Participation of ESCOM's NLP Group at TA1C-IberLEF2025: RoBERTa Model Fine-Tuned for Clickbait Detection

Omar Juárez Gambino[1,*], José-Emiliano Ledesma-Ramírez[1], Raul Rodas-Rodríguez[1], Omar-Alejandro Velázquez-Cruz[1], Yahir Arias-Morales[1], Oscar-Galo Ayala-García[1], Axel-Maximiliano Rivera-García[1], David Ramírez-Rosas[1] and Consuelo-Varinia García-Mendoza[1]

[1]*Escuela Superior de Cómputo (ESCOM-IPN), Instituto Politénico Nacional, J.D. Batiz e/ M.O. de Mendizabal s/n, Mexico City, 07738, Mexico*

Abstract

In this paper, we describe the participation of the ESCOM NLP group in the TA1C 2025 Clickabit detection task. The contest proposes a binary classification of tweets to determine whether the content is clickbait or not. We trained several models using traditional machine learning methods and LLMs. Our best model achieved an F1 score of 0.8152 and ranked second in the final competition results.

Keywords

Clickbait detection, Text Classification, Machine Learning,

1. Introduction

Internet access has revolutionized how people consume and share information, enabling a massive content exchange. New avenues have opened for companies to attract customers online, giving rise to digital marketing. Today, products or services can be promoted using digital technologies through electronic media [1]. Web pages can generate revenue through various monetization models. One of the most common is Pay Per Click, where site owners earn money each time a user clicks on an advertisement link. This type of advertising, known as contextual advertising, allows companies to pay only when their ads receive interaction, ensuring greater control over their digital marketing investment [2].The excessive and unethical use of this type of advertising has led to the rise of clickbait. This strategy has proven to be effective in attracting traffic. However, it has simultaneously harmed information quality, causing a decline in public trust due to sensationalist headlines that promise more than the content actually delivers. This practice has become common on digital platforms and social networks, where the competition for user attention is fierce [3].

Given the problems this type of information can cause users, techniques for automatically detecting this kind of text have been explored. Most of the work carried out has been for the English language, as reported in [4]. The authors determine if headlines of news articles are clickbaits. They use a Sentence Bidirectional Encoder (SBERT) to measure the dissimilarity between the headline of the news article and the content. The dissimilarity among other extracted features are provided to a SVM classifier to identify clickbait obtaining 0.84 of accuracy. There is also work for low-resource languages, like the study done in [5]. The authors used a combination of deep learning algorithms and classical machine learning techniques for identifying clickbait in Urdu. Using word embeddings and LSTM networks, they achieved 0.88 of accuracy. Work for the Spanish language is scarce. One of the most relevant efforts is the one carried out in [6]. This article describes the collection of a Spanish news corpus from

IberLEF 2025, September 2025, Zaragoza, Spain

[*] Corresponding author.

jjuarezg@ipn.mx (O. J. Gambino); jledesmar1800@alumno.ipn.mx (J. Ledesma-Ramírez); rrodasr1800@alumno.ipn.mx (R. Rodas-Rodríguez); ovelazquezc1801@alumno.ipn.mx (O. Velázquez-Cruz); yariasm1800@alumno.ipn.mx (Y. Arias-Morales); oayalag2100@alumno.ipn.mx (O. Ayala-García); ariverag2200@alumno.ipn.mx (A. Rivera-García); dramirezr1607@alumno.ipn.mx (D. Ramírez-Rosas); cvgarcia@ipn.mx (C. García-Mendoza)

18 media outlets, which was manually labeled as either clickbait or non-clickbait. The authors report a baseline machine learning method that achieved 0.84 of the F1-score.

2. Task and corpus description

The work presented in [6] was revisited as a task by the IberLEF 2025 conference [7]. TA1C [8] proposes two substask: Clickbait Detection and Clickbait Spoiling. The first subtask aims to determine whether a tweet's content is related to a news item considered clickbait. The second subtask involves generating or extracting a brief text from the article that fills the information gap, satisfies the curiosity generated, or, conversely, indicates that the article offers no answer on the matter.

We decided to participate solely in the clickbait detection subtask, therefore we will only describe the corpus created for this purpose. The event organizers provided three datasets: `TA1C_dataset_detection_train` (training), `TA1C_dataset_detection_dev_gold` (development), and `TA1C_dataset_detection_test` (test), all in CSV format. The training dataset contains 2,800 instances, while development and test datasets contain 700 instances.

The training and the development dataset includes the following columns: `Tweet ID`, `Tweet Date`, `Media Name`, `Media Origin`, `Teaser Text`, and `Tag Value` (indicating whether the teaser is clickbait or not). The datasets exhibit a class imbalance, with 71% of instances belonging to the ”No clickbait” class and the remaining 29% belonging to the ”Clickbait” class. This imbalance poses a significant challenge for machine learning models, as it often bias their predictions towards the more represented classes. The effect of this issue on the results of the developed models is detailed in Section 4.

3. Method

Two approaches were pursued in developing the clickbait detector. The first relied on traditional Machine Learning methods, while the second used Large Language Models. Below, we describe the processes applied in each approach.

3.1. Traditional Machine Learning methods

Traditional Machine Learning methods (TMLM) require the text to be preprocessed and converted into a suitable representation before use. This approach involved applying diverse text preprocessing and representation techniques.

3.1.1. Preprocessing

Since the corpus consists of tweets, it contains special strings like user mentions, hashtags, and URLs. We performed a cleaning process to remove this information, which was considered unhelpful in identifying clickbait. Additionally, POS tagging was used to identify determiners, prepositions, conjunctions, and pronouns, as words with these grammatical categories were removed because they were considered stopwords. Finally, a lemmatization process was applied to use words in their base form. The SpaCy tool[1] was used for this normalization process.

3.1.2. Text representation

After preprocessing, unigrams, bigrams, and a combination of both were extracted as features. These features were transformed into vector space using the following techniques:

- *Term frequency*: Transforms the text into a word count vector, where each entry indicates how many times a specific word appears in the document. Although simple, it can be effective in tasks where absolute frequency is relevant.

[1] https://spacy.io/

- *Binarization*: Similar to the previous method but only indicates the presence or absence of a word, disregarding its frequency. This helps reduce the impact of extremely frequent words.
- *Term Frequency - Inverse Document Frequency (TF-IDF)*: This approach weighs words based on their relative frequency within a document compared to the rest of the document collection. Common words across the entire collection are assigned a lower weight, allowing terms more representative of each class or category to stand out.

Details on how preprocessing and text representation techniques were used into the pipeline can be found in Section 4.

3.1.3. Classification

The task for determining whether a tweet is clickbait or not was was tackled as a binary classification problem. We use the following machine learning methods: Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Support Vector Machine (SVM), Multi-layer Perceptron(MLP), Random Forest (RF), and Gradient Boosting (GB). These methods were chosen based on their established efficacy in the text classification task [9]. For the implementation of the classifiers the `scikit-learn` library [10] was used. Further details regarding the training process, hyperparameter tuning and results can be found in Section 4.

3.2. Large Language Models

In addition to traditional machine learning approaches, we explored the use of Large Language Models (LLMs), which have demonstrated strong performance accross a range of Natural Language Processing (NLP) tasks due to their capacity to capture deep semantic and contextual information [11]. For this task, we fine-tuned two pre-trained transformer-based models: a Spanish version of BERT [12] known as BETO [13] and RoBERTa [14]. Unlike classical models that rely on predefined text vectorization methods, these models learn contextual representations of text directly from raw input. They use self-attention mechanisms to capture semantic relationships between words, making them highly effective for tasks involving subtle linguistic cues such as detecting clickbait.

To optimize the training process and reduce computational costs, we applied the **LoRA (Low-Rank Adaptation)** technique [15]. LoRA enables parameter-efficient fine-tuning by injecting trainable rank decomposition matrices into the transformer layers, allowing us to adapt large models to our specific dataset without retraining the entire model.

4. Experiments and Results

In this section, we present the experiments and results of the clickbait detection task. As explained above, we used TMLM and LLMs. For all of the experiments, we used the train and gold development corpora described in Section 2. The column *Teaser Text* was provided as data input and column *Tag Value* as a target for the classifiers.

4.1. TMLM experiments

The training corpus was split into 75% of the data for training and 25% for development. The gold development corpus was used as a test set. The corpus analysis revealed a substantial class imbalance, with the 'No clickbait' class being overrepresented (71% of instances). Consequently, a stratified partitioning method was employed to ensure that the training and development sets accurately reflect the class distribution in the original corpus.

For the experiments we tried several combinations of the preprocessing and text representation techniques describen in subsections 3.1.1 and 3.1.2. GridSearch was employed to conduct a systematic search for optimal hyperparameters for the classifiers. The hyperparameters adjusted for each machine learning method are listed below.

- LR: max_iter and penalty.
- SVM: kernel, C and Gamma
- MLP: hidden_layer_sizes, alpha and solver
- RF: n_estimators, max_depth, min_samples_leaf and min_samples_split
- GB: n_estimators, learning_rate, subsample and max_depth

Table 1 shows the performance of the classifiers in our set of development (25% of the training corpus) with the best settings.

ML Method	Best Hyperparameters	Preprocessing	Text Representation	Average F1-score (Macro)
LR	max_iter=200, penalty=’l2’	cleaning + lemmatization	Unigram + Bigram (Term Frequency)	0.842
SVM	kernel=’rbf’, C=10, Gamma=’scale’	cleaning + lemmatization	Unigram + Bigram (Term Frequency)	0.847
MLP	hidden_layer_sizes=(150,), alpha=0.0001, solver=’adam’, max_iter=2000	cleaning + lemmatization	Unigram + Bigram (Term Frequency)	**0.869**
RF	n_estimators=200, max_depth=None, min_samples_split=5, min_samples_leaf=1	cleaning + lemmatization	Unigram + Bigram (Term Frequency)	0.861
GB	n_estimators=200, learning_rate=0.1, subsample=0.8, max_depth=3	cleaning + lemmatization	Unigram + Bigram (Term Frequency)	0.858

Table 1
Configurations with the highest average F1-score (Macro) for the clickbait detection task. Hyperparameters were optimized using Grid Search.

As observed in the table, text cleaning and lemmatization proved to be the most effective preprocessing steps across all machine learning methods. Stop word removal did not enhance performance, which we attribute to the short text lengths being further diminished by their exclusion. For text representation, unigram and bigram frequency counts yielded the best results. This suggests that the presence of individual words and two-word sequences helps identify relevant features in the texts.

Subsequently, the best-performing model (MLP) was retrained using 100% of the training corpus data. The adjusted model was then used to predict instances in the gold development corpus, achieving a F1-score (Macro) of 0.649. The model’s performance indicates limited generalization power, and based on our analysis, this is primarily due to the class imbalance. Given the poor results, we decided not to use this model for predictions in the test corpus.

4.2. LLMs experiments

Given the nature of the task, LLMs offer a significant advantage by capturing subtle semantic and syntactic cues that traditional models may overlook. This is particularly beneficial in our case, as the corpus presented a notable class imbalance, with a greater number of non clickbait instances. LLMs have been shown to be more robust in handling unbalanced datasets due to their contextual understanding and capacity to learn from limited examples of the minority class.

A fine-tunning process was performed on three LLMs models:

- `dccuchile/bert-base-spanish-wwm-cased` (BETO)
- `PlanTL-GOB-ES/roberta-base-bne` (RoBERTa)
- `PlanTL-GOB-ES/roberta-base-bne + LoRA` (RoBERTa_LoRA)

The input text was tokenized using each model’s specific tokenizer, with sequences truncated to the model’s maximum length (512 for both BERT and RoBERTa) and padding activated. We did not apply text normalization techniques (e.g., text cleaning, stop words removal and lemmatization) before feeding the input into BERT and RoBERTa. This decision is based on the fact that these models are pre-trained on raw text, and altering the input could introduce a distributional mismatch that affects performance. Since tokenizers for BERT and RoBERTa are designed to handle casing, punctuation, and subword variations, preserving the original text allows the models to leverage their pretraining fully.

We used 100% of the `TA1C_dataset_detection_train.csv` corpus for training and 100% of the `TA1C_dataset_detection_dev_gold.csv` corpus for evaluation. Different hyperparameters were tested, and the best configuration found is detailed in Table 2.

Model	Hyperparameters
BETO	eval_strategy="steps", eval_steps=100, num_train_epochs=3, seed=0, learning_rate=5e-5
RoBERTa	eval_strategy="steps", eval_steps=100, num_train_epochs=10, seed=0, learning_rate=5e-5
RoBERTa_LoRA	eval_strategy="steps", eval_steps=100, num_train_epochs=10, seed=0, learning_rate=5e-5

Table 2
Hyperparameters used to train the LLMs

Below are the results obtained by the three LLMs on the gold development set. In Tables 3, 4 and 5 we show the classification report of the three fine-tuned LLMs.

Class	Precision	Recall	F1-score	Support
Clickbait	0.86	0.81	0.83	203
No clickbait	0.92	0.95	0.93	497
Accuracy			0.91	700
Macro avg	0.89	0.88	0.88	700
Weighted avg	0.90	0.91	0.90	700

Table 3
Results of BETO model on the gold development set.

Class	Precision	Recall	F1-score	Support
Clickbait	0.86	0.82	0.84	203
No clickbait	0.93	0.95	0.94	497
Accuracy			0.91	700
Macro avg	0.90	0.88	0.89	700
Weighted avg	0.91	0.91	0.91	700

Table 4
Results of RoBERTa model on the gold development set.

Class	Precision	Recall	F1-score	Support
Clickbait	0.89	0.83	0.86	203
No clickbait	0.93	0.96	0.95	497
Accuracy			0.92	700
Macro avg	0.91	0.89	0.90	700
Weighted avg	0.92	0.92	0.92	700

Table 5
Results of RoBERTa_LoRA model on the gold development set.

The RoBERTa model fine-tuned with LoRA obtained the best results based on the results. We can see that this model's macro F1-score was 0.90, representing a 38% improvement compared to the 0.649 achieved by the best TMLM (MLP). Additionally, despite the imbalance in the "No clickbait" class, the LLM achieved an F1-score of 0.86. This result shows that the bias towards the majority class has been significantly reduced.

4.3. Competition test set results

The RoBERTa_LoRA model was used to generate predictions on the final, unlabeled test file provided for the competition. The same tokenization method was applied to the 700 instances of the test set. According to the results published by the competition organizers, our model achieved a macro F1-score of 0.815249, placing us in second place, just 0.00039 behind first place. Table 8 shows the official competition results.

Place	User	Score
1	tomasbernal01	0.81564
2	**escom**	**0.81525**
3	viahes	0.81482
4	dcere	0.80480
5	julian_zsa	0.80347
6	gsdeyson	0.80115
7	Omar.Garcia	0.79558
8	gaspai	0.77748
9	danielrod99	0.77562
10	John94	0.75380
11	noetorres	0.75380
12	ChristianRuizU	0.74636
13	Carmen_Garcia	0.73529

Table 6
Results of RoBERTa_LoRA model on the test set.

5. Conclusions and future work

Clickbait has become a widespread advertising strategy designed to drive traffic to websites. While the information provided through this mechanism can sometimes be interesting, clickbait text often exaggerates or provides incomplete information, enticing users to follow the link. Task TA1C, proposed as part of the IBERLEF 2025 conference, aims to detect clickbait text automatically. During the competition, we developed a machine learning model that ranked second in correctly identifying both clickbait and non-clickbait texts. The use of the RoBERTa LLM, coupled with LoRA for parameter fine-tuning, significantly improved the results compared to traditional machine learning methods. Data imbalance was a challenge due to the common bias present in models; however, our LLM successfully overcame this situation.

In future work, we propose data augmentation for the minority class (clickbait) using an LLM for text generation, given that traditional oversampling techniques were ineffective in our experiments. Additionally, we can explore the use of more robust LLMs like DeepSeek or Gemini and employ prompting techniques to guide the models in this task.

Acknowledgments

This research was partially funded by SECIHTI-SNII and Instituto Politécnico Nacional (IPN), through grants SIP-20254348, COFAA-SIBE and EDI.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini and Grammarly to: check grammar and spelling. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

[1] A. Puthussery, Digital marketing: an overview (2020).

[2] K. K. Kapoor, Y. K. Dwivedi, N. C. Piercy, Pay-per-click advertising: A literature review, The Marketing Review 16 (2016) 183–202.

[3] A. B. Araujo, M. F. N. Jaso, J. Serrano-Puche, Use of clickbait in spanish digital native media. an analysis of el confidencial, el español, eldiario. es and ok, Dígitos. Revista de Comunicación Digital (2021) 185–210.

[4] Supriya, J. P. Singh, G. Kumar, Identification of clickbait news articles using sbert and correlation matrix, Social Network Analysis and Mining 13 (2023) 153.

[5] A. Muqadas, H. U. Khan, M. Ramzan, A. Naz, T. Alsahfi, A. Daud, Deep learning and sentence embeddings for detection of clickbait news from online content, Scientific Reports 15 (2025) 13251.

[6] G. Mordecki, G. Moncecchi, J. Couto, Te ahorré un click: A revised definition of clickbait and detection in spanish news, in: L. Correia, A. Rosá, F. Garijo (Eds.), Advances in Artificial Intelligence – IBERAMIA 2024, Springer Nature Switzerland, Cham, 2025, pp. 387–399.

[7] J. Á. González-Barba, L. Chiruzzo, S. M. Jiménez-Zafra, 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, 2025.

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

[9] A. Gasparetto, M. Marcuzzo, A. Zangari, A. Albarelli, A survey on text classification algorithms: From text to predictions, Information 13 (2022). URL: https://www.mdpi.com/2078-2489/13/2/83. doi:10.3390/info13020083.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[11] L. Tunstall, L. Von Werra, T. Wolf, Natural language processing with transformers, " O'Reilly Media, Inc.", 2022.

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), 2019, pp. 4171–4186.

[13] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, in: PML4DC at ICLR 2020, 2020.

[14] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, CoRR abs/1911.02116 (2019). URL: http://arxiv.org/abs/1911.02116. arXiv:1911.02116.

[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, in: International Conference on Learning Representations, 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.