature experimentations.

2. Related Studies

Automatic clickbait detection has gained increasing attention in NLP community due to its implications for digital misinformation, user trust, and the quality of online journalism [1]. Various datasets have been carefully compiled, typically consisting of short headlines or social media posts labeled as clickbait or non-clickbait, which are used to train machine learning models.

To enable these models to operate effectively, a vectorization phase is required in which textual data is transformed into numerical representations suitable for computational processing. Vectorization techniques range from traditional approaches such as bag-of-words or TF-IDF [5] to dense representations like Word2Vec [7], GloVe [8], or contextual embeddings derived from transformer-based models like BERT [9]. The choice of vectorization method directly influences the model's ability to capture semantic structures and signals associated with clickbait, such as vague pronouns, emotional cues, or curiosity gaps.

Regarding machine learning algorithms, models commonly used for clickbait detection range from traditional approaches such as Logistic Regression (LR), Support Vector Machines (SVM), Random Forests (RF), and Multilayer Perceptron (MLP), to more recent and complex deep learning architectures—including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and transformer-based models—designed to more effectively capture the contextual information inherent in natural language [1]. Below are some studies related to clickbait detection, highlighting the diversity of datasets, vectorization techniques, machine learning models, and evaluation metrics employed.

One early approach is proposed in [10], who built a custom dataset using Facebook posts categorized based on the reputation of their sources. The author applied Word2Vec for word

embedding and trained a fully connected deep neural network with dropout and batch normalization, achieving an accuracy of 0.93.

In [11], the researchers also used Facebook posts to build their dataset and employed Word2Vec for vectorization. Their experiments included both Feedforward Neural Networks and Convolutional Neural Networks (CNN), with CNN achieving an impressive accuracy of 98.3%. Their study demonstrated the effectiveness of deep learning architectures in capturing textual patterns indicative of clickbait.

Other researchers developed a dataset of headlines drawn from both reputable media outlets and sources known for their clickbait [12]. Using TF-IDF for text representation, they trained several classifiers, including Logistic Regression (LR), Support Vector Machines (SVM), and Random Forests (RF), with RF yielding the best performance at an F1-score of 0.93. This work underscored the importance of feature engineering in traditional machine learning pipelines [12].

In [4], the authors worked with the Webis Clickbait Corpus 2017, comprising 38,000 tweets annotated with clickbait intensity scores on a scale from 0 to 1. They used bag-of-words and n-gram features, applying SVM and Ridge Regression for regression-based predictions, with their ensemble model achieving a Mean Squared Error (MSE) of 0.04.

In a recent study, academics presented in [1] the TA1C dataset. This study explored both TF-IDF and transformer-based embeddings (fastText and Beto) and evaluated LR, XGBoost, and GPT-4, achieving the highest F1 score of 0.84 using the Beto model.

3. Proposed System

This section provides a detailed overview of the proposed approach. First, we detail the data stratification process used for evaluation. Second, we provide a description of the preprocessing steps applied to the datasets. Next, we describe the text representation methods employed, and finally, it introduces the classification algorithms used, along with their corresponding hyperparameter settings.

3.1. Data stratification

Following the TA1C shared task (*Te Ahorré Un Click – Clickbait Detection and Spoiling in Spanish*), we used the official clickbait detection corpus made available for the competition.

The TA1C dataset consists of 3,500 news items collected from Twitter (currently known as X), originating from 18 reputable media sources that are national or international (excluding local outlets), with a generalist focus (not topic-specific) and high popularity. The corpus is divided into three subsets: 2,100 instances for training, 700 for validation, and 700 for testing [5].

The TA1C shared task includes two subtasks. *Subtask 1*, called *Clickbait Detection*, aims to determine if the content of a tweet that links to a piece of news is clickbait. In contrast, *Subtask 2*, named *Clickbait Spoiling*, consists of generating or extracting from the article a short text – up to 280 characters – that, as concisely as possible, either satisfies the curiosity sparked by the headline or indicates that the article does not provide a clear answer. In this paper we focus on Subtask 1.

During the first phase, the organizers released two datasets: a training set and a development set. Only the training set included labels, allowing participants to build supervised classification models. At this stage, the development set labels were not disclosed; however, participants could evaluate their models using this data through the CodaLab platform [13]. Table 1 shows the class distribution of the training set.

In the second phase, the organizers re-released the training set along with the now-labeled development set and an unlabeled test set, which was used for the final evaluation. The class distribution of the development set is detailed in Table 2.

The class distribution imbalance, with 71.50% clickbait texts in the training set and 71.00% in the development set, indicates a high prevalence of clickbait content. This disproportion may bias models toward the majority of the class. Furthermore, detecting clickbait requires analyzing not

only the superficial content of the text but also semantic and contextual aspects that enable understanding of the communicative intent and potential ambiguities present in the message.

Table 1
Statistics of the training set

Class	# Samples	Percentage
Clickbait	798	28.50%
No	2002	71.50%
Total	2800	100.00%

Each dataset contains the followings columns: Tweet ID, Tweet Date, Media Name, Media Origin, Teaser Text and Tag Value. The data from Teaser Text column was processed and vectorized to generate input features for the classification models and the data from Tag Value column was used as the label to predict. Table 3 shows two representative examples of texts labeled as clickbait and two corresponding to the non-clickbait class.

Table 2
Statistics of the development set

Class	# Samples	Percentage
Clickbait	203	29.00 %
No	497	71.00 %
Total	700	100.00 %

The following examples illustrate certain linguistic and discursive features that differentiate content labeled as clickbait from non-clickbait. In general, texts classified as clickbait tend to employ questions, ambiguous phrasing, or curiosity-inducing expressions aimed at encouraging the reader to click in order to access the full content. In contrast, non-clickbait texts present information in a clear, direct, and specific manner, without omitting key elements of the text.

Table 3
Examples of texts labeled as clickbait and non-clickbait

Class	Text
Clickbait	Cinco países europeos quieren crear 'corredores sanitarios' frente al coronavirus
Clickbait	¿Qué es y cuándo será el Cyber Monday 2020? RT @Vive_USA: El Cyber Monday o Lunes Cibernético que marca el inicio de las compras navideñas en línea.
No	México llega a 668 mil 381 casos de Covid; hay 70 mil muertes. #ÚltimaHora México llegó a 70 mil 821 muertes por Covid-19, con 668 mil 381 casos confirmados de coronavirus
No	Policías recibían hasta $400.000 por avisarles a ladrones cuándo robar en TransMilenio, según investigación de la Fiscalía

3.2. Preprocessing

To ensure the quality and consistency of the input data before training the models, a preprocessing stage was applied to the TA1C dataset. This step is crucial in NLP tasks, as it reduces noise, standardizes textual input, and facilitates the extraction of relevant features. The processing aimed to transform raw tweets into structures and clean format for machine learning algorithms. Below we describe the preprocessing techniques considered in this study:

- Tokenization: As an initial step, tokenization was applied to split the text into individual units (tokens), typically corresponding to characters or words to improve the association or "understanding" of words for each sentence.
- Normalization: In this process we removed non-alphanumeric characters, and all texts were converted to lowercase.
- Stopwords removal: In some cases, we removed common words that carry little semantic weight, such as "el", "la", "y", or "de" in Spanish — were removed using a predefined list from the NLTK library. This step reduces noise and helps focus the model on more informative terms.

3.3. Model architecture

The model's architecture is built upon a component-based framework called NLPClassKit. This toolkit consists of a set of software components designed to facilitate the development of classification models for NLP projects. It is currently available in a GitLab repository (https://gitlab.com/JohnRojas/NLPClassKit) and follows a process-oriented architecture as illustrated in Figure 1.

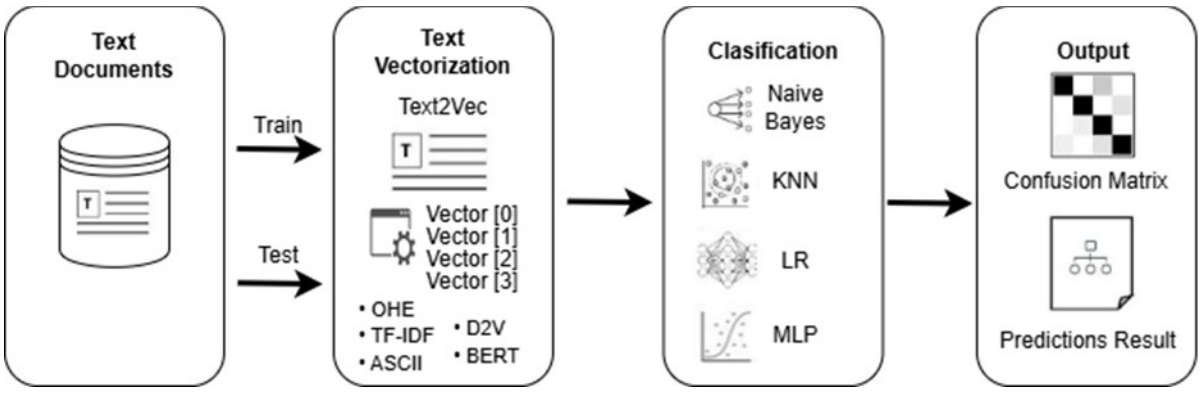

Figure 1: Architecture of NLPClassKit.

3.3.1. Text vectorization

Once the text documents were extracted or preprocessed, they were fed into various vectorization methods. These techniques are crucial for representing and extracting relevant information from the text, whether at the lexical or semantic level. Some of these approaches are implemented in a software component called Text2Vec, developed in Python and hosted in a GitLab repository (https://gitlab.com/MLComponents1/Text2Vec). Technical aspects related to syntax; parameters and the execution process are described in the README file of the repository. This module encodes text using the following strategies:

One-hot-encoding (OHE): A straightforward technique that represents words as binary vectors. First, vocabulary is built from the text, assigning a unique index to each term. Then, each word is converted into a binary vector with a value of 1 in the position corresponding to its presence in the document and 0 elsewhere. Although OHE requires minimal computational resources, it produces

high-dimensional vectors, especially with large vocabularies [14]. The length of the output vectors is variable and depends on the input vocabulary; for example, during our training phase, a vector of dimension 18,770 was produced using this method with the test set.

TF-IDF (Term Frequency-Inverse Document Frequency): A frequency-based method that evaluates the importance of a word in a document relative to a corpus. It combines Term Frequency (TF), which counts how often a word appears in a document with Inverse Document Frequency (IDF) that measures its rarity across the corpus. This weighting reduces the impact of common terms and highlights those more relevant to the document's content [15]. On the other hand, TF-IDF encoding outputs a sparse vector representation in which each dimension corresponds to a specific term from the corpus vocabulary. The value of each dimension reflects the weight of that term in the document based on its TF-IDF score. The length of the resulting vector depends on the size of the vocabulary, which is determined by steps such as tokenization, stop words removal, etc.

Doc2Vec: An extension of the Word2Vec model that generates vector representations for entire documents, paragraphs, or sentences rather than individual words. This approach captures the overall semantic context of the text and encodes it into a fixed-length vector, facilitating tasks like document comparison, classification, or clustering [16]. The length of the output vector is variable and can be controlled through its parameters. Other parameters such as context size, the use of iterations, or the use of bag-of-words or skip-grams, among others, can also be controlled by the model user.

BERT (Bidirectional Encoder Representations from Transformers): A pre-trained model based on attention mechanisms, capable of interpreting word context bidirectionally (analyzing both preceding and subsequent words). Unlike traditional methods, BERT uses a transformer architecture to capture complex relationships and deep contextual meanings. Additionally, it can be fine-tuned with domain-specific texts to enhance performance in specialized tasks [17]. BERT-base is a language model with 12 layers, 768 hidden units, 12 attention heads, and a vocabulary of 30,522 tokens. It can process sequences of up to 512 tokens and has approximately 110 million parameters. The extended version, BERT-large, reaches 340M parameters.

ASCII: An approach that analyzes the frequency of ASCII characters in a document to estimate the probability of each character's occurrence. Though simpler than other techniques, it is useful for specific tasks like style or formatting classification [18]. This encoding always outputs a 256-dimensional vector, in which the first 128 characters correspond to upper- and lower-case letters, numbers, punctuation marks and some control characters from the English language, and the remaining characters include symbols and accented letters from other languages.

3.3.2. Classification

The vector representations generated in the preceding stage serve as input for multiple supervised machine learning models, each specifically tailored to categorize textual data through pattern recognition. Below is a concise overview of the primary algorithms employed in this research:

Naive Bayes (NB): This probabilistic classification model operates on the foundation of Bayes' Theorem. It computes the likelihood of a text sample belonging to a specific category by analyzing its feature set. Although NB employs a simplified approach by assuming feature independence, it demonstrates notable efficacy in text classification scenarios, especially when this assumption approximately holds true.

K-Nearest Neighbors (KNN): As a non-parametric classification method, KNN determines class membership for new instances by identifying the predominant class among its K closest neighboring points within the vector space. The algorithm operates on a majority voting system and proves particularly valuable for complex, non-linear decision boundaries. The selection of neighbors depends crucially on distance computations, commonly employing measures such as Euclidean distance or cosine similarity.

Logistic Regression (LR): Logistic Regression is a linear classification algorithm used to predict binary outcomes. It models the relationship between a set of input features X and a binary target variable $y \in \{0,1\}$. The prediction is made using the sigmoid function:

$$P(y=1)=\frac{1}{1+e^{-z}}, z=\sum w_i x_i, \quad (1)$$

where w_i are the model parameters and x_i are the input features. The output is a probability value between 0 and 1, indicating the likelihood that the input belongs to class 1.

Multilayer Perceptron (MLP): The MLP represents a fundamental deep learning architecture comprising three distinct fully connected layer types: (i) an input layer that receives feature representations, (ii) multiple hidden layers capable of learning hierarchical feature transformations, and (iii) an output layer that produces classification probabilities. This neural network architecture demonstrates effectiveness in modeling non-linear relationships within complex datasets, making it well-suited for text classification applications where it can learn intricate patterns in textual representations.

4. Experiments and obtained results

Finally, the outcome of the classification process is presented in the final stage, where each algorithm described in the previous section produces the following output:

- Confusion Matrix (CM): Displays the number of texts correctly and incorrectly classified.
- Predictions: Lists the predicted labels generated during the training phase of each algorithm.
- Model Performance: Provides a detailed evaluation of each classification model based on metrics such as Recall, Precision, and F1-score.

During the experimentation and testing phase, the team members conducted over 1,500 experiments to explore various parameter configurations in both vectorization methods (VecM) and classifiers (Clf). We shared the best results among ourselves, aiming to explore as many options as possible. In the final phase, predictions were made on the test set of 700 unlabeled tweets, which were used to generate the official scores shown in Table 4. The performance of each model is presented in the Precision (P), Recall (R), and F1-score (F1) metrics. Moreover, we show the dimensions of vector representations (D_{vec}) and the number of parameters of BERT-based models (N_p) for each experiment.

Table 4 presents a comparative analysis of different features and classifiers tested in the NLPClassKit system. The best overall performance on the test set was achieved by UAEMex-1 and UAEMex-2, both using the pre-trained model albert-base-spanish combined with a LR classifier. These models obtained the highest F1-score of 0.7538, indicating a strong balance between precision and recall. UAEMex team corresponded to the set of classifiers with simple majority voting.

Interestingly, UAEMex-3, which used the same text vectorization but replaced LR with a MLP, showed a slightly lower precision (0.7273) but improved recall (0.7665) resulting in a competitive F1-score of 0.7463. In contrast, when we used ASCII vectorization combined with bert-base-multilingual-cased (UAEMex-4-2 and UAEMex-4-3), we achieved the lowest performance with F of 0.6610 and 0.6369, suggesting that this combination is less effective for the clickbait detection task. Meanwhile, configurations that combined bert-base-spanish-wwm-cased with albert-base-spanish (UAEMex-4-4 and UAEMex-4-5) outperformed the multiligual-based system, reaching an F1-score of 0.7122 and 0.7352. However, these results were slightly below those of the best models, where only alBERT features were used. These results demonstrate that not only the classifier architecture but also the quality and linguistic adequacy of the vector representations are key factors in optimizing performance in Spanish text classification tasks.

Finally, we combined the embedding generated by bert-base-spanish-wwm-cased and albert-base-spanish. Although this may initially seem redundant, given that both models are pre-trained on Spanish; However, we decided to carry out this experiment because they have different architectures, which means they learn and represent distinct aspects of the language. On one hand, BERT has a greater capacity to capture complex semantic relationships; while albert, being a lighter and more efficient model, tends to generalize better in simpler or noisier contexts.

Table 4
Best results obtained by the team in the evaluation phase. The highest results are italicized.

Experiment	VecM	Clf	D_{vec}	N_p	Training			Test		
					P	R	F1	P	R	F1
UAEMex-1	ABS	LR	768	12M	0.7892	0.9049	0.8431	*0.7654*	0.7425	*0.7538*
UAEMex-2	ABS	LR	768	12M	0.7892	0.9028	0.8422	0.7654	0.7425	*0.7538*
UAEMex-3	ABS	MLP	768	12M	0.9361	*0.9081*	0.922	0.7273	*0.7665*	0.7463
UAEMex-4-1	ASC	MLP	256	N/A	0.8752	0.8411	0.8578	0.4647	0.4731	0.4688
UAEMex-4-2	ABBM	MLP	512	179M	0.9124	0.8851	0.8985	0.6257	0.7006	0.6610
UAEMex-4-3	ABBM	LR	512	179M	0.8762	0.7712	0.8204	0.6331	0.6407	0.6369
UAEMex-4-4	BSWC	MLP	896	110M/ 12M	*0.9523*	0.8981	*0.9244*	0.6793	0.7485	0.7122
UAEMex-4-5	BSWC	LR	896	110M/ 12M	0.9441	0.8781	0.9099	0.7225	0.7485	0.7352

ABS: albert-base-spanish, ASC: ASCII, ABBM: ASCII + bert-base-multilingual-cased, and BSWC: bert-base-spanish-wwm-cased + albert-base-spanish.

In general, we expected that combination of both features would allow us to leverage their complementary strengths and thus enhance the classification system is performance, but in this case, the results showed a slight decrease in the F-measure compared to using albert-base-spanish. This drop in performance could be attributed to redundancies or conflicts in the embeddings generated by the two models, which may introduce noise rather than providing additional value.

Regarding vectorization methods, the best performance was obtained with albert, a linguistic model based on BERT, trained specifically for the Spanish language by the Computer Science Department of the University of Chile and available in the Hugging Face repository [13]. It should be noted that other Spanish-oriented BERT models were also evaluated, such as bert-base-spanish-wwm-uncased and bert-base-multilingual-cased, which showed lower performance than albert when used with the same classifiers. Table 5 presents the hyperparameter settings corresponding to the machine learning models that obtained the best results.

Table 5
Vectorization settings and methods.

Stage		Parameter	Values
Text vectorization	BERT	Model name	albert-base-spanish
		Max length	512
		Pooling	CLS
Classification	LR	Iterations	700
	MLP	Number of Hidden Layers (HL)	3
		Configuration of HL	100, 50, 10

Activation Function in HL	Relu
Number of Epochs	50
Learning Rate	0.0005
Activation Function in Output Layer	sigmoid

After the evaluation phase was completed, the results were published on the CodaLab platform, an open-source system designed to host and manage scientific challenges [11]. The performance of the participating teams, including our best experiments, is summarized in Table 6. It presents both the F1 scores obtained during the official competition classification and the internal evaluation metrics calculated during the training phase. This table compares the results of our team with those of the other participants.

Table 6
Overall performance of the proposed experimentation against other participants.

Team	Task 1 score
tomasbernal01	0.81564
escom	0.81525
viahes	0.81482
dcere	0.80480
julian_zsa	0.80347
gsdeyson	0.80115
Omar.Garcia	0.79558
gaspai	0.77748
danielrod99	0.77562
UAEMex-1	*0.75380*
UAEMex-2	*0.75380*
UAEMex-3	0.74636
UAEMex-4	0.73529

According to the F1 scores, our team ranked 10th, 11th, 12th, and 13th. Compared to the top-performing team (gsdeyson), the difference was 0.047 points. It is worth noting that all the metrics presented in the table were calculated using a combined training set composed of the 2,800 labeled tweets provided during the development phase and an additional 700 labeled tweets from the evaluation phase.

Additionally, the confusion matrices corresponding to the best results are displayed in Figure 2. It can be observed that the models exhibit a slight bias toward predicting that a given text does not contain clickbait. This tendency is primarily attributed to the class imbalance in the training data, with 1,001 clickbait examples versus 2,499 non-clickbait examples (see Tables 1 and 2). On the other hand, we observe that the best-performing model (UAEMex-2) does not show a better prediction of clickbait in the training stage compared to UAEMex-3 and UAEMex-4-5 models. We assume that this may be an indicator of generalization in the proposed model, enabling a balance between non-clickbait (0) and clickbait (1).

5. Conclusions and Future Works

In this study, the team collaborated to address the clickbait detection task proposed by Mordecki and colleagues as part of the IberLEF 2025 forum, with the goal of contributing to the development of algorithms capable of eliminating unethical practices exhibited by some digital media creators and promoting high-quality journalism.

During the competition, various text vectorization techniques were explored, allowing for a comparative analysis between several custom BERT-based models for Spanish and traditional vectorization approaches. This experimentation enabled the identification of the most suitable method for this specific natural language processing task. Among the tested models, Albert proved to be the most effective, consistently outperforming other Spanish-language BERT models during the evaluation phase.

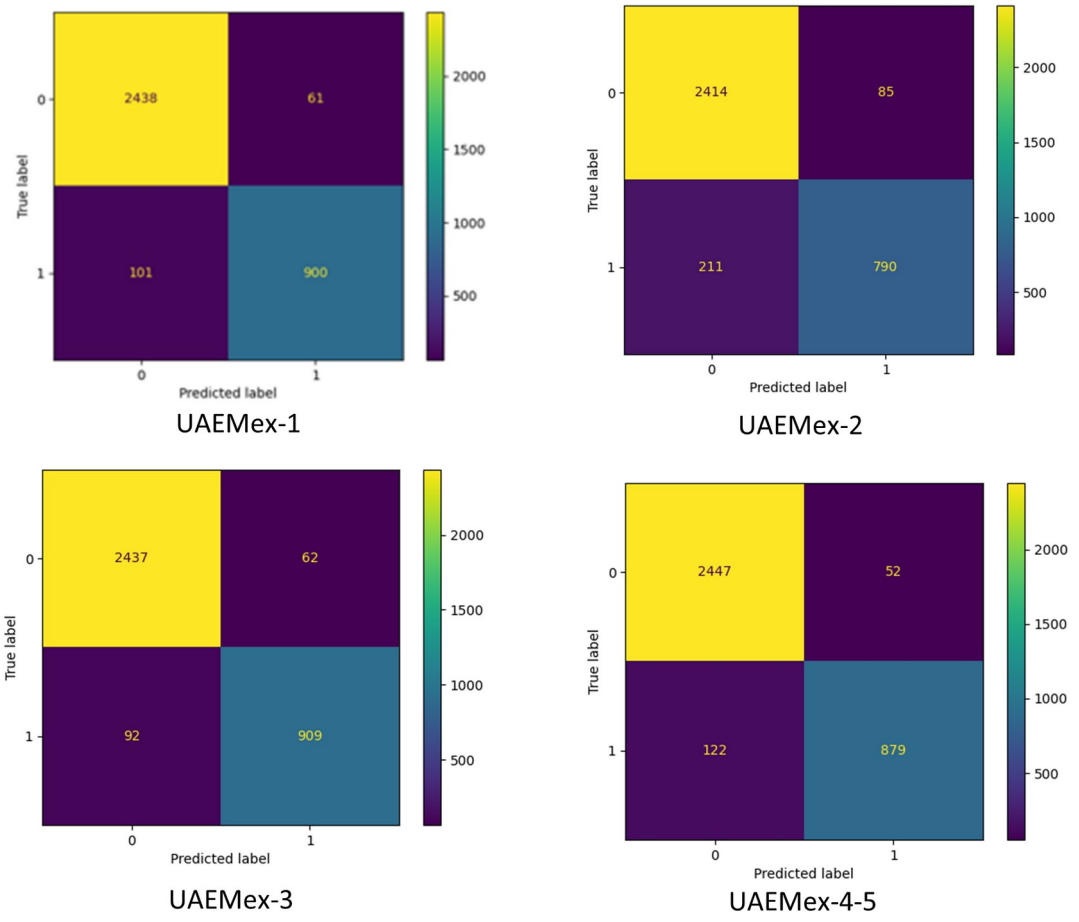

Figure 2: Confusion matrices of the best results in the training stage.

Additionally, four classic supervised machine learning algorithms were tested, yielding several noteworthy findings. For instance, while the Multilayer Perceptron achieved strong results during training, Logistic Regression demonstrated superior generalization performance during the competition, which is the primary goal of a classification model. Similarly, the ensemble classifier based on majority voting achieved comparable performance to logistic regression, likely due to its internal inclusion of similar models.

For future work, we recommend further exploration of BERT-based Spanish word embedding models and experimentation with alternative classifiers beyond those evaluated in this study. Furthermore, no dimensionality reduction techniques were applied in this project; consequently, classification efficiency could potentially be enhanced through the implementation of methods such as Principal Component Analysis (PCA), Genetic Algorithms (GAs), or Information Gain, among others. Additionally, the implementation of a probability-based ensemble classifier is suggested as a promising line of research.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly tool in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Mordecki, G., Moncecchi, G., & Couto, J. (2025). Te Ahorré Un Click: A Revised Definition of Clickbait and Detection in Spanish News. En L. Correia, A. Rosá, & F. Garijo (Eds.), Advances in Artificial Intelligence – IBERAMIA 2024 (pp. 387–399). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-80366-6_32

[2] Bazaco, Á., Redondo, M., & Sánchez-García, P. (2019). El clickbait, como estrategia del periodismo viral: Concepto y metodología. Revista Latina de Comunicación Social, 74, pp. 94-115. https://doi.org/10.4185/RLCS-2019-1323

[3] Palau-Sampio, D. (2016). Reference press metamorphosis in the digital context: clickbait and tabloid strategies in Elpais.com. Communication & Society, 29(2), pp. 63–80. https://doi.org/10.15581/003.29.35924

[4] Potthast, M., Gollub, T., Komlossy, K., Schuster, S., Wiegmann, M., Garces Fernandez, E. P., Hagen, M., & Stein, B. (2018). Crowdsourcing a Large Corpus of Clickbait on Twitter. En E. M. Bender, L. Derczynski, & P. Isabelle (Eds.), Proceedings of the 27th International Conference on Computational Linguistics (pp. 1498–1507). Association for Computational Linguistics. https://aclanthology.org/C18-1127

[5] Mordecki, G., and Chiruzzo, L., Laguna, R., Prada, J. J., Rosá, A., Sastre, I., & Moncecchi, G. (2025). Overview of TA1C at IberLEF 2025: Detecting and Spoiling Clickbait in Spanish-Language News. Procesamiento de Lenguaje Natural, 75.

[6] González-Barba, J. A., Chiruzzo, L., Jiménez-Zafra, S. M. (2025). Overview of IberLEF 2025: Natural Language Processing Challenges for Spanish and other Iberian Languages. In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS.org

[7] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, 1–12. https://doi.org/10.48550/arXiv.1301.3781

[8] Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. En A. Moschitti, B. Pang, & W. Daelemans (Eds.), In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

[9] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186, Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

[10] Agrawal, A. (2016). Clickbait detection using deep learning. In 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), pp. 268–272. https://doi.org/10.1109/NGCT.2016.7877426

[11] Rony, M. M. U., Hassan, N., & Yousuf, M. (2017). Diving Deep into Clickbaits: Who use them to what extents in which Topics with what Effects? In Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, pp. 232-239. https://doi.org/10.1145/3110025.3110054

[12] Chakraborty, A., Paranjape, B., Kakarla, S., & Ganguly, N. (2016). Stop clickbait: Detecting and preventing clickbaits in online news media. In 2016 IEEE/ACM international conference on

advances in social networks analysis and mining (ASONAM). pp. 9-16. https://doi.org/10.1109/ASONAM.2016.7752207

[13] Pavao, A., Guyon, I., Letournel, A.-C., Tran, D.-T., Baro, X., Escalante, H. J., Escalera, S., Thomas, T., & Xu, Z. (2023). CodaLab Competitions: An Open Source Platform to Organize Scientific Challenges. Journal of Machine Learning Research, 24(198), pp. 1-6.

[14] Rojas-Simon, J., Ledeneva, Y., & Garcia-Hernandez, R. A. (2022). Evaluation of Text Summaries Based on Linear Optimization of Content Metrics (Vol. 1048). Springer International Publishing. https://doi.org/10.1007/978-3-031-07214-7.

[15] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In International conference on machine learning. 32(2), pp. 1188-1196.

[16] Abubakar, H. D., Umar, M., & Bakale, M. A. (2022). Sentiment classification: Review of text vectorization methods: Bag of words, Tf-Idf, Word2vec and Doc2vec. *SLU Journal of Science and Technology*, *4*(1), pp. 27-33. https://doi.org/10.56471/slujst.v4i.266

[17] Devlin J., Chang M.-W., Lee K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. pp. 4171–4186, Accessed: May 20, 2025. [Online]. Available: https://github.com/tensorflow/tensor2tensor.

[18] Departamento de Ciencias de la Computación de la Universidad de Chile (DCC UChile). (2023). ALBERT Base Spanish BERT: dccuchile/albert-base-spanish. Available: https://huggingface.co/dccuchile/albert-base-spanish.