

Studies in Big Data 134

Gilberto Rivera
Alejandro Rosete
Bernabé Dorronsoro
Nelson Rangel-Valdez *Editors*

Innovations in Machine and Deep Learning

Case Studies and Applications

 Springer

Studies in Big Data

Volume 134

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland


Gilberto Rivera · Alejandro Rosete ·
Bernabé Dorronsoro · Nelson Rangel-Valdez
Editors


Innovations in Machine and Deep Learning

Case Studies and Applications

 Springer

Editors

Gilberto Rivera 
División Multidisciplinaria de Ciudad
Universitaria
Universidad Autónoma de Ciudad Juárez
Chihuahua, Mexico

Bernabé Dorronsoro 
School of Engineering
University of Cadiz
Cádiz, Spain

Alejandro Rosete 
Universidad Tecnológica de La Habana
“José Antonio Echeverría”
La Habana, Cuba

Nelson Rangel-Valdez 
Instituto Tecnológico de Ciudad Madero
Tecnológico Nacional de México
Tamaulipas, Mexico

ISSN 2197-6503

ISSN 2197-6511 (electronic)

Studies in Big Data

ISBN 978-3-031-40687-4

ISBN 978-3-031-40688-1 (eBook)

<https://doi.org/10.1007/978-3-031-40688-1>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2023

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Analytics-Oriented Applications

Recursive Multi-step Time-Series Forecasting for Residual-Feedback Artificial Neural Networks: A Survey	3
Waddah Saeed and Rozaida Ghazali	
Feature Selection: Traditional and Wrapping Techniques with Tabu Search	21
Laurentino Benito-Epigmenio, Salvador Ibarra-Martínez, Mirna Ponce-Flores, and José Antonio Castán-Rocha	
Pattern Classification with Holographic Neural Networks: A New Tool for Feature Selection	39
Luis Diago, Hiroe Abe, Atsushi Minamihata, and Ichiro Hagiwara	
Reusability Analysis of K-Nearest Neighbors Variants for Classification Models	63
José Ángel Villarreal-Hernández, María Lucila Morales-Rodríguez, Nelson Rangel-Valdez, and Claudia Gómez-Santillán	
Speech Emotion Recognition Using Deep CNNs Trained on Log-Frequency Spectrograms	83
Mainak Biswas, Mridu Sahu, Maroi Agrebi, Pawan Kumar Singh, and Youakim Badr	
Text Classifier of Sensationalist Headlines in Spanish Using BERT-Based Models	109
Heber Jesús González Esparza, Rogelio Florencia, José David Díaz Román, and Alejandra Mendoza-Carreón	
Arabic Question-Answering System Based on Deep Learning Models	133
Samah Ali Al-azani and C. Namrata Mahender	

Healthcare-Oriented Applications

Machine and Deep Learning Algorithms for ADHD Detection: A Review	163
Jonathan Hernández-Capistran, Laura Nely Sánchez-Morales, Giner Alor-Hernández, Maritza Bustos-López, and José Luis Sánchez-Cervantes	
Mosquito on Human Skin Classification Using Deep Learning	193
C. S. Ayush Kumar, Advaith Das Maharana, Srinath Murali Krishnan, Sannidhi Sri Sai Hanuma, V. Sowmya, and Vinayakumar Ravi	
Analysis and Interpretation of Deep Convolutional Features Using Self-organizing Maps	213
Diego Sebastián Comas, Gustavo Javier Meschino, Agustín Amalfitano, and Virginia Laura Ballarin	
A Hybrid Deep Learning-Based Approach for Human Activity Recognition Using Wearable Sensors	231
Deepak Sharma, Arup Roy, Sankar Prasad Bag, Pawan Kumar Singh, and Youakim Badr	
Predirol: Predicting Cholesterol Saturation Levels Using Big Data, Logistic Regression, and Dissipative Particle Dynamics Simulation	261
Reyna Nohemy Soriano-Machorro, José Luis Sánchez-Cervantes, Lisbeth Rodríguez-Mazahua, and Luis Rolando Guarneros-Nolasco	
Convolutional Neural Network-Based Cancer Detection Using Histopathologic Images	287
Jayesh Soni, Nagarajan Prabakar, and Himanshu Upadhyay	
Artificial Neural Network-Based Model to Characterize the Reverberation Time of a Neonatal Incubator	305
Virginia Puyana-Romero, Lender Michael Tamayo-Guamán, Daniel Núñez-Solano, Ricardo Hernández-Molina, and Giuseppe Ciaburro	
A Comparative Study of Machine Learning Methods to Predict COVID-19	323
J. Patricia Sánchez-Solís, Juan D. Mata Gallegos, Karla M. Olmos Sánchez, and Victoria González Demoss	
Sustainability-Oriented Applications	
Multi-product Inventory Supply and Distribution Model with Non-linear CO₂ Emission Model to Improve Economic and Environmental Aspects of Freight Transportation	349
Santiago Omar Caballero-Morales, Jose Luis Martinez-Flores, and Irma Delia Rojas-Cuevas	

Convolutional Neural Networks for Planting System Detection of Olive Groves	373
Cristina Martínez-Ruedas, Samuel Yanes Luis, Juan Manuel Díaz-Cabrera, Daniel Gutiérrez Reina, Adela P. Galvín, and Isabel Luisa Castillejo-González	
A Conceptual Model for Analysis of Plant Diseases Through EfficientNet: Towards Precision Farming	401
Roneeta Purkayastha and Subhasish Mohapatra	
Ginger Disease Detection Using a Computer Vision Pre-trained Model	419
Olga Kolesnikova, Mesay Gemedo Yigezu, Atnafu Lambebo Tonja, Michael Meles Woldeyohannis, Grigori Sidorov, and Alexander Gelbukh	
Anomaly Detection in Low-Cost Sensors in Agricultural Applications Based on Time Series with Seasonal Variation	433
Adrián Rocha Íñigo, José Manuel García Campos, and Daniel Gutiérrez Reina	
Coconut Tree Detection Using Deep Learning Models	469
Deepthi Sudharsan, K. Harish, U. Asmitha, S. Roshan Tushar, H. Theivaprakasham, V. Sowmya, V. V. Sajith Variyar, Krishnamoorthy Deva Kumar, and Vinayakumar Ravi	
Hybrid Neural Network Meta-heuristic for Solving Large Traveling Salesman Problem	489
Santiago Omar Caballero-Morales, Gladys Bonilla-Enriquez, and Diana Sanchez-Partida	

Text Classifier of Sensationalist Headlines in Spanish Using BERT-Based Models



Heber Jesús González Esparza, Rogelio Florencia, José David Díaz Román, and Alejandra Mendoza-Carreón

Abstract Information technologies play a crucial role in keeping society informed during global events like pandemics. However, sensational headlines can negatively impact public perception and trust in institutions. In this chapter, several BERT-based text classifiers were developed to classify sensational and non-sensational health-related headlines in Spanish. The models were fine-tuned on almost 2000 headlines from major Mexican newspapers, achieving up to 94% F1-Score and accuracy. This demonstrates the effectiveness of machine learning techniques in detecting sensationalism in news headlines.

Keywords Natural language processing · Machine learning · Deep learning · Sensationalism · Short-text classification

1 Introduction

Global pandemics are among the most significant challenges society faces. One of the most important roles when a health crisis occurs is to inform and educate the public on ways of mitigating it [1]. Still, several studies suggest that news coverage about health-related issues does not always pursue this objective [2–5].

In recent years, with more competition than ever, there has been a growing trend among news outlets of packaging news articles in headlines that attempt to capture

H. J. González Esparza · R. Florencia (✉) · J. D. Díaz Román · A. Mendoza-Carreón
División Multidisciplinaria de Ciudad Universitaria, Universidad Autónoma de Ciudad Juárez,
Av. José de Jesús Macías Delgado #18100, C.P. 32000, Ciudad Juárez, Chihuahua, México
e-mail: rogelio.florencia@uacj.mx

H. J. González Esparza
e-mail: gonzalez_heber@outlook.com

J. D. Díaz Román
e-mail: david.roman@uacj.mx

A. Mendoza-Carreón
e-mail: alemendo@uacj.mx

people's attention. This inclination has sometimes been pushed to the limit, resulting in headlines that are inaccurate, misleading, or too emotional.

A sensationalist headline emphasizes elements that could provoke emotional responses rather than focusing on factual information that is valuable to readers [6]. This presents a problem because headlines can significantly shape the reader's worldview [7]. When dealing with health-related issues, for example, an emotional or misleading headline can lead to a change in people's decision-making and their trust in institutions [8, 9].

One of the most important branches of computer science is Natural Language Processing (NLP). Some NLP techniques take advantage of Machine Learning (ML) algorithms for their development, which can be faster and cheaper than doing them manually [10]. This makes ML models a very effective tool when dealing with tasks involving human languages, such as sentiment analysis or text classification [10].

Bidirectional Encoder Representations from Transformers (BERT) is a simple and powerful language representation model that can be used for many different NLP tasks, like text classification [11]. BERT is one of the most popular deep learning-based language models and has been described as a 'quantum leap' in the artificial intelligence and NLP fields [12]. Other models based on its architecture have also been proposed, improving performance in some scenarios.

Only a few projects in Spanish have taken advantage of NLP techniques for developing systems that automatically identify news headlines, even less so when dealing with sensationalism. This chapter presents three text classifiers for detecting sensationalist headlines in Spanish. These models were generated using several BERT-based models and fine-tuned with data collected and labeled manually for this project.

This chapter is structured as follows: Sect. 2 describes the theoretical background of the project. Other projects with similar objectives and comparable approaches are presented in Sect. 3. Section 4 describes the process of collecting, labeling, and analyzing the data used to train and validate the models. It also describes the methods used to build and fit the models, while the performance of the classifiers can be found in Sect. 5. Finally, Sect. 6 concludes the chapter by describing the strengths and limitations of the project.

2 Background

This Section defines crucial concepts necessary for a better understanding of the problem and, therefore, the proposed solution. Section 2.1 describes the usage of the term ‘sensationalism’ in various contexts and the way it can be a harmful kind of journalism. Section 2.2 presents a brief explanation of BERT-based models.

2.1 Sensationalism

When dealing with problems as big as global pandemics, news coverage on these subjects becomes a critical element for solving these issues properly. Emotional tone in news stories can be a decisive factor in influencing people’s risk perceptions, attitudes, and behaviors towards health-related topics [8].

The current market-driven state of mass media has led to health-related news being exaggerated or ‘sensationalized’, aiming to attract and hold the public’s attention. Many different contexts make use of the term ‘sensationalism’. Some authors define it as a type of journalism aimed at the popular classes, where violence, pornography and tragedy are the norms. The most important thing for a sensationalist journalist is to gain the reader’s attention, even if the facts themselves are compromised [13, 14].

Still, there are other circumstances in which this term is also used. One of these cases is when describing the kind of journalism that uses stylistic techniques to provoke emotions, including excitement, fear, and astonishment. The word ‘sensationalism’ is also used to reference the ‘discursive strategy’ of packaging information into headlines with the intention of making the news look more interesting, extraordinary, or relevant [6].

Given the fact that the definition of ‘sensationalism’ heavily depends on context, a more concise description was needed. With the intention of limiting the definition of the term, a list of criteria selected using the opinion of many different authors [5, 6, 15–18] was selected. A headline would be considered sensationalist if it met at least one of the following:

- Omit words that can represent uncertainty (such as ‘could’ or ‘might’).
- Dramatize the information by implying that it was previously hidden from the public (e.g., ‘Are we about to witness the end of Britain?’).
- Make use of superlatives or other extreme words to exaggerate the headline (such as ‘miracle’ or ‘revolutionary’).
- Make use of vivid metaphors to gain the reader’s attention (e.g., referencing zombies in the case of an infectious outbreak).
- Make use of capital letters to emphasize words that can gain the reader’s attention (such as ‘HISTORIA’ in Spanish or ‘HISTORY’ in English).
- Make use of narratives where two groups are opposed to each other (e.g., ‘them’ versus ‘us’).

- Make use of words that are susceptible to alarm people without the need to use them, such as (in Spanish): ‘Apocalyptic’ (‘Apocalíptico’), ‘Emergency’ (‘Emergencia’), ‘Alarm’ (‘Alarma’), ‘Crisis’ (‘Crisis’), ‘Chaos’ (‘Caos’), etc.

As a disclaimer, every single one of the characteristics listed above should not be considered as ‘definitive’. The idea of which headlines can be regarded as sensationalist is far from being universally accepted. All criteria written here are subject to verification, as the classification of a sensationalist headline depends heavily on the perception and opinion of each reader.

2.2 *BERT-Based Models*

BERT is a very popular deep-learning model developed by Google that is based on a neural network known as Transformer [11]. A Transformer, first described in [19], is a combination of two different concepts: Convolutional Neural Networks (CNN) and the Attention mechanism, making models based on Transformers able to learn contextual relationships between words in a text. Transformers were developed as an alternative to Recurrent Neural Networks (RNN) for NLP tasks, primarily because of RNN’s lack of ability to parallelize work and poor performance when dealing with long sentences [20]. Even though Transformers were first intended to be used in translation tasks, several studies have shown that many other NLP tasks, such as text classification, sentiment analysis, emotion recognition, and spam detection, can be performed with very good results using the Transformer architecture, and more specifically the BERT model [12, 21, 22].

One of the key features of BERT is its ability to process text bidirectionally. Meaning that it considers the context of a word in relation to the words that precede and follow it, allowing it to capture the full meaning and context of a sentence. This contrasts with traditional NLP models, which only consider the context of a word in relation to the words that precede it.

The two steps that can be found in the BERT framework are pre-training and fine-tuning. For pre-training, BookCorpus and the English Wikipedia were used as the corpus (around 3,300 M words combined) for the English version of BERT, enabling it to grasp language patterns. This is done by making use of two unsupervised tasks: Masked-Language Modeling (MLM) and Next Sentence Prediction (NSP).

For MLM, the model must predict the 15% of the words of a sentence that were randomly masked at the beginning of the process. This characteristic is what allows BERT to learn a bidirectional representation of the sentence. In the NSP task, the model takes as an input two masked sentences and then must predict if they were next to each other in the original text.

For fine-tuning, task-specific inputs are needed to get better results in the NLP task that the project is aimed to perform. In this case, the fine-tuning step was made using the sensationalist headlines dataset presented in this chapter.

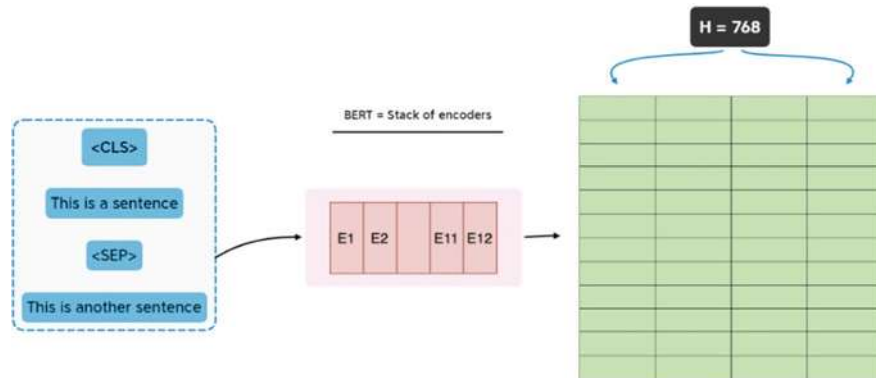


Fig. 1 Diagram representing the BERT architecture

When first released in 2018, BERT had two different versions: BERT-Base and BERT-Large, both of which had ‘uncased’ and ‘cased’ variants (in the uncased version, ‘John Smith’ would become ‘john smith’, while in the cased version this does not happen). BERT-Large makes use of a 24-layer model with 16 attention heads, whereas BERT-Base’s model is ‘only’ 12-layer and has 12 attention heads.

Like the original Transformer encoder, BERT takes a series of words as input. For its functioning, each input must start with the special token *[CLS]* and be separated by *[SEP]*. Each layer of encoders then applies the Self-Attention mechanism before passing its results to the next encoder. This generates a vector of size *hidden_size* (768 in BERT-Base), as seen in Fig. 1. The resulting vector can now be used in many different NLP tasks, including text classification.

Many other models based on the BERT architecture have been proposed to improve performance on specific tasks. In this chapter, experimentation was carried out using three pre-trained models based on BERT:

- **Multilingual BERT-Base** [23]: Pre-trained in the top 104 languages with the largest Wikipedias (including several romances languages, such as French, Portuguese, and Spanish), it delivers slightly worse results than the single-language models of BERT (such as English or Chinese versions). Everything previously stated about BERT is also true for this model, as it is only a multilingual version.
- **BETO-Base (Spanish Pre-trained BERT model)** [24]: A variant of BERT-Base that was pre-trained in a large Spanish corpus making use of the Whole-Word Masking (WWM) technique. In some cases, it has provided better results than the multilingual version of BERT [25].
- **XLNet-Base (Robustly Optimized BERT Pre-training Approach)** [26]: This is a multilingual model that was pre-trained in 100 different languages in a self-supervised way [27]. It is based on RoBERTa and XLM. RoBERTa is a variation of the BERT model that tries to optimize it by using more data, larger batches, and longer training than its predecessor [28]. On the other hand, XLM

is an extension of BERT with two additional purposes: Translation Language Modeling (TLM) and Cross-lingual Language Modeling (XLM). This allows the model to be trained on parallel sentences in different languages and monolingual sentences in other languages.

3 Related Work

In this Section, a description of other projects with similar goals to the ones described in this chapter is presented.

Most of the efforts in NLP over the years have been made having the English language in mind. For this reason, only a few advancements in this area have been accomplished in languages other than English, like Mandarin or Spanish [29–32].

There are a few projects that tackle the text classification task with the usage of ML algorithms, such as Bag of Words or Word2Vec in Spanish, though few, it's important to remark on the existence of several projects that do. Some projects even use some pre-trained model based on BERT as a way of solving their problems.

In [32] sensitive data on several clinical datasets in Spanish was automatically anonymized. They used two datasets for their project, both of which comprised of plain text that contained clinical narrative and manual annotations of sensitive information. To generate the model, they made use of the Base Multilingual Cased version of BERT and the *PyTorch* library. After following two different experiments, each one with different tasks, F1 Scores between 0.925 and 0.979 were obtained.

Another project that used BERT to solve a problem involving natural language can be found in [31]. It consisted of a sentiment analysis task of comments on the Google Play Store. For the fine-tuning task, 15,985 comments were gathered from the 15 most downloaded education apps on the Google Play Store, such as Google Classroom or Duolingo. The generated model got a 0.8 F1-Score when classifying between negative, neutral, and positive comments.

Also, in [33], aggressiveness in Mexican social media was detected using the same three models as the text classifiers presented in this chapter: Multilingual BERT, XLM-RoBERTa, and BETO. Their model was fine-tuned using the MEX-A3T dataset, which consists of more than 7,000 tweets written by Mexican Spanish Speakers (2,110 of which were labeled as aggressive), as well as the OffenseEval [34] data and the HatEval [35] Spanish subset for some experimentation. They used a test set of 3,143 elements, achieving an F1-Score of 79% in their best model.

Even though said projects do not tackle the same problem as the one described in this chapter, other solutions that do can be found in the English language. In [36], a model for quantifying scientific quality and sensationalism of news records mentioning pandemics was built. The sensationalism of news records was measured with the usage of a tool that identified sensationalism through surveys and focus groups that was also partially used in this project [6]. They built a maximum entropy model using a random sample of 500 news records as a training set and then applied the regression to 10,000 randomly selected news records that mentioned pandemics.

The resulting model reached an accuracy of 73% when scoring sensationalism in a testing set consisting of 200 records.

Another project that tackles the same problem was developed during the 2017 edition of Google's annual program, Summer of Code. In this solution [37], they built a non-linear Support Vector Machine (SVM) model. It used features such as punctuation counts, average sentence length and the number of caps letters in a news record to classify them as sensationalist or non-sensationalist. Using a dataset of 16,000 elements, an F1-Score of 0.82 was achieved through a 5-fold cross-validation when training the model only on headlines.

4 Dataset and Methods

Searching the web for a sensationalist and non-sensationalist headlines dataset did not go well, as no dataset that fitted the needs of the project was found. Even less so when searching for a dataset that contained elements in Spanish. As is well known, a vast quantity of data is needed to develop an ML text classifier. With no valuable data to be found on the internet, the decision to collect and manually label the data was taken. Section 4.1 presents the procedures taken for the gathering and labeling of the data necessary to train and validate the model, while Sect. 4.2 describes the generated dataset and its content.

4.1 Data Gathering and Data Labeling

The first requirement necessary before gathering the headlines was to really understand what patterns can be found in sensationalist headlines. As the classification was binary (whether a headline can be considered or not as sensationalist), a headline would be classified as sensationalist if it met one or more of the criteria listed in Sect. 2.

The first step in the gathering process was selecting which news outlets should be considered for the collecting process. In this case, two factors were considered to choose the news sources:

1. Reputation of the news outlet.
2. How easy its website makes it to search and collect headlines.

Among the most important news outlets in Mexico, the ones that had a website that facilitated the data gathering task were Milenio (www.milenio.com) and El Universal (www.eluniversal.com.mx), so, they were both selected for those reasons.

Even though having a good reputation is not an exclusion from being somewhat sensational, a more popularly known news media for its sensationalism could be helpful while gathering this kind of headlines. One of the most prominent and

‘colorful’ news outlets in Mexico is known as ‘La Prensa’ (www.la-prensa.com.mx), and it was also selected.

Following the process of the collecting task, a range of dates on which the headlines would be considered had to be set. To capture as much of the recent events regarding the COVID-19 pandemic as possible, the range of dates that was selected started on the first of January of 2020 and ended on the first of June of 2022.

As the main objective of this project was to build a machine learning model capable of classifying sensationalist and non-sensationalist headlines regarding health-related issues, the list of topics considered when gathering the headlines focused on (but were not limited to) these subjects:

- New cases and accumulated cases of a disease or medical condition.
- New deaths and accumulated deaths caused by a disease or medical condition.
- Discoveries regarding health-related issues (such as new treatments).
- Measures taken by government officials regarding a disease or medical condition.
- Economic or social consequences caused by a disease or medical condition.
- Vaccines.

Web Scrapping techniques were considered, but the decision not to use them came from the fact that manual selection would still be needed. The tasks regarding the collection and labeling process varied from site to site but can be summarized in the next steps:

1. Make use of the site’s search engine to search for articles containing different keywords, such as ‘coronavirus’, ‘cancer’, ‘vaccines’, or ‘dengue’.
2. Set the date range for the search. Usually, it was done month by month, starting with January of 2020 and ending with June of 2022.
3. If the site allowed it, the search was made featuring the most popular articles first. If not, the chronological order was used.
4. Read each headline and manually determine if it met any of the criteria described in Sect. 2, then label it accordingly.
5. Copy the data to a Comma Separated Value (CSV) file.

All data collected was organized into five different columns in the CSV file, four of which were gathered from the news outlets’ websites, including the following:

- ‘encabezado’: The headline, in plain text.
- ‘fecha’: The date the article was published (in format DD-MM-YYYY).
- ‘fuente’: The news outlet from which the headline was gathered, in plain text (‘universal’ for El Universal, ‘milenio’ for Milenio and ‘laprensa’ for La Prensa).
- ‘enlace’: A link to the article featuring the headline.

The fifth and last column featured in the dataset, ‘clase’, can have two different values: ‘1’ if the headline was considered sensationalist, and ‘0’ otherwise.

Given the fact that the COVID-19 pandemic took place in the entirety of the date range selected for this project, and because of its social, political, and medical relevance, most of the headlines collected in the process referenced, in some level, the Coronavirus disease and its consequences.

Table 1 Samples from the dataset

Headline (original)	Headline (translated to English)	Class
“Es un trabajo difícil, muy triste y desgarrador”: enfermera de Hubei sobre brote de coronavirus	“It’s a difficult job, very sad and heartbreaking”: Hubei nurse on coronavirus outbreak	1
Coronavirus genera temor en Europa y paraliza a Italia	Coronavirus generates fear in Europe and paralyzes Italy	1
“¡Mátenla!”, pide hombre en Honduras para presunta enferma de coronavirus	“Kill her!”, asks a man in Honduras for an alleged coronavirus patient	1
¿México...inmune al COVID-19?	Mexico...immune to COVID-19?	1
Parecía el Apocalipsis zombie: tampiqueño que regresó de China por coronavirus	It looked like the zombie apocalypse: man from Tampico who returned from China due to coronavirus	1
Por coronavirus, intensifican medidas preventivas y de higiene en el Metro	Due to coronavirus, preventive and hygiene measures are intensified in the Metro	0
Ebrard anuncia red de América Latina para investigar coronavirus	Ebrard announces Latin American network to investigate coronavirus	0
OMS reúne a 400 expertos para estudiar coronavirus	WHO brings together 400 experts to study coronavirus	0
Episcopado mexicano hace recomendaciones ante Coronavirus	Mexican Episcopate makes recommendations against Coronavirus	0
La Organización Mundial de la Salud abre cuenta en TikTok para informar sobre el coronavirus	The World Health Organization opens an account on TikTok to report on the coronavirus	0

The data collection task only aimed to gather sensationalist and non-sensationalist headlines from the selected newspapers in no particular order or pattern. So, as a disclaimer, all data gathered, labeled, and present in the dataset do not intend to represent the quality of the news coverage each news outlet can offer.

Table 1 contains ten samples drawn from the dataset. The first column shows the headline of the source (in Spanish). The second column is an English translation of each headline. The third column indicates whether a headline was considered sensationalist (1 for sensationalist and 0 otherwise). Columns 1 and 3 were used to train the text classification models.

4.2 Data Analysis

In total, 2,200 headlines were collected and labeled, 1,080 of which were labeled as sensationalist and 1,120 as non-sensationalist. This means the dataset has a nearly 50/50 ratio between its classes and therefore can be considered balanced, as shown in Fig. 2.

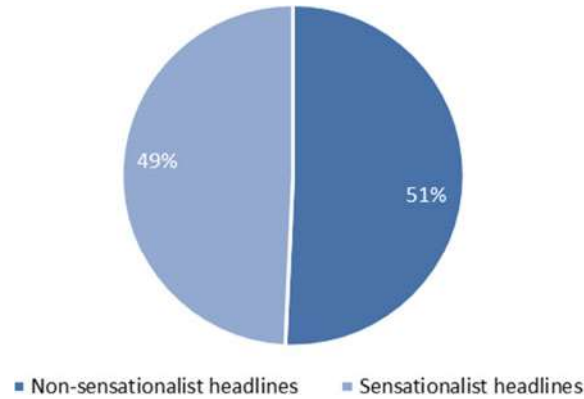


Fig. 2 Ratio between sensationalist and non-sensationalist headlines in the dataset

Figure 3 shows the distribution of sensationalist and non-sensationalist headlines among the three news outlets selected for this project. 502 headlines were collected from 'La Prensa' (www.la-prensa.com.mx), representing 23% of the dataset. Among those 502 headlines, 320 were manually classified as sensationalist (64%), while the remaining 182 were not (36%). In the case of 'El Universal' (www.eluniversal.com), 395 of the headlines gathered from this outlet were classified as sensationalist (47.6%) and 434 were classified as non-sensationalist (52.4%), making up 37% of the dataset. Regarding the headlines gathered from 'Milenio' (www.milenio.com), that represent a 40% of the dataset, 366 of them were classified as sensationalist (42%) and the remaining 503 were not (58%).

After the lemmatization process and the removal of stop words from the dataset, Figs. 4 and 5 display the most used words found in each class. Both classes feature

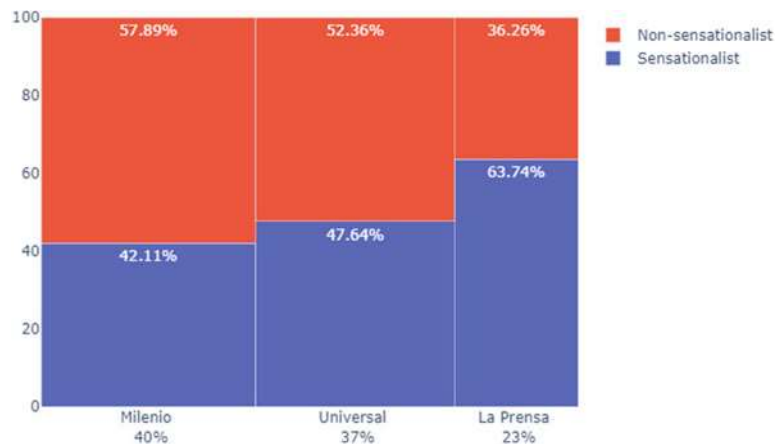


Fig. 3 Class distribution among the news outlets

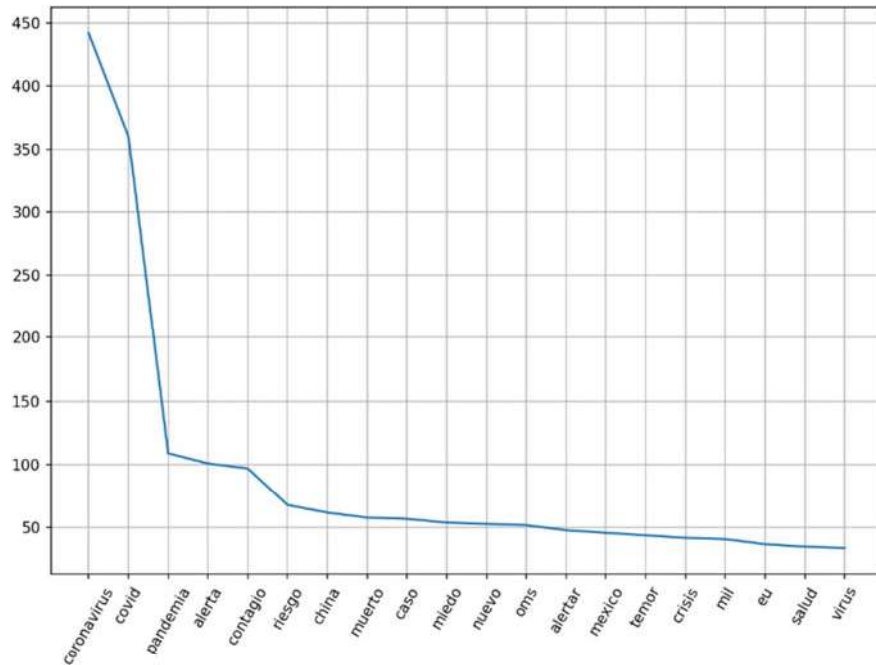


Fig. 4 Most used words in sensationalist headlines from the dataset

a large quantity of headlines of news articles regarding the COVID-19 pandemic coverage. Proof of this can be found in the fact that, in both cases, the two most used words are 'coronavirus' and 'covid', adding up to 1675 combined. In the case of sensationalist headlines, the words 'pandemia' ('pandemic'), 'riesgo' ('risk') and 'contagio' ('contagion'), make up the rest of the five most used words. The words 'caso' ('case'), 'México' ('Mexico') and 'vacuna' ('vaccine') do the same for the non-sensationalist headlines.

Figures 6 and 7 show the word clouds of each class, which highlight the most frequent words in each class by size.

Algorithm 1. Pseudocode of the Multilingual BERT text classifier.

In: Set of news headlines previously labeled as sensationalist and non-sensationalist
Out: A text classifier capable of classifying sensationalist and non-sensationalist news headlines

```

# reading the dataset in a pandas dataframe
1. df ← pd.read('headlines_dataset.csv')
2. df ← df.drop_column('link', 'source', 'date')
# creation of training, testing and validation sets
3. training_set, validation_set, testing_set ←
   train_test_split(df)
# converting sets into bert features using ktrain
4. preprocess_mode ← 'bert'
5. class_names ← ['0', '1']
6. training_set_bert, validation_set_bert, preproc ←
   texts_from_array(training_set, validation_set,
   class_names, preprocess_mode, 'es')
# getting a text classification model and a learner in-
# stance from ktrain
7. model ← text_classifier(preprocess_mode, train-
   ing_set_bert, preproc)
8. learner ← get_learner(model, training_set_bert, valida-
   tion_set_bert)
# finding the best learning rate to train the model
9. learning_rate ← learner.lr_find( )
10. learner.autofit(learning_rate)
# validation process
11. confusion_matrix ← learner.validate(validation_set_bert)
12. metrics ← calculate_metrics(confusion_matrix)
13. print(metrics)
# getting a predictor instance to make predictions on un-
# labeled data
14. predictor ← get_predictor( learner.model, preproc)
# making predictions on the testing set and getting met-
# rics
15. predictions ← predictor.predict(testing_set)
16. metrics ← calculate_metrics(predictions)
17. print(metrics)

```

Line 1 of Algorithm 1 deals with the process of creating a data frame object using the *pandas* library. This object was named *df* and contained all the collected headlines, as well as a label that determined if a headline was considered sensationalist or not. The link, date, and publisher of each news article were also included in the data set.

Given the fact that the only two columns necessary to train and validate the proposed ML model were the headline itself and the label, Line 2 deals with the process of dropping the rest of the described columns from the *df* object.

As seen in Line 3, the data frame object is divided into three different sets: a training set, a validation set, and a testing set. The testing set contained 10% of the

proposed dataset (220 headlines) and was used to evaluate the model once the fine-tuning phase was finished. On the other hand, training and validation sets comprise the remaining 90% (1980 headlines). This is done by using the *sklearn* function *train_test_split*, and its purpose is to allow a good validation process.

In Lines 4–6, the first usage of the *ktrain* library can be seen with the usage of the *text_from_arrays* function, which loads and preprocesses text data from arrays. This function has several parameters, two in particular that determine that the task needed at the time is text classification using the BERT model: *class_names* (if empty, a regression task is assumed) and *preprocess_mode* (with three possibilities: 'standard', 'distilbert' and 'bert'). This function also takes as parameters the training and validation sets generated in Line 3. It is also important to mention that the 'lang' parameter is used to define the language. This parameter can be autodetected, but in this case it was manually set to 'es' (Spanish).

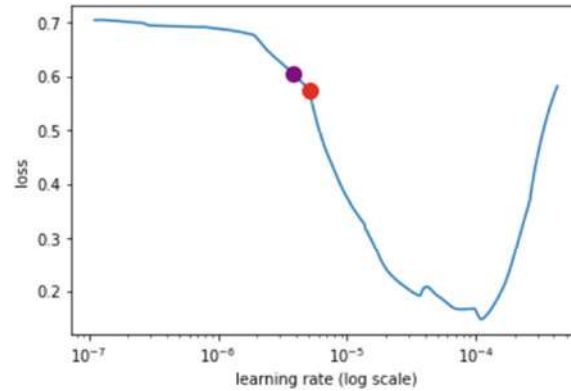
This function provides a way of preprocessing the text using a method called Word-Piece tokenization, a technique that splits words into smaller elements (called word-pieces). Other preprocessing, such as lemmatization or stop word removal, is generally unnecessary when dealing with models based on the Transformer architecture, as its usage can cause a loss of context.

Lines 7 and 8 make use of two very important functions in the development of a text classifier using the *ktrain* library: *text_classifier* and *get_learner*. The *text_classifier* function builds and returns a text classification model and takes as arguments the type of text classifier needed (in this case, 'bert'), the training data, and a preprocessing variable (*preproc*) that was generated in Line 6. On the other hand, the *get_learner* function returns a Learner instance that can be used to train and tune *Keras* models (such as in this case). It takes as arguments the model just generated (*model*) and the training and validation data.

One of the advantages of using *ktrain* to generate, train and validate a machine learning model is the availability of the *lr_find* function, as it simulates training and then plots loss as the learning rate is increased. The function works by training a model on a small portion of the data using a range of learning rates. Then, it plots and determines the learning rate that results in the lowest loss. According to the *ktrain* library documentation, which can be found on [39], the highest learning rate that corresponds to a still falling loss (as displayed in the resulting plot) should be chosen. In this case, and with the attribute *suggest* set to true, the resulting plot can be seen in Fig. 8, with the red and purple dots representing the suggested learning rates by the function for the training phase.

To train the generated model, the *autofit* function is used, as seen in Line 10. In the case of this project, and following the suggestion found in Fig. 8, a learning rate of 4.01e-6 was chosen. The *autofit* function automatically sets *early_stopping* enabled at *patience* = 5, meaning that training will automatically stop after five epochs with no improvement in validation loss. In the same way, *reduce_on_plateau* is also automatically enabled at *patience* = 2, which reduces the learning rate when validation loss does not improve after two epochs. Both *early_stopping* and *reduce_on_plateau* are optional parameters in the *autofit* function and can be edited if needed.

Fig. 8 Resulting plot of the *lr_find* function, which shows loss as learning rate is increased



Lines 11–13 deal with the validation process of the model, which is done by making use of the *validate* function, which is also part of the *ktrain* library. This function takes as an argument the validation set that was generated in Line 6 and returns a confusion matrix. After following several processes regarding the metrics selected for validating this project (such as precision, recall and F1-Score) and using the resulting confusion matrix, these metrics were calculated to then be displayed in a classification report, as seen in Line 13.

Finally, in Line 14, a *predictor* instance is created with the purpose of making predictions on unlabeled data. The *get_predictor* function takes as an argument the previously trained *Keras* model and the same *preproc* variable declared in Line 6, which was also used in several other lines during the process. This *predictor* instance can be saved to disk and then be reloaded as part of other applications using a function called *load_predictor*.

The creation of the *predictor* instance is of help when dealing with the last part of Algorithm 1, seen in Lines 15–17. These lines have the objective of testing the resulting model by using the training set generated in Line 6. The *predict* function in Line 15 saves an array of predictions in the *predictions* variable that can then be compared to the actual labels on the testing set. This process is done in Line 16 and, after following a similar approach as the one described in Line 12, the chosen metrics are displayed in a classification report in Line 17. Results regarding the testing process can be found in Sect. 5.

Algorithms 1 and 2 are similar. The first few lines of Algorithm 2 follow the same idea as those of Algorithm 1. The first change is introduced in Line 5 where the *Transformer* function used to create a *Transformer* object is used. It takes as arguments the name of the Hugging Face pretrained model to use and the *class names*, along with some hyperparameters like *batch_size* and *maxlen*. After some experimentation, *maxlen* was set to 128 in both text classifiers, while *batch_size* was set to 16.

Algorithm 2. Pseudocode of the BETO and RoBERTa text classifiers.

In: Set of news headlines previously labeled as sensationalist and non-sensationalist
Out: A text classifier capable of classifying sensationalist and non-sensationalist news headlines

```

# reading the dataset in a pandas dataframe
1. df ← pd.read('headlines_dataset.csv')
2. df ← df.drop_column('link', 'source', 'date')
# creation of training, testing and validation sets
3. training_set, validation_set, testing_set ←
   train_test_split(df)
# creating a Transformer object
4. class_names ← ['0', '1']
5. transformer ← text.Transformer('roberta/beto',
   class_names)
# preprocessing the training and validation sets
6. train ← transformer.preprocess_train(training_set)
7. val ← transformer.preprocess_test(validation_set)
# creating a classifier object
8. model ← transformer.get_classifier( )
# getting a learner instance from ktrain
9. learner ← get_learner(model, train, val)
# finding the best learning rate to train the model
10. learning_rate ← learner.lr_find( )
11. learner.autofit(learning_rate)
# validation process
12. confusion_matrix ← learner.validate(validation_set_bert)
13. metrics ← calculate_metrics(confusion_matrix)
14. print(metrics)
# getting a predictor instance to make predictions on unlabeled data
15. predictor ← get_predictor(learner.model, transformer)
# making predictions on the testing set and getting metrics
16. predictions ← predictor.predict(testing_set)
17. metrics ← calculate_metrics(predictions)
18. print(metrics)

```

In this case, preprocessing was made in Lines 6 and 7 by using the *preprocess_train* and *preprocess_test* functions. These two functions return objects that can be used by the classifier object *model* created in Line 8 with the *get_classifier* function. The *model* object is based on the Transformer object *transformer* created earlier, so it will use the previously selected pre-trained model, *maxlen*, *batch_size*, and *class_*

Fig. 9 Resulting plot of the *lr_find* function for the BETO classifier

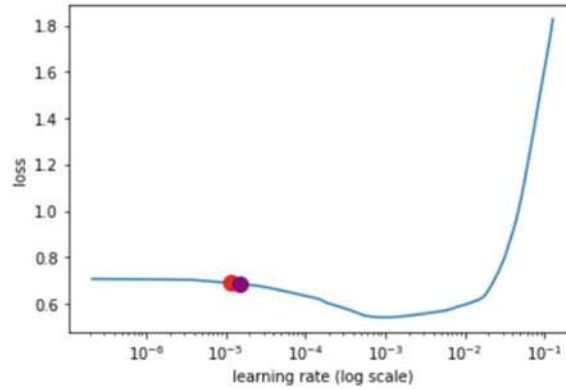
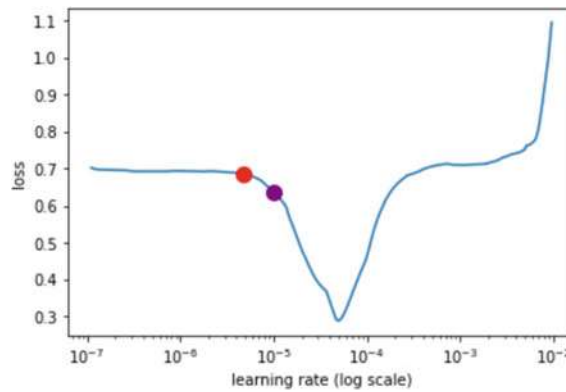


Fig. 10 Resulting plot of the *lr_find* function for the RoBERTa classifier



names as the Transformer. Lines 9–18 are very similar to Algorithm 1, except for Line 15, where the *predictor* instance is created using the *transformer* object created in Line 5 instead of the *preproc* variable used in Algorithm 1. The resulting plots from the *lr_find* function for both classifiers are displayed in Figs. 9 and 10.

5 Results

Performance was evaluated using the test set containing 10% of the data set (220 headlines). Table 2 shows the results obtained by each of the models. The first column shows the name of the three models (Multilingual BERT, BETO, and XLM-RoBERTa). Columns 2, 3, and 4 show each class's F1 score, precision, and recall, where 1 represents sensationalism (the class of interest) and 0 otherwise.

The experimentation results showed that the three text classifiers reached an accuracy above 90%. The classifier that obtained the highest accuracy was

Table 2 Results of the BERT based models after fine-tuning

Model	F1 score		Precision		Recall		Accuracy
	1	0	1	0	1	0	
Multilingual BERT	0.93	0.93	0.92	0.94	0.94	0.92	0.93
BETO	0.93	0.93	0.95	0.91	0.91	0.95	0.93
XLM-RoBERTa	0.94	0.94	0.92	0.96	0.96	0.92	0.94

XLM-RoBERTa with 94%, BETO reached 93%, and multilingual BERT obtained 93%.

Predicting class 1, XLM-RoBERTa achieved 96% on the recall metric, Multilingual BERT 94%, and BETO 91%. When predicting class 0, BETO achieved 95%, XLM-RoBERTa 92%, and Multilingual BERT 92%.

Regarding the precision metric, XLM-RoBERTa and Multilingual BERT reached 92% and BETO 95% in class 1. When predicting class 0, they reached 96%, 94%, and 91%, respectively.

In the F1-Score metric, the best performance was XLM-RoBERTa with 94%, and BETO and Multilingual BERT achieved 93% in class 1 and 94%, 93%, and 93%, respectively, in class 0. A graphic way of showing the results can be seen in Figs. 11, 12 and 13, where the three confusion matrices of the 220 predictions made by each model during the testing phase are presented.

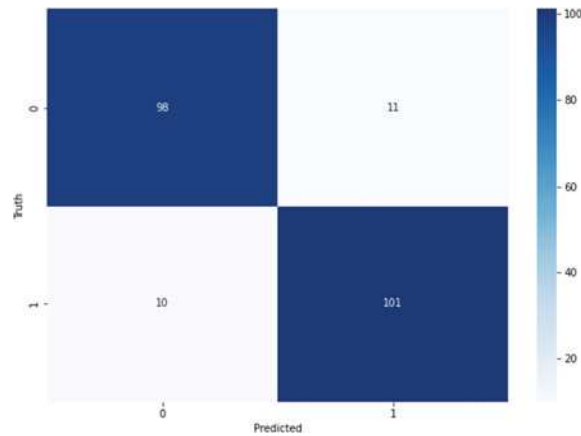
Fig. 11 Confusion matrix of the multilingual BERT model

Fig. 12 Confusion matrix of the BETO model

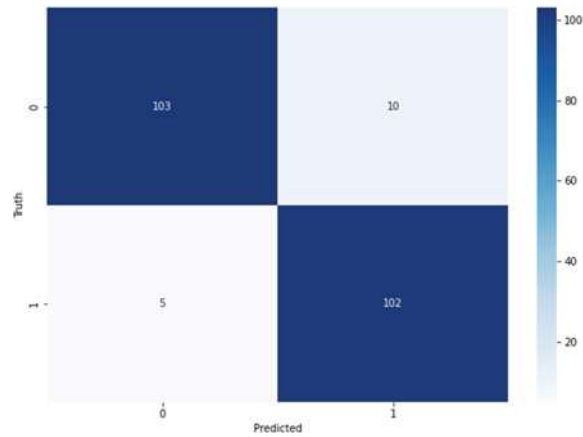
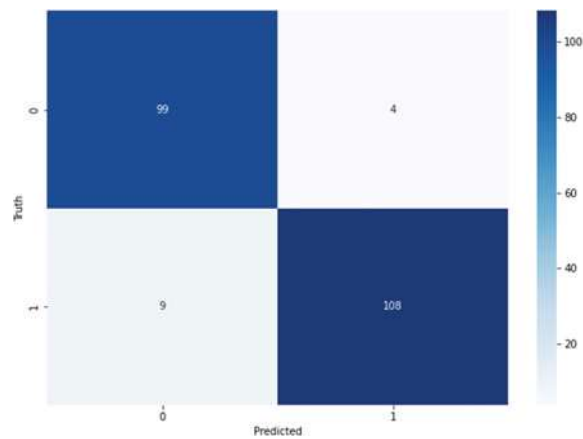


Fig. 13 Confusion matrix of the XLM-RoBERTa model



6 Conclusion

This project explored the problem of detecting sensationalism in written media using three pre-trained models based on the BERT architecture. Previous projects with similar goals focused on detecting sensationalism in news articles written in English. However, with the performance that BERT-based models can offer in other languages, good results can be obtained in classifying news headlines in Spanish.

The experimental results showed that the classifiers presented in this chapter achieved accuracy and F1-Score metrics of 94%, indicating that the models could identify both sensational and non-sensational headlines. Additionally, the classifiers performed well on other metrics, such as accuracy and recall.

The results demonstrate that the tools presented in this chapter have the potential to be used in real-world applications that can help users discern between credible

and sensational sources of information. Therefore, it allows them to make informed decisions about the sources they trust and consume.

As future work, a multiclass classification can be used. Instead of relying on a binary classification, different levels of sensationalism can be determined depending on a set of criteria. This could be potentially helpful if the model's goal is to detect the most harmful headlines from the somewhat sensationalist ones. As one of the main goals of this project was to automatically detect sensationalism in health-related topics, the criteria selected for gathering and labeling headlines was targeted in that direction. Different criteria can be applied to classify news headlines regarding other subjects, such as politics or sports.

Despite limitations (such as a binary classification and the somewhat limited size of the dataset for this type of task), the models presented in this chapter represent an important step forward in promoting the consumption of credible and trustworthy information. It also demonstrates the potential of machine-learning in combating the spread of sensationalism in written media.

References

1. Laing, A.: The H1N1 crisis: roles played by government communicators, the public and the media. *J. Prof. Commun.* **1**(1) (2011). <https://doi.org/10.15173/jpc.v1i1.88>
2. Mach, K.J., et al.: News media coverage of COVID-19 public health and policy information. *Humanit. Soc. Sci. Commun.* **8**(1), 220 (2021). <https://doi.org/10.1057/s41599-021-00900-z>
3. Pieri, E.: Media framing and the threat of global pandemics: the Ebola crisis in UK media and policy response. *Sociol. Res. Online* **24**(1), 73–92 (2019). <https://doi.org/10.1177/1360780418811966>
4. Frangogiannis, N.G.: The significance of COVID-19-associated myocardial injury: how over-interpretation of scientific findings can fuel media sensationalism and spread misinformation. *Eur. Heart J.* **41**(39), 3836–3838 (2020). <https://doi.org/10.1093/eurheartj/ehaa727>
5. Ottwell, R., Puckett, M., Rogers, T., Nicks, S., Vassar, M.: Sensational media reporting is common when describing COVID-19 therapies, detection methods, and vaccines. *J. Investig. Med.* **69**(6), 1256–1257 (2021). <https://doi.org/10.1136/jim-2020-001760>
6. Molek-Kozakowska, K.: Towards a pragma-linguistic framework for the study of sensationalism in news headlines. *Discourse Commun.* **7**(2), 173–197 (2013). <https://doi.org/10.1177/1750481312471668>
7. Waage, H.: *Hyper-reading headlines: how social media as a news-platform can affect the process of news reading*. University of Stavanger (2018)
8. Nabi, R.L., Prestin, A.: Unrealistic hope and unnecessary fear: exploring how sensationalistic news stories influence health behavior motivation. *Health Commun.* **31**(9), 1115–1126 (2016). <https://doi.org/10.1080/10410236.2015.1045237>
9. van Scoy, L.J., et al.: Public anxiety and distrust due to perceived politicization and media sensationalism during early COVID-19 media messaging. *J. Commun. Healthc.* **14**(3), 193–205 (2021). <https://doi.org/10.1080/17538068.2021.1953934>
10. Pedrycz, W., Martínez, L., Espin-Andrade, R.A., Rivera, G., Gómez, J.M. (eds.): Preface. In: *Computational Intelligence for Business Analytics*, pp. v–vi. Springer (2021). <https://doi.org/10.1007/978-3-030-73819-8>
11. Devlin, J., Chang, M.-W., Lee, K., Google, K.T., Language, A.I.: BERT: pre-training of deep bidirectional transformers for language understanding. <https://github.com/tensorflow/tensor2tensor>

12. Koroteev, M.V.: BERT: a review of applications in natural language processing and understanding
13. Pedroso, R.: Elementos para una teoría del periodismo sensacionalista. *Comun. y Soc.* **21**, 139–157 (1994)
14. Torrico, E.: El sensacionalismo: algunos elementos para su comprensión y análisis. *Sala de prensa*, vol. 2, no. 45 (2002)
15. Lin, L.: *Semantic Comparisons for Natural Language Processing Applications*. University of Washington (2021)
16. Doherty, J.-F.: When fiction becomes fact: exaggerating host manipulation by parasites. *Proc. R. Soc. B: Biol. Sci.* **287**(1936), 20201081 (2020). <https://doi.org/10.1098/rspb.2020.1081>
17. Costa-Sánchez, C.: Tratamiento informativo de una crisis de salud pública: Los titulares sobre gripe A en la prensa española. *Revista de Comunicación de la SEECI* **0**(25), 29 (2011). <https://doi.org/10.15198/seeci.2011.25.29-42>
18. Alonso-González, M.: coronavirus a través de los titulares de El Mundo y La Vanguardia. *Revista de Comunicación y Salud* **10**(2), 503–524 (2020). [https://doi.org/10.35669/rcys.2020.10\(2\).503-524](https://doi.org/10.35669/rcys.2020.10(2).503-524)
19. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
20. Giacaglia, G.: Transformers. *Medium* (2019). <https://towardsdatascience.com/transformers-141e32e69591>. Accessed 21 Sep. 2022
21. Özçift, A., Akarsu, K., Yumuk, F., Söylemez, C.: Advancing natural language processing (NLP) applications of morphologically rich languages with bidirectional encoder representations from transformers (BERT): an empirical case study for Turkish. *Automatika* **62**(2), 226–238 (2021). <https://doi.org/10.1080/00051144.2021.1922150>
22. González-Carvajal, S., Garrido-Merchán, E.C.: Comparing BERT against traditional machine learning text classification (2021)
23. bert-base-multilingual-cased · Hugging Face. <https://huggingface.co/bert-base-multilingual-cased>. Accessed 06 June 2023
24. dccuchile/bert-base-spanish-wwm-cased · Hugging Face. <https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased>. Accessed 06 June 2023
25. Că, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., Pérez, J.: Spanish pre-trained Bert model and evaluation data. <https://github.com/josecannete/spanish-corpora>. Accessed 06 Mar. 2023
26. xlm-roberta-base · Hugging Face. <https://huggingface.co/xlm-roberta-base>. Accessed 06 Mar. 2023
27. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale (2019). <https://doi.org/10.48550/arxiv.1911.02116>
28. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach (2019)
29. Dai, Z., Wang, X., Ni, P., Li, Y., Li, G., Bai, X.: Named entity recognition using BERT BiLSTM CRF for Chinese electronic health records. In: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–5 (2019). <https://doi.org/10.1109/CISP-BMEI48845.2019.8965823>
30. Liu, H., et al.: Use of BERT (bidirectional encoder representations from transformers)-based deep learning method for extracting evidences in Chinese radiology reports: development of a computer-aided liver cancer diagnosis framework. *J. Med. Internet Res.* **23**(1), e19689 (2021). <https://doi.org/10.2196/19689>
31. López Condori, J.J., Gonzales Saji, F.O., López Condori, J.J., Gonzales Saji, F.O.: Análisis de sentimiento de comentarios en español en Google Play Store usando BERT. *Ingeniare. Revista chilena de ingeniería* **29**(3), 557–563 (2021). <https://doi.org/10.4067/S0718-33052021000300557>
32. García-Pablos, A., Perez, N., Cuadros, M.: Sensitive data detection and classification in Spanish clinical text: experiments with BERT (2020). <https://doi.org/10.48550/arXiv.2003.03106>
33. Tanase, M.-A., Zaharia, G.-E., Cercel, D.-C., Dascalu, M.: Detecting aggressiveness in Mexican Spanish social media content by fine-tuning transformer-based models (2020). https://www.facebook.com/communitystandards/hate_speech. Accessed 07 Mar. 2023

34. Zampieri, M., et al.: SemEval-2020 task 12: multilingual offensive language identification in social media (OffensEval 2020), pp. 1425–1447 (2020). <http://sites.google.com/site/offensevalsharedtask/offenseval2019>. Accessed 07 Mar. 2023
35. Basile, V., et al.: SemEval-2019 task 5: multilingual detection of hate speech against immigrants and women in twitter, pp. 54–63. <http://evalita.org>. Accessed 07 Mar. 2023
36. Hoffman, S.J., Justicz, V.: Automatically quantifying the scientific quality and sensationalism of news records mentioning pandemics: validating a maximum entropy machine-learning model. *J. Clin. Epidemiol.* **75**, 47–55 (2016). <https://doi.org/10.1016/j.jclinepi.2015.12.010>
37. Ivenskaya, M.: Fake news detection. Google Summer of Code (2017). <https://summerofcode.withgoogle.com/archive/2017/projects/5547741878943744>. Accessed 28 Mar. 2022
38. Maiya, A.S.: ktrain: a low-code library for augmented machine learning (2020). <https://github.com/amaiya/ktrain>
39. Maiya, A.: ktrain API documentation. <https://amaiya.github.io/ktrain/index.html>. Accessed 27 Oct. 2022

DA&CI 2022 - Springer Book notification for paper 3676

DA&CI 2022 - Springer Book <daci2022springerbook@easychair.org>

Mié 2023-02-15 3:55 AM

Para:Rogelio Florencia Juarez <rogelio.florencia@uacj.mx>

Dear Rogelio Florencia,

The review of your chapter, "Text classifier of sensationalist headlines in Spanish using BERT," has just been completed. Although our reviewers find the topic pertinent, they believe you should strengthen the coverage before publishing the chapter.

I have compiled the feedback from reviewer evaluations for your perusal to emphasize particular changes that I feel would be best for you to make to your chapter. Please study the evaluations carefully and let me know if you have any questions about any comments or suggestions.

Once you have completed the revisions, you must upload a PDF file with the following parts:

PART 1. A list of your responses to every single one of the reviewers' comments. Also, when applicable, you should indicate where the revised manuscript addresses the review comments by referencing line numbers.

PART 2. A revised version of your chapter with line numbering. Here, the revisions should be explicitly marked. Also, anonymize your manuscript (by removing any information revealing the authors' identity).

Please, provide this revision by no later than MARCH 15 (2023), uploading the document as an update of your previous submission (<https://easychair.org/conferences/?conf=daci2022springerbook>). Please, be advised that a revision does not guarantee acceptance. The decision regarding the approval of your chapter depends on additional review.

Before you upload the revision, you should:

- (a) Check all requirements and guidelines have been met as outlined in the Manuscript Preparation guide: <https://www.springer.com/de/authors-editors/book-authors-editors/resources-guidelines/book-manuscript-guidelines/manuscript-preparation/5636> (see section "Chapters").
- (b) Use the Word/LaTeX template provided for book chapters. Also, note that "Studies in Big Data" follows the reference style "MathPhys," using reference numbers in square brackets sequentially by citation. We encourage the authors to provide the DOI of the references. For your convenience, I have shared a folder with the (LaTeX and Word) templates and a brief description of the reference style (https://drive.google.com/drive/folders/1HJSs5s2O3C1WGRO95aqZfcqoondC0yDW?usp=share_link).
- (c) Consider an extension of 10,000–16,000 words for the full manuscript.
- (d) Ensure proper use of the English language, formal grammatical structure, and correct spelling and punctuation. If necessary, consult a professional (e.g., <https://www.proof-reading-service.com/>).
- (e) Provide the information of all the chapter authors in the EasyChair Platform.
- (f) Consider the possibility of including (or making publicly available) the code and data (e.g., using GitHub or GoogleColab). The purpose of this suggestion is to promote transparency and reproducibility.

We would like to point out that Remarks (c) and (f) are not mandatory but preferable and much appreciated.

Thank you for your interest and diligent work in your contribution to "Data Analytics and Computational Intelligence: Novel Models, Algorithms and Applications," I greatly value your manuscript and look

forward to seeing your revision! If you have any questions, please do not hesitate to contact me, Gilberto Rivera, at gilberto.rivera@uacj.mx (with a copy to gilberto.rivera@eurekascommunity.org).

SUBMISSION: 3676

TITLE: Text classifier of sensationalist headlines in Spanish using BERT

----- REVIEW 1 -----

SUBMISSION: 3676

TITLE: Text classifier of sensationalist headlines in Spanish using BERT

AUTHORS: Heber González and Rogelio Florencia

----- Overall evaluation -----

SCORE: 2 (Accept after minor revision)

----- TEXT:

This project is innovative because it is working with headlines in Spanish and for this purpose a new extensive database has been created. But with respect to this there are some points to comment:

1. The data could have been extracted automatically, using web scrapping.
2. To annotate them, no tool is mentioned that simplifies the process, such as Prodigy (<https://prodi.gy/>). This would speed up the process and metadata such as the person who annotated them or the exact date could be stored.

Taking this into account, it could have been made faster allowing to annotate a larger dataset.

Regarding training and model selection there are also several things to comment:

1. The best learning rate is searched, but nothing is commented about other hyperparameters such as: dropout, learning rate, batch size, optimizer type, etc.... It would be convenient to make a complete hyperparameters search. Tools such as Weights & Biases (<https://wandb.ai/site>) or MLflow (<https://mlflow.org/>) can be used for this purpose.
2. There are other multilingual models that could be better such as: RoBERTa, XLNet, ALBERT, etc, ...
3. There is also a specific model for Spanish called BETO (<https://github.com/dccuchile/beto>), which usually works better than the multilanguage ones.

Combining these models and doing a correct hyperparameter search should improve the results.

I consider that it is a minor revision because the problem could be studied in more depth. This would require some extra studies, such as the search for hyperparameters and the use of other models, even if they give worse results.

----- REVIEW 2 -----

SUBMISSION: 3676

TITLE: Text classifier of sensationalist headlines in Spanish using BERT

AUTHORS: Heber González and Rogelio Florencia

----- Overall evaluation -----

SCORE: -2 (Weakly reject)

----- TEXT:

The paper presents a model to classify headlines into "sensationalist" or "not sensationalist", the model is based on BERT framework.

The idea to provide a tool to know whether news are sensationalist is good, however the methodology to build the model has some flaws.

I suggest rethinking the features of the dataset, are they enough to know whether a news is sensationalist? The date, the url, or the media are relevant? I suggest analyzing the words and their correlation to the class.

What is the training of the labeling team? Are they students? Are they journalist (or something like that)? How did you state a headline is sensationalist or not?

I would like to see examples of the dataset

There are many steps in building a machine learning, such preprocessing and cleaning data, I think there are several steps missing. I would like to see the NLP process. How did you conduct lemmatization and stop-words removal?

In a scientific paper, the theoretical concepts, ideas, and fundamentals of decisions must be presented, instead to describe code.

The related work must be analyzed deeply

The abstract must provide an overview of the entire paper, from the introduction until conclusions.

Some presentation issues:

The abstract should be just one paragraph.

The paper does not fulfil the format.

First paragraphs of section 2 and section 3 are irrelevant, please delete.

Referencing and numeration of figures are incorrect.

Most of the figures do not provides much information.

La información contenida en este correo electrónico y anexos, está dirigida únicamente para el uso del individuo o entidad a la que fue dirigida y puede contener información propietaria que no es de dominio público. Cualquier uso, distribución o reproducción de este correo que no sea por el destinatario de intención, podría vulnerar la normatividad aplicable.
