IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 July 2023, accepted 5 August 2023, date of publication 30 August 2023, date of current version 7 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3310244

RESEARCH ARTICLE

Offensive Language Detection in Spanish Social Media: Testing From Bag-of-Words to Transformers Models

JOSÉ MARÍA MOLERO, JORGE PÉREZ-MARTÍN, ALVARO RODRIGO, AND ANSELMO PEÑAS

Computer Engineering School, Universidad Nacional de Educación a Distancia (UNED), 28040 Madrid, Spain

Corresponding author: Jorge Pérez-Martín (jperezmartin@dia.uned.es)

This work was supported in part by the Spanish Research Agency (Agencia Estatal de Investigación) through DeepInfo Project (MCIU/AEI/FEDER,UE) under Grant PID2021-127777OB-C22, in part by the Holistic Analysis of Organized Misinformation Activity in Social Networks Project under Grant PCI2022-135026-2, and in part by the Projects Given to the Intelligent Systems for Learning (ISL) Group in the PID 20/21 and 21/22 calls under Grant GID2016-39. The work of Jorge Pérez-Martín was supported in part by the Spanish Government under Grant PID2019-110686RB-I00, and in part by Universidad Nacional de Educación a Distancia (UNED) under Grant 2022V/ITEMP/005.

ABSTRACT Social networks allow us to communicate with people around the world. However, some users usually take advantage of anonymity for writing offensive comments to others, which might affect those who receive offensive messages or discourage the use of these networks. However, it is impossible to manually check every message. This has promoted several proposals for automatic detection systems. Current state-of-the-art systems are based on the transformers' architecture and most of the work has been focused on the English language. However, these systems do not pay too much attention to the unbalanced nature of data, since there are fewer offensive comments than non-offensive in a real environment. Besides, these previous works have not studied the impact on the final results of pre-processing or the corpora used for pre-training the models. In this work, we propose and evaluate a series of automatic methods aimed at detecting offensive language in Spanish texts addressing the unbalanced nature of data. We test different learning models, from those based on classical Machine Learning algorithms using Bag-of-Words as data representation to those based in large language models and neural networks such as transformers, paying more attention to minor classes and the corpora used for pre-training the transformer-based models. We show how transformer-based models continue obtaining the best results, but we improved previous results by a 6,2% by adding new steps of pre-processing and using models pre-trained with Spanish social-media data, setting new state-of-the-art results.

INDEX TERMS Offensive language, natural language processing, transformers-based models.

I. INTRODUCTION

With the rise of social networks, we are more connected but also more exposed to receiving offensive comments because of our ideas, gender, race, or physical condition. Offensive language can be defined as hurtful, derogatory, or obscene comments made by one person to another or a group of people and is related to other concepts such as abusive language, hate speech, cyberbullying, or toxic language [1].

As shown in their reports, social networks are aware of the problem and are trying to implement effective mechanisms to mitigate its effect. For example, Instagram[1] reported in 2023 the deletion of 5.1 million messages with hate speech between January and March, 95.30% of them detected before being reported by users. Between July and December 2020, Twitter[2] removed 1.2 million accounts for violating its hate speech policy, during the period between July and

The associate editor coordinating the review of this manuscript and approving it for publication was Alberto Cano.

[1] https://transparency.fb.com/data/community-standards-enforcement/hate-speech/facebook/

[2] https://time.com/6080324/twitter-hate-speech-penalties/

December 2021, the last period published by the company in its transparency portal, the company suspended 104,565 accounts.[3] Although there is no recent official reports in Twitter's transparency portal, according to [2], hate speech on Twitter appears to have increased significantly in the last year. From January to March 2023, YouTube[4] removed 6 million hate speech comments, almost 178 thousand videos and 51 thousand channels.

The use of language to cause harm to third parties does not only affect these social networks, it also affects instant messaging applications, review pages, or more traditional media such as forums. In short, it is a problem that affects many people on a wide range of platforms and involves a very high volume of data. It is an important problem because receivers of offensive messages could suffer stress or other mental health problems [3], [4]. Given the magnitude of the problem, and the impossibility of manually checking all the social-media messages, a great deal of effort is being devoted to research and the development of systems that automatically detect comments with offensive language, whether it is hate speech or cyberbullying [5], [6].

There have been several evaluation tasks oriented to this issue [7], [8], especially for the English language. One of the most recent tasks has been MeOffendES at IberLEF 2021 [9], where the organizers proposed to detect offensive language in Spanish social-media. We focus on this collection given its novelty and the fact that it has been created in Spanish, where there are available fewer language resources for dealing with Natural Language Processing (NLP) problems. Besides, this collection reflects the unbalanced nature of the problem in social networks, where there are fewer offensive than non-offensive messages. This distribution must be tackled by detection systems, which might find problems to learn the main features of offensive messages given the low proportion of these messages in training collections. However, previous studies have not paid too much attention to it.

In this paper, we evaluate a variety of systems, from classical Machine Learning models based on Bag-of-Words representations to the more recent transformers-based models. We study the main strengths and weaknesses of each approach and offer clear insights into their performance. We focus on studying the ability of systems for detecting messages of minor classes, given the unbalanced nature of the data. In our study, we outperform the best-reported results for offensive language detection in Spanish, according to the MeOffendES task, setting new state-of-the-art results.

The main contributions of this paper are as follows:

- We perform a systematic evaluation of different types of machine learning approaches to the problem of offensive language detection in Spanish.
- We compare the impact on results of pre-training transformer-based models using different corpora.
- We analyze the impact of pre-processing in the best of the proposed systems.
- We obtain the best results for detecting offensive language using the MeOffendES dataset.

The rest of this paper is organized as follows: Section II describes the main state of the art for detecting offensive language. In Section III, we introduce the dataset and evaluation measures used in this paper. Section IV contains the pre-processing steps applied to the input data, while the models used for detecting offensive language are detailed in Section V. We show and analyze the results in Section VI. Finally, we include the main conclusions and future work in Section VII.

II. PREVIOUS WORK

In this Section, we describe the main relevant works related to our paper. We first describe the main approaches proposed for detecting offensive language. Then, we survey the main evaluation campaigns aimed at detecting offensive language with special attention to the evaluation of models targeting the Spanish language. Finally, we include in Table 1 a summary of the main evaluation campaigns reviewed in this paper.

A. OFFENSIVE LANGUAGE DETECTION

Despite the recent interest raised by the use of social networks, the detection of offensive language has been also studied in the past. For example, some authors used an architecture called Lexical Syntactic Feature (LFS) for the detection of both offensive comments and potentially offensive users [10]. They propose a two-phase method: the first phase involves obtaining lexical and syntactic characteristics of each sentence using data mining and natural language processing techniques. In the second phase, they incorporate user-level characteristics calculated by analyzing the author's behavioral patterns. To measure the offense level of a sentence, they use a lexicon of offensive words and syntactic rules that regulate the intensification of the offense. To measure how offensive a user is, they aggregate the level of offensiveness of the user's message history and combine it with other characteristics such as the writing style. With this set of characteristics, they made various experiments with different variations of the dataset. In the first experiment, strong and weak offensive words were included, but the method developed was not as effective as using a method based solely on offensive words. In the second experiment, only weak offensive words were used, this time the method proposed was the best. The authors concluded that in the absence of offensive words, it is necessary to interpret the context to detect offensive language, and this ability is what they have managed to develop with their method.

Before the explosion of deep-learning methods, which are the current trend, other approaches were based on classic machine-learning techniques like those used in [11]. In this work, the authors extracted comments from Twitter that contained certain offensive keywords and performed a

[3] https://transparency.twitter.com/en/reports/rules-enforcement.html#2021-jul-dec

[4] https://transparencyreport.google.com/youtube-policy/removals?hl=en

manual classification in various categories. Then, they tested different automatic classifiers such as random forests, SMO, and multilayer perceptron. The models were trained with features extracted using the LIWC[5] library, designed for the study of emotions in texts. The best result was obtained by a random forest model with an F1 value of 94.47%.

Since the emergence of transformer-based models, like *BERT*, in 2018 [12], many of the proposals to detect abusive language have employed *BERT* trained on formal corpus or texts from social networks. In [13], a new *BERT* model is proposed, called *HateBERT*, trained on comments collected from Reddit communities that were banned for being offensive and promoting hate. The authors retrained the BERT model based on a total of 1.5 million comments and subsequently tested the performance of the model on ensembles of data used in three abusive language detection tasks. The results obtained by HateBERT exceeded those obtained in these evaluation tasks.

Some of the newest approaches have combined several models for improving final results [14], [15] or leveraged knowledge from other tasks using multi-task learning [16] or meta-learning [17], while other approaches have focused on a multilingual setting [18], [19]. Besides, there are continuous efforts for creating new data in new languages [14], [20].

In the next sections, we analyze the main evaluation tasks aimed at detecting offensive language in social networks. This analysis is of special interest because most research on the detection of such language has been carried out in the context of evaluation campaigns. We expose the most outstanding methods, showing the evolution of the approaches used to deal with this problem.

B. SHARED EVALUATION INITIATIVES IN ENGLISH AND OTHER LANGUAGES

1) OffensEval AT SemEval 2019

At the International Semantic Evaluation Workshops (SemVal)[6] of 2019 and 2020, there have been some tasks aimed to detect offensive language in social networks. These tasks were named OffensEval.[7]

In OffensEval 2019 [8], three tasks were proposed using a dataset composed of Twitter comments in English. The three tasks were: (1) classify messages as ''offensive'' or ''non-offensive'', (2) classify the type of offense as ''directed'' (towards a specific person or group) or ''non-targeted'', and (3) identify the target to which the offense is aimed at: to an individual, group, or another target.

In the first task, the winner used a BERT model adjusted to handle the least represented tag classes [21]. The following positions were held by teams that also used BERT models with different settings. The first team to use a model not based on BERT used an ensemble of Convolutional Neural Networks (CNNs) and Bi-LSTM+GRU networks along with Word2vec vectors pre-trained on Twitter [22].

In the second task, ensemble-based models predominated. The winner built a probabilistic model based on the calculation of the level of offense of the comments by applying a dictionary with keywords and hashtags [23]. The second built a neural network based on the ensemble of CNN networks together with BERT [24]. The third used a logistic regression classifier to combine the output of an LSTM network, whose inputs were ELMo contextual vectors, with text characteristics such as unigrams and bigrams [25].

In the third task, the best model was based on a BERT model applying less weight to the most represented classes [26]. The second team combined models such as OpenAI Finetune, LSTM, Transformers, and other non-neural network-based models like SVM and RandomForest. The label for each instance was selected using a majority voting [27].

2) OffensEval AT SemEval 2020

In OffensEval 2020 [28], the tasks were identical to those at OffensEval 2019 but the dataset consisted of Twitter comments in 5 languages: English, Arabic, Danish, Greek, and Turkish.

In the first task, the winner used an ensemble of ALBERT models of different sizes [5]. The second used a ROBERTa-large model [28] and the third used an ensemble of models based on XLM-RoBERTa [29]. In this first task, the top 10 participants used BERT models, sometimes as part of an ensemble with other networks based on CNN or Long Short Term Memory (LSTM).

In the second task, the first place went to the system developed by [29], who proposed a model based on XLM-RoBERTa. The second place combined a BERT model with LSTM layers, whose training used the Noisy Student method to reduce the noise that can be caused by semi-supervised tagging [30]. The team in third place created an architecture that allowed them to address the three subtasks in a hierarchical manner using BERT models [31].

In the third task, the first system was based on an XLM-RoBERTa model [29]. The second used an oversampled BERT model to improve unbalanced classes [32]. The third system combined BERT with some features of texts such as the length of tweets, misspelled words, or use of emojis [33].

A related task, called Toxic Spans Detection, was proposed at SemEval 2021 [34]. This task proposes to detect the exact spans of texts containing toxic language. This task was related to the one tackled in this work, but it differs in the fact of being proposed as a sequence labeling problem instead of text classification.

On the other hand, task 7 at SemEval 2021, called HaHackathon: Detecting and Rating Humor and Offense [35], proposed to rate the degree of offense in comments from 0 to 5. However, this task differs from ours since

[5] http://www.liwc.net/liwcespanol/

[6] https://semeval.github.io

[7] https://sites.google.com/site/offensevalsharedtask/

the authors treat the problem as a regression task instead a classification task.

3) SHARED TASK ON OFFENSIVE LANGUAGE DETECTION at OSACT4

The 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4) proposed a shared task addressed to detecting offensive language and hate speech in Arabic Twitter comments [7]. They proposed two subtasks. The first subtask aimed at classifying tweets into offensive or not offensive. The second subtask was devoted to detecting hate speech.

The winner built a system with two SVM models, a CNN+BiLSTM network, and a multilingual BERT (M-BERT) model. The label of each comment was decided by majority vote [36]. The second used an AraBERT model [37]. The third used a traditional SVM learning model but applied intensive pre-processing: emoticon processing, word categorization, letter normalization, hashtag segmentation, and emoji-to-text conversion.

C. SHARED EVALUATION INITIATIVES IN SPANISH

1) MEX-A3T AT IberLEF

MEX-A3T was a series of tasks held at the Iberian Languages Evaluation Forum (IberLEF) between 2018 and 2020. These tasks consisted of the detection of aggressive language in Twitter comments in Spanish from Mexico.

In the 2018 edition of MEX-A3T [38], participants found two subtasks: author profiling and aggressiveness detection, which is the one most related to our work. The winner of the subtask used a method based on the ensemble of different classifiers along with a lexicon of affective and aggressive words [39]. The second used an LSTM network and the third proposed a method consisting of four stages where different n-grams representations are generated and which use an SVM model for classification [40].

2) MEX-A3T AT IberLEF 2019

In this edition of MEX-A3T, the organizers proposed the same subtasks as the previous edition [41]. The best system used a multilayer-perceptron neural network together with FastText vectors [42]. The second was the baseline of the task, which was based on the winner of the previous year's task. The third used a multilayer-perceptron network with the texts represented using Term Frequency - Inverse Document Frequency (TF-IDF) [43].

3) MEX-A3T at IberLEF 2020

The third edition of MEX-A3T proposed, again, the detection of aggressiveness and included the identification of fake news [44].

The best system detecting aggressiveness created a model based on the ensemble of BETO (BERT models pre-trained in Spanish) and also applies data augmentation by changing words for synonyms and swapping word positions [45]. The second system also used a BETO model and increased the training set by adding samples from the HatEval Spanish dataset [46], which consists of comments with hate speech extracted from Twitter [47]. The third also used BETO and added to the model metadata of the comments and users with the GetOldTweets3 library [48].

4) MeOffendEs AT IberLEF 2021

This task, also held at the IberLEF forums, proposed the identification of offensive language extracted from three social networks (Twitter, Instagram, and YouTube) and its classification into 4 categories[8] [9]. The first system used an XLM-RoBERTa model, which is multilingual, and pre-trained with Twitter texts and sentiment analysis. The second used a combination of a BERT model with language features, such as the use of negations [49]. The third place also used a pre-trained BERT model to which they applied pseudo-labeling to expand the labeling set and focal loss to address the imbalance in the number of samples in each label.

This task offers a fine-grained classification by using four possible labels and the dataset reflects the nature of the problem, where offensive messages are a minor class in the real world. The task also deals with messages from different social networks. Besides, the task is one of the last proposals aimed at detecting offensive language. Therefore, the most recent technologies have been tested in this setting. This is why we have focused on this task.

III. EVALUATION FRAMEWORK

In this Section, we describe the dataset and evaluation metrics used in our experiments, as well as the baselines proposed for comparing our results. All experiments were conducted in Google Colab,[9] a Jupyter notebook environment that runs entirely in the cloud. For deep-learning experiments, we selected the GPU environment. Google Drive was used to store the datasets because of its easy integration with Google Colab.

A. DATASET FOR 4-LABEL CLASSIFICATION

We use the OffendES dataset [50], created by the organizers of the MeOffendES task at IberLEF 2021 (described in Section II-C4) [9]. We have selected this benchmark because it is the latest dataset available for detecting offensive language in Spanish and includes posts from different social media platforms. The dataset is available, under request, at the shared task website.[10]

The organizers manually tagged 30,416 comments collected between February and March 2020 from three different platforms: Twitter, Instagram, and YouTube. Specifically, comments were collected from the accounts of 12 Spanish *influencers* that generate great controversy and have a significant number of followers, whose ages are between

[8] We give more details of the collection in Section III-A
[9] https://colab.research.google.com/
[10] https://competitions.codalab.org/competitions/28679

TABLE 1. Summary of the shared evaluation initiatives on offensive language detection.

Task	Year	Best model	Language
OffensEval	2019	BERT model	English
OffensEval	2020	ALBERT model	English, Arabic, Danish, Greek, Turkish
OSACT-4	2020	2xSVM - CNN+BiLSTM - BERT	Spanish
MEX-A3T (aggresiveness)	2018	Ensemble of classifiers	Spanish
MEX-A3T (aggresiveness)	2019	Multilayer Perceptron	Spanish
MEX-A3T (aggresiveness)	2020	BETO	Spanish
MeOffendEs	2021	XLM-RoBERTa	Spanish

TABLE 2. Label distribution in training, validation, and test subsets of the MeOffendES dataset.

Label	Training	%	Validation	%	Test	%	Total	%
OFP	2051	12.27	10	10	1404	13.32	3465	11.39
OFG	212	1.27	4	4	211	1.55	427	1.40
NOM	1235	7.39	22	22	2340	17.20	3597	1183
NO	13 212	79.07	64	64	9651	70.93	22 927	75.38
Total	16 710	100	100	100	13 606	100	30 416	100

14 and 24 years old. Comments were manually tagged through the Amazon Mechanical Turk platform.[11] The available tags are:

- **NO**: Non-offensive comment nor contains expletive language.
- **NOM**: Comment that is not offensive but contains expletive language.
- **OFG**: Offensive comment towards a group of people belonging to the same ethnic group, gender or sexual orientation, political ideology, religious belief, or other common characteristic
- **OFP**: Offensive comment towards a person.

The kappa coefficient of the dataset is 39.37%, a value that is not too high, mainly due to the discrepancies between annotators in the classification of comments as "OFP" or "OFG".Table 2 shows the number of comments for each label and each subset of the dataset.

As we show in Table 2, the tags are very unbalanced. The comments labeled as offensive accounted for only 12% of the entire dataset. This distribution might affect the quality of the systems, since the two most interesting labels are underrepresented and, therefore, their reliability is not as high as it could be desired. Anyway, this is the distribution expected in a real environment and, therefore, detection systems must deal properly with it.

The training subset represents a 55% of the dataset, while the test subset represents a 44,7% and the validation subset only a 0,03%.

B. DATASET FOR BINARY CLASSIFICATION

Several tasks that focus on detecting offensive language have been proposed to evaluate systems in a binary setting [38], taking into account that for users it is only important to know if a comment is offensive or not. This is why we also evaluate our models in a binary setting. For this purpose, the authors of the OffendES dataset proposed to group some labels into two [50]: OFFENSIVE and NO OFFENSIVE. More in detail, non-offensive and foul language comments are grouped under the label "NO OFFENSIVE", while offensive comments to a group or a person are grouped under the label "OFFENSIVE". The distribution of labels and sets for the binary classification task can be seen in Table 3.

TABLE 3. Label distribution for the binary classification task.

Label	Training (%)	Validation (%)	Test (%)	Total (%)
OFFENSIVE	2 263 (13.54 %)	14 (14 %)	1 615 (11.87 %)	3 892 (12.8 %)
NO OFFENSIVE	14 447 (86.46 %)	86 (86 %)	11 991 (88.13 %)	26 524 (87.2 %)
Total	16 710 (100 %)	100 (100 %)	13 606 (100 %)	30 416 (100 %)

C. EVALUATION METRICS

We evaluate our models using the common metrics found in the literature for evaluating the detection of offensive language:

- **Precision** (see Eq. 1). It is the ratio between elements correctly classified as true instances, or true positives (TP), and all the instances classified as true (i.e. including the elements incorrectly classified as true instances, or false positives (FP)).

$$\frac{TP}{(TP+FP)} \tag{1}$$

- **Recall** (see Eq. 2). It is the ratio between the TP and all the true elements (i.e. including the elements incorrectly classified as false instances, or false negatives (FN)).

$$\frac{TP}{(TP+FN)} \tag{2}$$

- **F1-score** (see Eq. 3). This metric combines precision and recall using the harmonic mean.

$$F1 = 2 * \frac{(precision * recall)}{(precision + recall)} \tag{3}$$

Given the unbalanced nature of the OffendES dataset, we use macro-average precision, recall, and F1. This is because macro metrics offer a better interpretation of performance in the underrepresented classes. It is important to take into account this unbalanced nature of data because it represents the real nature of these comments on the Internet, where offensive comments are less frequent, but it is important to detect them. Besides, we also include results according to micro-average precision, recall, and F1, which were the main metrics used at the MeOffendES shared task [9].

For binary classification, we use weighted average instead of micro-average metrics. This is because the MeOffendES

[11] https://www.mturk.com/worker

TABLE 4. Micro results of the proposed baselines at MeOffendES for 4-label classification.

Model	Precision	Recall	F1
Best at MeOffendES	0.8815	0.8815	0.8815
baseline-svm	0.8285	0.8285	0.8285

TABLE 5. Macro results of the proposed baselines at MeOffendES for 4-label classification.

Model	Precision	Recall	F1
Best at MeOffendES	0.7679	0.7093	0.7324
baseline-svm	0.6278	0.4831	0.5236

shared task did not evaluate binary classification. Hence, we cannot compare our results in binary classification with those from participants at the MeOffendES shared task. However, the authors of the OffendES dataset included in their paper weighted-average results of a baseline system doing binary classification [50]. Thus, we can compare our results with those from that baseline system.

D. BASELINES AND CURRENT STATE-OF-THE-ART SYSTEMS

We have considered some participant systems at the MeOffendES shared task as our baselines. Firstly, we have selected the baseline proposed by the organizers of the task. This baseline is based on a linear SVM classifier which takes as input features Bag-of-Words of unigrams, bigrams, and trigrams. We name this baseline as *baseline-svm*.

The second baseline is the best participant system at the MeOffendES shared task. This system was developed by the NLP-CIC team [51] and it is based on a multilingual XLM-RoBERTa model pre-trained on Twitter and sentiment analysis data. We name this baseline as *Best at MeOffendES*. This system represents the best performing system, and the current state-of-the-art for this dataset, outperforming results published in the Codalab site of the shared task.[12] The results of these two baselines are available at the overview of the task, and we show them in Tables 4 and 5.

We see in the tables how both systems obtain similar scores with micro measures, while the difference is quite bigger for macro results. This is due to the low performance of the baseline in the less representative labels. Therefore, these results show the importance of taking into account macro scores as we have pointed out above.

These baselines are only used for 4-label classification given that the MeOffendES shared task did not evaluate binary classification. So, for binary classification, we take results of the baseline used in the paper that introduces the collection [50], where the authors tested a BETO model in its uncased version.[13]

TABLE 6. Conversion from numbers to letter.

Number	Letter
0	O
1	I
3	E
4	A
5	S
7	T
8	B

IV. DATA PRE-PROCESSING

Comments on social networks are informal and contain many elements that introduce noise and reduce the effectiveness of tools that work frequently with formal text-based corpus such as news, books, etc. This is why we have applied some methods oriented to clean up comments and make them as formal as possible. The methods implemented are:

- **Remove repeated phrases**: It is common to find comments in which the same word or phrase is repeated consecutively. These repeated elements may add some noise to the tweet and give more importance to the same words that are not so important. We reduce the text and leave only the first appearance of the word or phrase. For example, the text *"Correr es vivir, Correr es vivir, Correr es vivir, Correr es vivir"* would be replaced by "*Correr es vivir,*"
- **Remove character repetitions**: Some comments may contain repetitions of the same character in a word as a way to emphasize it. This causes the words to be left out of common vocabulary and, as a consequence, these words cannot be assigned to a vector when using pre-trained embeddings. The implemented method eliminates repetitions of characters taking into account that, in Spanish, there are valid repetitions like the consonants "r", "l", "c" or "n". For example, the text *"El -13 tiene un currrroooo hace aaaños"* would be replaced by "*El -13 tiene un curro hace años*".
- **Treatment of *leet* speak:** *Leet* consists of replacing certain letters with numbers whose shape bears some resemblance to the letter they replace. One of the current uses of Leet is to make reading difficult for users unrelated to this type of writing[14] but they also make it difficult for computer systems to interpret the messages. A method has been implemented that treats words where there are mixtures of numbers and letters. In these cases, we convert numbers to letters following the conversion given in Table 6. For example, the text *"el que da m3 gu5t4 (m1ra mi nombr3)"* would be replaced by "*el que da me gusta (mira mi nombre)*".
- **Number cleanup**: All numbers have been replaced by a single number, thus reducing the number of different tokens and consequently the dimensionality of the vectorized texts.
- **Emoji and emoticons cleanup**: Emojis and emoticons have been removed from comments. This is a common

[12] https://competitions.codalab.org/competitions/28679#results

[13] We have not included this baseline system for 4-label classification given that it obtains worst results than the best-performing system at the MeOffendES shared task, and the authors did not include micro-average results. Moreover, we propose the use of a slightly different BETO model for our experiments, that improves those results

[14] https://en.wikipedia.org/wiki/Leet

processing when working with tweets, although other researchers include a text associated with these symbols [52].

- **URL Cleanup**: Detected URLs have been replaced with the word "address".
- **Hashtag and tag cleanup**: Hashtags are replaced by the word "label".
- **User cleanup:** Usernames are replaced by the word "user". We considered a word as a noun when it begins with "@" and is followed by a capital letter.
- **Standardization of laughs**: The ways to represent laughs in a text are tremendously varied. We have tried to detect as many representations as possible. All of them have been replaced by the word "laughs".
- **Space adjustments**: We applied a specific treatment for blanks:
  - Sequences of 2 or more spaces are replaced by a single space.
  - A space is inserted after the symbols "?" and "!" to improve the effectiveness of the tokenizer.

V. MODELS

In this section, we describe the models tested in this paper. We test the most common methods in the literature, beginning from classic Machine Learning models using Bag-of-Words, to more complex deep-learning architectures and transformers.

A. BAG-OF-WORDS BASED MODELS

The first type of models is based on extracting features from texts using BoW and classifying the texts using different Machine Learning methods. For this type of models, we pre-process texts applying stemming and removing stop-words. Additionally, we only keep terms with at least three occurrences in the collection and represent them by their TF-IDF scores in the collection (see Eq. 4). We have selected this representation after testing several variants, such as the use of lemmas, different categories of words, etc. The definition of the TF-IDF score of a term t in a document d is as follows:

$$TF-IDF_{t,d} = (1 + \log tf_{t,d}) \cdot \log \frac{N}{df_t}, \tag{4}$$

where $tf_{t,d}$ is the frequency of term t in document d, df_t is the number of documents containing term t and N is the number of documents in the collection.

We select hyperparameters[15] through an exhaustive search using cross-validation on the training set. The metric employed to measure the performance was the balanced score (see Eq. 5), which is defined as the average completeness obtained in each class.

$$\text{balanced-accuracy} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \tag{5}$$

This metric was chosen because it can adequately work with classification problems with unbalanced labeling as occurs in the training dataset.

In this group of experiments, we test the most common Machine Learning methods in NLP tasks:

- **Stochastic Gradient Descent (SGD) Classifier** (see Eq. 6). It implements regularized linear models with SGD learning. This classifier has the advantage of being able to efficiently handle large and high-dimensional datasets. This model has a huge amount of hyperparameters but thanks to its very fast training it is possible to perform a huge number of combinations of them in a reasonable time. To find the model parameters, the regularized training error given by the following expression must be minimized:

$$E(w, b) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) + \alpha R(w), \tag{6}$$

  where $f(x)$ is the linear scoring function ($f(x) = w^T x + b$), L is the loss function, α is a non-negative parameter that controls the regularization strength, and R is a regularization term that penalizes model complexity.
- **Support Vector Machine (SVM)**. Represents each sample in a n-dimensional space according to the values given for its features. The goal of the SVM classifier is to find a hyper-plane that separates the two classes trying to maximize the space between each class by maximizing the margin. The output is the predicted class, instead of a probabilistic score. For spaces that are not linearly separable, kernels are used, which are functions that increase the dimensionality of the problem, so that a non-linearly separable problem in a specific dimensional space can be separable in a higher dimensional space.
- **RandomForest**. RadomForest is part of the models that work employing estimator ensembles. These models consist of combining a fixed number of decision trees that are trained using a technique called bagging. This way of training consists in that each tree is trained with a different set of samples, the samples are taken randomly from the training data set. With this technique, by combining the results of all the trees, the errors of some trees are compensated with those of others, which improves the generalization capacity of the method. To make the global prediction, the predictions of all the trees are combined and the option with the most votes is chosen, weighting the vote according to the probability given by each tree.
- **GradientBoosting**. It is another ensemble model like random forests but applying a training technique known as boosting (instead of bagging). While in random forests each decision tree is trained independently by bagging, in boosting-based models, such as this one and Adaboost, each decision tree is built on top of the previous tree, i.e. it is an additive process. In the

[15] Hyperparameters are listed in Appendix C

GradientBoosting model, each new tree tries to correct the residual error of the previous one.

- **AdaBoost**. This is the last ensemble method used, its training is also done by boosting but with a different philosophy than GradientBoosting: estimators are added that pay more attention to instances that were misclassified by the previous estimator. To increase attention to the misclassified examples, the algorithm increases the weights of the misclassified instances and trains a new classifier using the new weights. Contrary to the previous methods, other base estimators than decision trees can be used.

Furthermore, we test some oversampling techniques for some of the best methods. The objective was to improve results for minority classes (OFG and OFP). We obtained the best results by adding 7000 samples only for minority classes.

B. DEEP LEARNING MODELS

In this section, we describe models based on deep-learning architectures using CNNs and Bi-LSTMs, which have been successfully tested in other related NLP tasks such as sentiment analysis [53], [54], stance detection [55], [56], etc. The input features in these experiments are extracted using FastText word-embeddings [57], after applying automatic word correction to each tweet. We have chosen FastText given that it uses sub-words instead of entire words, which might be more suitable for representing texts from social networks. The final models are selected after following an incremental process testing different alternatives [58].

CNNs are a type of network that exploits spatial information and therefore perform very well on problems with images as input data but can also be applied to natural language processing. Such a network is built by stacking layers: the lower-level layers can detect low-level features while the upper layers detect high-level features. In each layer, a convolution and filtering operation is performed: convolution consists of associating an input submatrix with a single neuron in the layer, while filtering is an operation that allows highlighting some feature of the data. The output of a neuron in a 1D convolutional layer is shown on Equation 7.

$$z_{i,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{k'=0}^{f_{n'}-1} x_{i',k'} \times w_{u,k,k'} \quad \text{with } i' = i \times s_h + u, \tag{7}$$

where z_i, k is the output of the neuron in the ith position and feature map k; s_h is the stride of the kernel; f_h is the size of the receptive field; f'_n is the number of feature maps in the previous layer; $x_{i',k'}$ is the output of the neuron in the previous layer; b_k is the bias term for the feature map; and $w_{u,k',k}$ is the weight between any neuron in feature map k and the input at the ith position, and feature map k'.

For CNN-based models, we test two variants: one with a single convolutional layer, and a second one with three convolutional layers. The architectures and hyperparameters for these models can be seen in Appendix A.

LSTM networks are a type of Recurrent Neural Networks (RNNs). RNNs are a type of network that uses sequential data, which makes them a very interesting tool in the processing of natural language. The information from previous data is passed to each time step t using a hidden state h_t. Then, the output is computed from the hidden state. LSTMs include memory cells able to store more accurate information from previous inputs. When using a bidirectional architecture, for example, a Bi-LSTMs, the hidden state is computed using information from both the left and right context. The basic equations for these networks are Equations 8 and 9.

$$h_t = g(W_x X_t + W_{hl} h_{t-1} + W_{hr} h_{t+1} + b_h), \tag{8}$$

$$y_t = f(W_y h_t + b_t), \tag{9}$$

where g and f are activation functions, Ws and bs are, respectively, weights and biases to be learned.

We test a model with a single Bi-LSTM and a stacked model with two Bi-LSTMs (the input to the second Bi-LSTM is the output from the first LSTM). Concrete architectures for these models can be seen in Appendix B.

We do not include any oversampling technique given that in the development period, we did not see any special contribution of such processing.

C. TRANSFORMER-BASED MODELS

In this Section, we propose the use of two transformer-based models, which are obtaining the best results in several NLP tasks [59]. The transformers architecture uses the self-attention mechanism, which allows to include information from any input token in the following layers of the network [60]. Transformer-based models are pre-trained on vast amounts of text and then, fine-tuned to specific tasks using less data [61]. These models have proved their robustness when they are fine-tuned on unbalanced data, given that they can work properly even for the minor classes. Thus, this type of models seems suitable for detecting offensive language, where real data, such as the one used in MeOffendEs, is unbalanced.

BERT models are a family of transformer-based models suitable for classifying input texts [12]. These models take input text and map them to input features learned during pre-training. We have tested BETO, a BERT model pre-trained on Spanish documents [62]. We also propose to test RoBERTuito [52], which is a RoBERTa-base model trained on 500 million Spanish tweets. Thus, we can compare the performance of a system pre-trained on formal texts against a system pre-trained on texts from social networks. We hypothesize that using a model pre-trained on non-formal data can favor the model when fine-tuning it on social data.

RoBERTuito differs from the system that obtained the best performance at MeOffendEs, an XLM-RoBERTa model, in the data used for pre-training the model. Both models were pre-trained using Twitter data, but RoBERTuito was pre-trained on Spanish data while XLM-RoBERTa was pre-trained on multilingual data. So, we can compare in this

TABLE 7. Macro-average results for 4-labels classification.

Model	Precision	Recall	F1
RoBERTuito	0.7968	0.8137	0.8011
BETO	0.7684	0.7425	0.7544
Best at MeOffendES	0.7679	0.7093	0.7324
CNN	0.6632	0.7063	0.6813
Bi-LSTM 2 layers	0.6295	0.7672	0.6767
Bi-LSTM	0.6408	0.7164	0.6717
CNN 3 layers	0.6221	0.6829	0.6477
SGD oversampling	0.6121	0.6366	0.6173
SVM	0.5948	0.6451	0.5947
SGD	0.6041	0.6167	0.5918
SVM oversampling	0.5804	0.6365	0.5893
GradientBoosting	0.6166	0.5201	0.5509
AdaBoost	0.5233	0.6024	0.5427
RandomForest	0.5951	0.5012	0.5301
baseline-svm	0.6278	0.4831	0.5236

TABLE 8. Micro-average results for 4-labels classification.

Model	Precision	Recall	F1
RoBERTuito	0.9011	0.9011	0.9011
BETO	0.8855	0.8855	0.8855
Best at MeOffendES	0.8815	0.8815	0.8815
CNN	0.8403	0.8403	0.8403
Bi-LSTM	0.8374	0.8374	0.8374
SGD oversampling	0.8385	0.8325	0.8346
baseline MeOffendES	0.8285	0.8285	0.8285
Bi-LSTM 2 layers	0.8227	0.8227	0.8227
CNN 3 layers	0.8235	0.8235	0.8235
SGD	0.8356	0.8216	0.8233
SVM	0.8399	0.8060	0.8203
SVM oversampling	0.8330	0.8028	0.8158
GradientBoosting	0.8135	0.8325	0.8147
RandomForest	0.8106	0.8304	0.8058
AdaBoost	0.7873	0.7149	0.7401

paper the effect of pre-training on social-media data using a monolingual model, RoBERTuito, instead of a multilingual model, XLM-RoBERTa.

We fine-tune the two transformer-based models on the training dataset for only one epoch to avoid over-fitting. Similar to the previous deep-learning methods, we did not obtain any improvement by applying oversampling in the development period. So, we have not included it in these experiments.

VI. RESULTS

In this Section, we show and analyze the results of the models described in Section V. We first report the results using 4 labels and then we report the results of the best models in the binary setting.

A. 4-LABEL CLASSIFICATION RESULTS

We show macro-average results for 4-labels classification in Table 7. We also include the results of the best system at MeOffendES, which is the current state-of-the-art (named in the Table as *Best at MeOffendES*), and the baseline proposed at the task (named in the Table as *baseline-svm*), which have been described in Section III-D.

We see how all our proposals outperform the baseline given at MeOffendES (system *baseline-svm*). Besides, the transformer-based models described in Section V-C outperform the best system at the MeOffendES shared task. Thus, we have established a new state-of-the-art result for this dataset.

According to the results, the RoBERTuito model performs better than the BETO model. This means that the fact of pre-training the system with Twitter texts, which are more similar to those in the dataset, contributes to such improvement. While the previous best model, based on an XLM-RoBERTa model, was also pre-trained on Twitter data, the fact of using a monolingual model for this task seems to be more suitable. Our BETO model, which is monolingual, also outperforms the XLM-RoBERTa model.

Regarding the other models, we see how all the deep-learning methods outperform the models based on BoW. When including oversampling, only the SGD classifier improves results. Thus, it is unclear if oversampling can help with these methods.

In Table 8, we show the micro-average results of our models and the proposed baselines. Again, the transformer-based models outperform the results of the best previous system, setting a new state-of-the-art result using the primary MeOffendES measures.[16] RoBERTuito achieves an F1 score of 0,9011, which means that the system can classify input texts with few errors. However, some of the other proposals obtained lower scores than the baseline proposed at MeOffendES.

Differences in micro and macro-average scores concerning the baseline suggest that our models focus more on obtaining good scores across the different classes than on the whole collection. That is, given that offensive messages were represented by the two minor classes, a system able to correctly classify non-offensive messages but failing with offensive messages, would obtain good micro-average results but lower macro-average results. This is why we have given more importance to macro-average scores. We have observed a low performance in the OFG class, which is the class with fewer samples. Some of our models (those based on BoW) did not return any value for this class given its low appearance in the training set. Our best system was the one able to obtain similar results across the different classes. We show the detailed results of the RoBERTuito model in Table 9.

In Table 9, we can see good results for each class, with the lowest F1 score obtained in the OFG class with a score of 0.6721. Thus, the RoBERTuito model can detect each class no matter their presence in the training collection. Nevertheless, there is still room for improvement in the minority classes.

B. BINARY CLASSIFICATION RESULTS

We show macro-average results in Table 10, while weighted-average results are shown in Table 11. We only include results of the best models of each group from Sections V-A, V-B, and V-C. We also include the baseline

[16] Remember that micro-average were the primary measures at MeOffendES, while macro-average results were complementary

TABLE 9. Detailed results for RoBERTuito model.

	Precision	Recall	F1
NO	0.9320	0.9598	0.9457
NOM	0.8135	0.7700	0.7911
OFG	0.5920	0.7772	0.6721
OFP	0.8497	0.7479	0.7956
macro	0.7968	0.8137	0.8011
micro	0.9011	0.9011	0.9011

TABLE 10. Macro-average results for binary classification.

Model	Precision	Recall	F1
RoBERTuito	0.9008	0.8570	0.8767
CNN	0.7998	0.8467	0.8195
SGD	0.7739	0.8132	0.7906
OffendES baseline	0.7042	0.7674	0.7839

TABLE 11. Weighted-average results for binary classification.

Model	Precision	Recall	F1
RoBERTuito	0.9269	0.9290	0.9270
OffendES baseline	0.8906	0.8959	0.8926
CNN	0.8935	0.8814	0.8857
SGD	0.8752	0.8632	0.8678

TABLE 12. Macro-average results for 4-labels classification with different levels of pre-processing.

Model	Precision	Recall	F1
With pre-processing	0.7968	0.8137	0.8011
Default pre-processing	0.7912	0.8150	0.7984
Without pre-processing	0.7860	0.8149	0.7946

TABLE 13. Macro-average results for binary classification with different levels of pre-processing.

Model	Precision	Recall	F1
Default pre-processing	0.8902	0.8720	0.8807
Without pre-processing	0.8912	0.8700	0.8800
With pre-processing	0.9008	0.8570	0.8767

TABLE 14. Weighted-average results for binary classification with different levels of pre-processing.

Model	Precision	Recall	F1
Default pre-processing	0.9278 0.9291 0.9282		
Without pre-processing	0.9276 0.9290 0.9280		
With pre-processing	0.9269	0.9290	0.9270

from the paper introducing the collection and described in Section III-D (we call it as *OffendES baseline*). However, we were unable to include results from participants at the MeOffendES shared task because this setting was not proposed in the task.

In both Tables, we can see how the RoBERTuito model outperforms the other models, with a bigger difference for macro-average results (where each class receives the same weight). Thus, the RoBERTuito model can perform a good classification of most of the tweets, no matter their class. The other models defeat the baseline for macro-average, but not for weighted-average results.

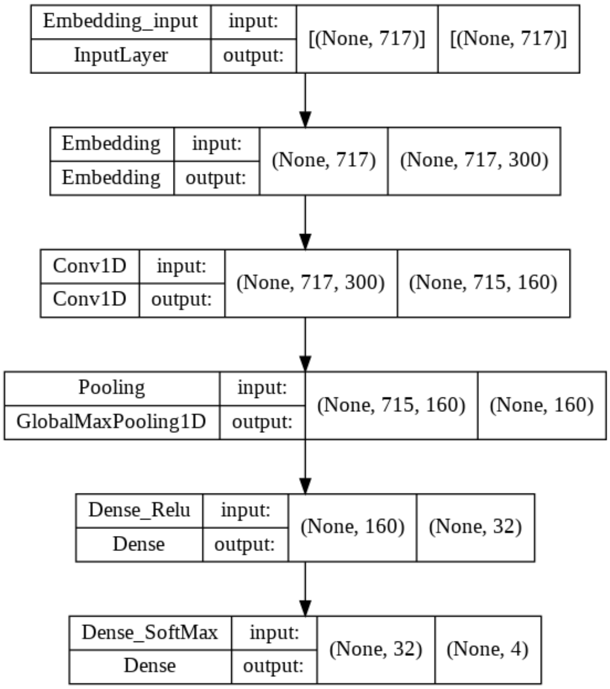

FIGURE 1. CNN of 1 layer.

C. IMPACT OF PRE-PROCESSING

In this Section, we analyze the impact of pre-processing in the final results. We focus on the results of the best system, the RoBERTuito model, for simplicity. We show the results of this study in Tables 12, for 4-label classification and, Tables 13 and 14 for binary classification. Each Table contains results of the RoBERTuito model after applying each one of the following three types of pre-processing to the input text:

- Without pre-processing: we do not apply any kind of pre-processing. That is, the model receives the raw text as it is in the dataset.
- Default pre-processing: we apply the default pre-processing of the model. This pre-processing, fully described in [52], mainly consists in limiting character repetitions, converting user handles to a common token, and replacing hashtags and emojis by a special token and the hashtag or the emoji's textual representation.
- With pre-processing: we apply the pre-processing described in Section IV. In the development period, we obtain the best results using this pre-processing.

In these Tables, we can see that the differences in performance are quite small, even below 0,01, and therefore, not significant. Hence, the pre-processing approaches studied do not affect results and could be omitted for detecting offensive language with the RoBERTuito model.

VII. CONCLUSION AND FUTURE WORK

The detection of Offensive language remains a critical problem in the current society, with a special focus on social networks. In this paper, we have tested several systems, from

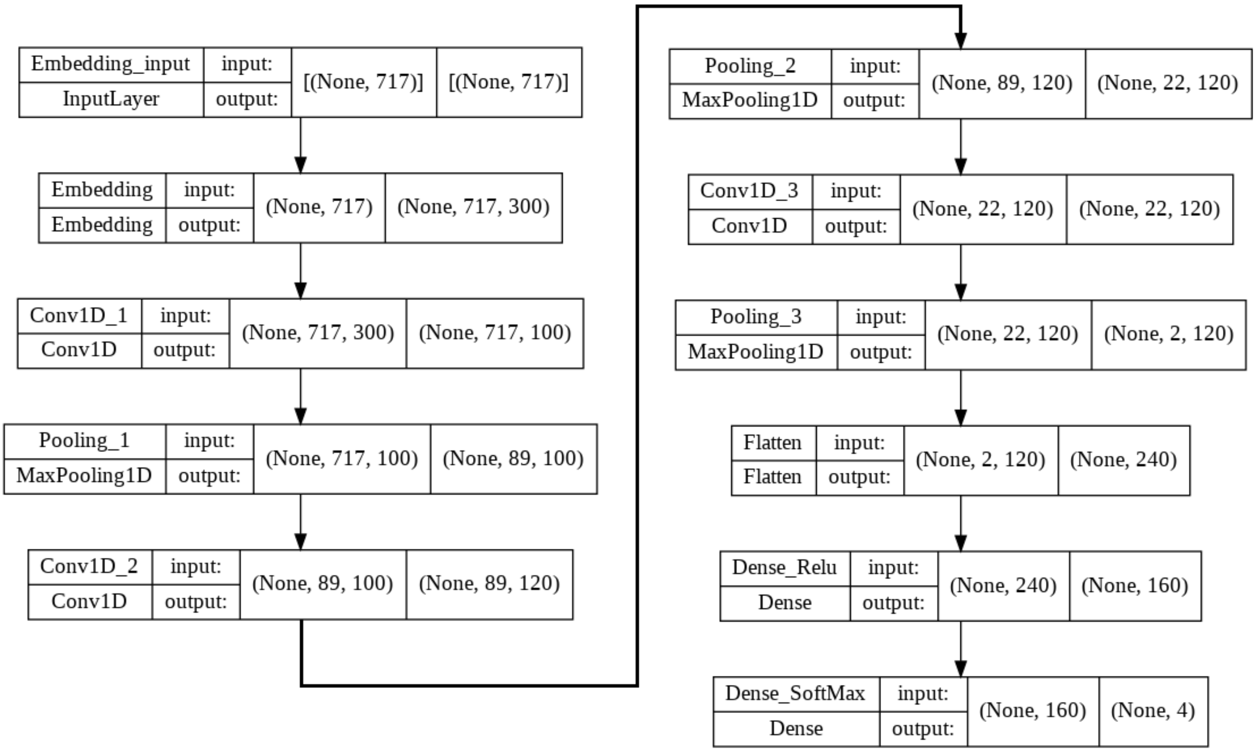
Embedding_input | InputLayer | input: | output: | [(None, 717)] | [(None, 717)]
Embedding | Embedding | input: | output: | (None, 717) | (None, 717, 300)
Conv1D_1 | Conv1D | input: | output: | (None, 717, 300) | (None, 717, 100)
Pooling_1 | MaxPooling1D | input: | output: | (None, 717, 100) | (None, 89, 100)
Conv1D_2 | Conv1D | input: | output: | (None, 89, 100) | (None, 89, 120)
Pooling_2 | MaxPooling1D | input: | output: | (None, 89, 120) | (None, 22, 120)
Conv1D_3 | Conv1D | input: | output: | (None, 22, 120) | (None, 22, 120)
Pooling_3 | MaxPooling1D | input: | output: | (None, 22, 120) | (None, 2, 120)
Flatten | Flatten | input: | output: | (None, 2, 120) | (None, 240)
Dense_Relu | Dense | input: | output: | (None, 240) | (None, 160)
Dense_SoftMax | Dense | input: | output: | (None, 160) | (None, 4)

FIGURE 2. CNN of 3 layers.

classical Machine Learning algorithms using Bag-of-Words (BoW) representations to transformer-based models using the collection of the MeOffendES shared task.

The main problem detected in this task is the low proportion of offensive comments in real data, which limits the learning capacity of the models. While transformer-based models still perform well in minor classes, the other models suffer from them.

We have established new state-of-the-art results using a transformer-based model pre-trained on Spanish social data such as tweets, given that they are more similar to the texts used in the task. We obtain the best results using RoBERTuito adding several text pre-processing steps. When facing the task as a binary classification problem, the results measured with an F1 score rise to 0.9, showing the feasibility of using such models in a real environment. Thus, social networks could use our approach for detecting and avoiding harmful messages in an early stage, increasing the confidence of users.

There is still room for improvement in the minor classes, so future work is oriented to improve results in such classes and, therefore, in overall results. We would also like to test the impact of including additional pre-processing techniques such as spelling correction.

APPENDIX A
CNN ARCHITECTURES

We show in Figures 1 and 2 the architectures of the CNN-based models described in Section V-B. We use the RMSprop optimizer, a learning rate of 0.003 and train the models for 15 epochs using early stopping with a patience of 10 epochs. We did not set any other hyperparameters and use their default values.

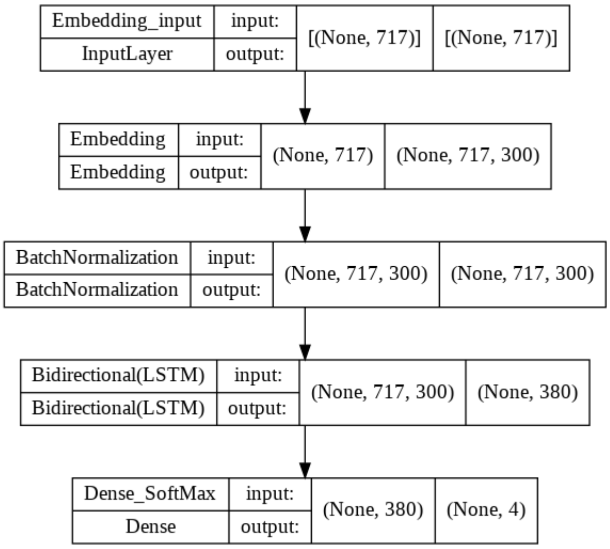
Embedding_input | InputLayer | input: | output: | [(None, 717)] | [(None, 717)]
Embedding | Embedding | input: | output: | (None, 717) | (None, 717, 300)
BatchNormalization | BatchNormalization | input: | output: | (None, 717, 300) | (None, 717, 300)
Bidirectional(LSTM) | Bidirectional(LSTM) | input: | output: | (None, 717, 300) | (None, 380)
Dense_SoftMax | Dense | input: | output: | (None, 380) | (None, 4)

FIGURE 3. Architecture of the model using a single Bi-LSTM.

APPENDIX B
BI-LSTMs ARCHITECTURES

We show in Figures 3 and 4 the architectures of the LSTM-based models described in Section V-B. We use

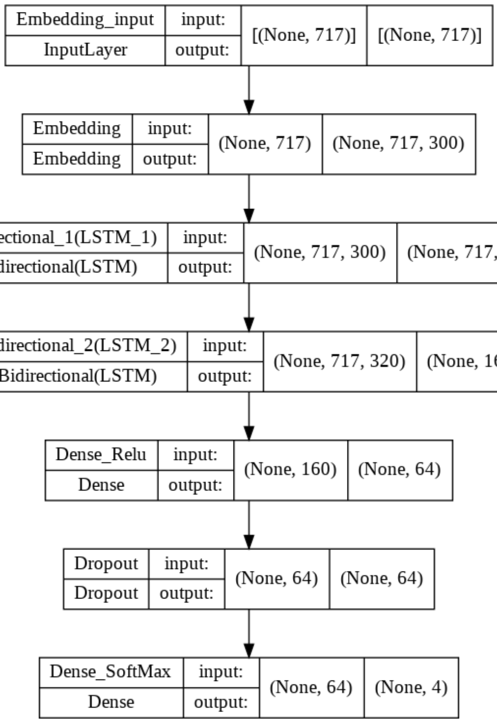
Embedding_input | input: | [(None, 717)] | [(None, 717)]
InputLayer | output:
Embedding | input: | (None, 717) | (None, 717, 300)
Embedding | output:
Bidirectional_1(LSTM_1) | input: | (None, 717, 300) | (None, 717, 320)
Bidirectional(LSTM) | output:
Bidirectional_2(LSTM_2) | input: | (None, 717, 320) | (None, 160)
Bidirectional(LSTM) | output:
Dense_Relu | input: | (None, 160) | (None, 64)
Dense | output:
Dropout | input: | (None, 64) | (None, 64)
Dropout | output:
Dense_SoftMax | input: | (None, 64) | (None, 4)
Dense | output:

FIGURE 4. Architecture of the model using a stacked Bi-LSTM.

the RMSprop optimizer, a learning rate of 0.002 and train the models for 50 epochs using early stopping with a patience of 10 epochs. We did not set any other hyperparameters and use their default values.

APPENDIX C
MODELS CONFIGURATIONS

In this Appendix, we include the exact parameters used in the Bag-of-Words models (parameters of the other models are given in their descriptions).

A. 4-LABEL CLASSIFICATION

- SGDClassifier (both with and without oversampling):
  - alpha: 0.0001
  - epsilon: 0.001
  - loss: hinge
  - penalty: l1
  - tol: 0.01
- SVM (both with and without oversampling):
  - C: 0,3
  - gamma: scale
  - kernel: sigmoid
- RandomForest:
  - criterion: gini
  - max_features: sqrt
  - n_estimators: 2000
- GradientBoosting:
  - criterion: squared_error
  - learning_rate: 0,1
  - max_depth: 16
  - max_features: sqrt
  - n_estimators: 500
- AdaBoost:
  - base_estimator: RandomForestClassifier con max_depth=3
  - learning_rate: 0,01
  - n_estimators: 200

B. BINARY CLASSIFICATION

- SGDClassifier:
  - alpha: 0,0001
  - epsilon: 0,001
  - loss: log
  - penalty: l2
  - tol: 0,01

REFERENCES

[1] M. Wiegand, M. Siegel, and J. Ruppenhofer, "Overview of the GermEval 2018 shared task on the identification of offensive language," in *Proc. 14th Conf. Natural Lang. Process. (KONVENS)*. Vienna, Austria: Austrian Academy of Sciences, Sep. 2018, pp. 1–10.

[2] D. Hickey, M. Schmitz, D. Fessler, P. E. Smaldino, G. Muric, and K. Burghardt, "Auditing Elon Musk's impact on hate speech and bots," in *Proc. Int. AAAI Conf. Web Social Media*, Jun. 2023, vol. 17, no. 1, pp. 1133–1137.

[3] M. Wypych and M. Bilewicz, "Psychological toll of hate speech: The role of acculturation stress in the effects of exposure to ethnic slurs on mental health among Ukrainian immigrants in Poland," *Cultural Diversity Ethnic Minority Psychol.*, Jan. 2022. [Online]. Available: https://psycnet.apa.org/record/2022-23266-001

[4] O. Ştefăniţă and D.-M. Buf, "Hate speech in social media and its effects on the LGBT community: A review of the current research," *Romanian J. Commun. Public Relations*, vol. 23, no. 1, pp. 47–55, 2021.

[5] G. Wiedemann, S. M. Yimam, and C. Biemann, "UHH-LT at SemEval-2020 task 12: Fine-tuning of pre-trained transformer networks for offensive language detection," in *Proc. 14th Workshop Semantic Eval.* Barcelona, Spain: International Committee for Computational Linguistics, 2020, pp. 1638–1644.

[6] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *ACM Comput. Surveys*, vol. 51, no. 4, pp. 1–30, Jul. 2018.

[7] H. Mubarak, K. Darwish, W. Magdy, T. Elsayed, and H. Al-Khalifa, "Overview of OSACT4 Arabic offensive language detection shared task," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools, Shared Task Offensive Lang. Detection*. Marseille, France: European Language Resource Association, May 2020, pp. 48–52.

[8] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval)," in *Proc. 13th Int. Workshop Semantic Eval.*, 2019, pp. 75–86.

[9] F. M. P.-D. Arco, M. Casavantes, H. J. Escalante, M. T. Martín-Valdivia, A. Montejo-Ráez, M. Montes, H. Jarquín-Vásquez, and L. Villaseñor-Pineda, "Overview of MeOffendEs at IberLEF 2021: Offensive language detection in Spanish variants," *Procesamiento del Lenguaje Natural*, vol. 67, pp. 183–194, Sep. 2021.

[10] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *Proc. Int. Conf. Privacy, Secur., Risk Trust Int. Conf. Social Comput.*, Sep. 2012, pp. 71–80.

[11] P. Nand, R. Perera, and A. Kasture, "'How bullying is this message?' A psychometric thermometer for bullying," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, Osaka, Japan, Dec. 2016, pp. 695–706.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.* Minneapolis, MN, USA: Association for Computational Linguistics, vol. 1, Jun. 2019, pp. 4171–4186.

[13] T. Caselli, V. Basile, J. Mitrović, and M. Granitzer, "HateBERT: Retraining BERT for abusive language detection in English," in *Proc. 5th Workshop Online Abuse Harms (WOAH)* Cedarville, OH, USA: Association for Computational Linguistics, 2021, pp. 17–25.

[14] P. K. Roy, S. Bhawal, and C. N. Subalalitha, "Hate speech and offensive language detection in dravidian languages using deep ensemble framework," *Comput. Speech Lang.*, vol. 75, Sep. 2022, Art. no. 101386.

[15] M. Anand, K. B. Sahay, M. A. Ahmed, D. Sultan, R. R. Chandan, and B. Singh, "Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques," *Theor. Comput. Sci.*, vol. 943, pp. 203–218, Jan. 2023.

[16] F. M. Plaza-del-Arco, M. D. Molina-González, L. A. Ureña-López, and M.-T. Martín-Valdivia, "Integrating implicit and explicit linguistic phenomena via multi-task learning for offensive language detection," *Knowl.-Based Syst.*, vol. 258, Dec. 2022, Art. no. 109965.

[17] M. Mozafari, R. Farahbakhsh, and N. Crespi, "Cross-lingual few-shot hate speech and offensive language detection using meta learning," *IEEE Access*, vol. 10, pp. 14880–14896, 2022.

[18] F.-Z. El-Alami, S. O. E. Alaoui, and N. E. Nahnahi, "A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 8, pp. 6048–6056, Sep. 2022.

[19] K. Shanmugavadivel, V. E. Sathishkumar, S. Raja, T. B. Lingaiah, S. Neelakandan, and M. Subramanian, "Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data," *Sci. Rep.*, vol. 12, no. 1, p. 21557, Dec. 2022.

[20] M. Subramanian, R. Ponnusamy, S. Benhur, K. Shanmugavadivel, A. Ganesan, D. Ravi, G. K. Shanmugasundaram, R. Priyadharshini, and B. R. Chakravarthi, "Offensive language detection in Tamil Youtube comments by adapters and cross-domain knowledge transfer," *Comput. Speech Lang.*, vol. 76, Nov. 2022, Art. no. 101404.

[21] P. Liu, W. Li, and L. Zou, "NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers," in *Proc. 13th Int. Workshop Semantic Eval.*, 2019, pp. 87–91.

[22] D. Mahata, H. Zhang, K. Uppal, Y. Kumar, R. R. Shah, S. Shahid, L. Mehnaz, and S. Anand, "MIDAS at SemEval-2019 task 6: Identifying offensive posts and targeted offense from Twitter," in *Proc. 13th Int. Workshop Semantic Eval.*, 2019, pp. 683–690.

[23] J. Han, S. Wu, and X. Liu, "Jhan014 at SemEval-2019 task 6: Identifying and categorizing offensive language in social media," in *Proc. 13th Int. Workshop Semantic Eval.* Minneapolis, MN, USA: Association for Computational Linguistics, 2019, pp. 652–656.

[24] A. Rozental and D. Biton, "Amobee at SemEval-2019 tasks 5 and 6: Multiple choice CNN over contextual embedding," in *Proc. 13th Int. Workshop Semantic Eval.* Minneapolis, MN, USA: Association for Computational Linguistics, 2019, pp. 377–381.

[25] A. Oberstrass, J. Romberg, A. Stoll, and S. Conrad, "HHU at SemEval-2019 task 6: Context does matter—Tackling offensive language identification and categorization with ELMo," in *Proc. 13th Int. Workshop Semantic Eval.*, 2019, pp. 628–634.

[26] A. Nikolov and V. Radivchev, "Nikolov-radivchev at SemEval-2019 task 6: Offensive tweet classification with BERT and ensembles," in *Proc. 13th Int. Workshop Semantic Eval.*, 2019, pp. 691–695.

[27] A. Seganti, H. Sobol, I. Orlova, H. Kim, J. Staniszewski, T. Krumholc, and K. Koziel, "NLPR@SRPOL at SemEval-2019 task 6 and task 5: Linguistically enhanced deep learning offensive sentence classifier," in *Proc. 13th Int. Workshop Semantic Eval.* Minneapolis, MN, USA: Association for Computational Linguistics, 2019, pp. 712–721.

[28] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, and Ç. Çöltekin, "SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020)," in *Proc. 14th Workshop Semantic Eval.*, 2020, pp. 1425–1447.

[29] S. Wang, J. Liu, X. Ouyang, and Y. Sun, "Galileo at SemEval-2020 task 12: Multi-lingual learning for offensive language identification using pre-trained language models," in *Proc. 14th Workshop Semantic Eval.* Barcelona, Spain: International Committee for Computational Linguistics, 2020, pp. 1448–1455.

[30] B.-T. Pham-Hong and S. Chokshi, "PGSG at SemEval-2020 task 12: BERT-LSTM with tweets' pretrained model and Noisy Student training method," in *Proc. 14th Workshop Semantic Eval.*, 2020, pp. 2111–2116.

[31] P.-C. Chen, H.-H. Huang, and H.-H. Chen, "NTU_NLP at SemEval-2020 task 12: Identifying offensive tweets using hierarchical multi-task learning approach," in *Proc. 14th Workshop Semantic Eval.*, 2020, pp. 2105–2110.

[32] M. Pàmies, E. Öhman, K. Kajava, and J. Tiedemann, "LT@Helsinki at SemEval-2020 task 12: Multilingual or language-specific BERT?" in *Proc. 14th Workshop Semantic Eval.* Barcelona, Spain: International Committee for Computational Linguistics, Dec. 2020, pp. 1569–1575.

[33] G. L. De la Peña Sarracén and P. Rosso, "PRHLT-UPV at SemEval-2020 task 12: BERT for multilingual offensive language detection," in *Proc. 14th Workshop Semantic Eval.*, 2020, pp. 1605–1614.

[34] J. Pavlopoulos, J. Sorensen, L. Laugier, and I. Androutsopoulos, "SemEval-2021 task 5: Toxic spans detection," in *Proc. 15th Int. Workshop Semantic Eval. (SemEval)*. Cedarville, OH, USA: Association for Computational Linguistics, 2021, pp. 59–69.

[35] J. A. Meaney, S. Wilson, L. Chiruzzo, A. Lopez, and W. Magdy, "SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense," in *Proc. 15th Int. Workshop Semantic Eval. (SemEval)*. Cedarville, OH, USA: Association for Computational Linguistics, Aug. 2021, pp. 105–119.

[36] S. Hassan, Y. Samih, H. Mubarak, A. Abdelali, A. Rashed, and S. A. Chowdhury, "ALT submission for OSACT shared task on offensive language detection," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools, Shared Task Offensive Lang. Detection*. Marseille, France: European Language Resource Association, May 2020, pp. 61–65.

[37] M. Djandji, F. Baly, W. Antoun, and H. Hajj, "Multi-task learning using AraBert for offensive language detection," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools, Shared Task Offensive Lang. Detection*. Marseille, France: European Language Resource Association, May 2020, pp. 97–101.

[38] M. Á. Álvarez-Carmona, E. Guzmán-Falcón, M. M.-Y. Gómez, H. J. Escalante, L. Villasenor-Pineda, V. Reyes-Meza, and A. Rico-Sulayes, "Overview of MEX-A3T at IberEval 2018: Authorship and aggressiveness analysis in Mexican Spanish tweets," in *Proc. Notebook Papers 3rd SEPLN Workshop Eval. Human Lang. Technol. Iberian Lang. (IberEval)*, seville, spain, vol. 6, 2018, pp. 75–96.

[39] C. Sánchez-Gómez, "INGEOTEC at MEX-A3T: Author profiling and aggressiveness analysis in Twitter using μTC and EvoMSA," in *Proc. OPENAIRE*, 2018, pp. 1–6.

[40] M. E. Aragón and A. P. López-Monroy, "Author profiling and aggressiveness detection in Spanish tweets: MEX-A3T 2018," in *Proc. IberEval@ SEPLN*, 2018, pp. 134–139.

[41] M. E. Aragón, M. A. A. Carmona, M. M.-Y. Gómez, H. J. Escalante, L. V. Pineda, and D. Moctezuma, "Overview of MEX-A3T at IberLEF 2019: Authorship and aggressiveness analysis in Mexican Spanish tweets," in *Proc. IberLEF@ SEPLN*, 2019, pp. 478–494.

[42] M. Casavantes, R. López, and L. C. González-Gurrola, "UACh at MEX-A3T 2019: Preliminary results on detecting aggressive tweets by adding author information via an unsupervised strategy," in *Proc. IberLEF@ SEPLN*, 2019, pp. 537–543.

[43] G. L. D. L. P. Sarracén and P. Rosso, "Aggressive analysis in Twitter using a combination of models," in *Proc. IberLEF@ SEPLN*, 2019, pp. 531–536.

[44] M. E. Aragón, H. J. Jarquín-Vásquez, M. M.-Y. Gómez, H. J. Escalante, L. V. Pineda, H. Gómez-Adorno, J. P. Posadas-Durán, and G. Bel-Enguix, "Overview of MEX-A3T at IberLEF 2020: Fake news and aggressiveness analysis in Mexican Spanish," in *Proc. IberLEF@ SEPLN*, 2020, pp. 222–235.

[45] M. Guzman-Silverio, Á. Balderas-Paredes, and A. P. López-Monroy, "Transformers and data augmentation for aggressiveness detection in Mexican Spanish," in *Proc. IberLEF@ SEPLN*, 2020, pp. 293–302.

[46] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. R. Pardo, P. Rosso, and M. Sanguinetti, "SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter," in *Proc. 13th Int. Workshop Semantic Eval.* Minneapolis, MN, USA: Association for Computational Linguistics, 2019, pp. 54–63.

[47] M.-A. Tanase, G.-E. Zaharia, D.-C. Cercel, and M. Dascalu, "Detecting aggressiveness in Mexican Spanish social media content by fine-tuning transformer-based models," in *Proc. IberLEF@ SEPLN*, 2020, pp. 236–245.

[48] M. Casavantes, R. López, and L. C. González-Gurrola, "UACh at MEX-A3T 2020: Detecting aggressive tweets by incorporating author and message context," in *Proc. IberLEF@ SEPLN*, 2020, pp. 273–279.

[49] J. A. García-Díaz, S. M. J. Zafra, and R. Valencia-García, ''UMUTeam at MeOffendEs 2021: Ensemble learning for offensive language identification using linguistic features, fine-grained negation, and transformers,'' in *Proc. Iberian Lang. Eval. Forum (IberLEF), 37th Int. Conf. Spanish Soc. Natural Lang. Process. (SEPLN)*, vol. 2943, M. Montes, P. Rosso, J. Gonzalo, M. E. Aragón, R. Agerri, M. Á. Á. Carmona, E. Á. Mellado, J. Carrillo-de-Albornoz, L. Chiruzzo, L. A. de Freitas, H. Gómez-Adorno, Y. Gutiérrez, S. M. J. Zafra, S. Lima, F. M. P. del Arco, and M. Taulé, Eds. Málaga, Spain: CEUR, Sep. 2021, pp. 329–345.

[50] F. M. Plaza-del-Arco, A. Montejo-Ráez, L. A. Ureña-López, and M.-T. Martín-Valdivia, ''OffendES: A new corpus in Spanish for offensive language research,'' in *Proc. Conf. Recent Adv. Natural Lang. Process. (RANLP)*, 2021, pp. 1096–1108.

[51] S. T. Aroyehun and A. Gelbukh, ''Evaluation of intermediate pre-training for the detection of offensive language,'' in *Proc. Iberian Lang. Eval. Forum (IberLEF)*, 2021, pp. 313–320.

[52] J. M. Pérez, D. A. Furman, L. A. Alemany, and F. Luque, ''RoBERTuito: A pre-trained language model for social media text in Spanish,'' in *Proc. Lang. Resour. Eval. Conf. (LREC)*, 2022, pp. 1–9.

[53] P. Lin and X. Luo, ''A survey of sentiment analysis based on machine learning,'' in *Proc. CCF Int. Conf. Natural Lang. Process. Chin. Comput.* Cham, Switzerland: Springer, 2020, pp. 372–387.

[54] M. Imran, S. Hina, and M. M. Baig, ''Analysis of learner's sentiments to evaluate sustainability of online education system during COVID-19 pandemic,'' *Sustainability*, vol. 14, no. 8, p. 4529, Apr. 2022.

[55] B. Zhang, M. Yang, X. Li, Y. Ye, X. Xu, and K. Dai, ''Enhancing cross-target stance detection with transferable semantic-emotion knowledge,'' in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*. Cedarville, OH, USA: Association for Computational Linguistics, 2020, pp. 3188–3197.

[56] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B.-W. On, ''Fake news stance detection using deep learning architecture (CNN-LSTM),'' *IEEE Access*, vol. 8, pp. 156695–156706, 2020.

[57] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, ''Enriching word vectors with subword information,'' *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[58] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Prentice-Hall, Jan. 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

[59] O. Ram, Y. Kirstain, J. Berant, A. Globerson, and O. Levy, ''Few-shot question answering by pretraining span selection,'' in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 3066–3079.

[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[61] L. Tunstall, L. V. Werra, and T. Wolf, *Natural Language Processing With Transformers*. Springfield, MO, USA: O'Reilly, 2022.

[62] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez, ''Spanish pre-trained BERT model and evaluation data,'' in *Proc. ICLR*, 2020, pp. 1–9.

JOSÉ MARÍA MOLERO received the first degree in computer science engineering from the University of Seville, and the second degree in engineering and data science from Universidad Nacional de Educación a Distancia (UNED), in 2022.

He is currently a data engineer of the financial sector with company. He was a consultant in the business intelligence and quality assurance projects.

JORGE PÉREZ-MARTÍN received the degree in IT business management from Complutense University of Madrid, in 2014, and the master's degree in advanced artificial intelligence and the Ph.D. degree in intelligent systems from Universidad Nacional de Educación a Distancia (UNED), in 2015 and 2019, respectively.

He is currently an Assistant Professor with the Department of Artificial Intelligence, UNED. He is a member of the Research Centre for Intelligent Decision-Support Systems and miniXmodular, a teaching innovation group. He has participated in six research projects on different applications of artificial intelligence to medicine and health technology assessment and six teaching innovation projects related to accessible interactive materials. He received the Pre-Doctoral Grant from the Spanish Ministry of Education.

ALVARO RODRIGO is an Assistant Professor with the Computer Science Faculty, Universidad Nacional de Educación a Distancia (UNED). He researches on natural language processing, mainly in the areas of question answering systems and their evaluation. Besides, he has been taking part in the organization of question answering evaluations with the Cross Language Evaluation Forum, since 2006. He has involved in several Spanish research projects. He serves as a reviewer for several international journals and conferences.

ANSELMO PEÑAS received the Ph.D. degree (Hons.) from the NLP and IR Group, Universidad Nacional de Educación a Distancia (UNED), Spain, in 2002.

He is a Full Professor with the School of Informatics, UNED. In 2010, he stayed with the University of Southern California, as a Visiting Scholar, where he has collaborated with the DARPA's Machine Reading Program for a year. In 2016, he stayed with the University of York, for six months, working on unsupervised machine learning techniques applied to natural language interpretation. His research aims at the machine interpretation of the natural language and its applications to tasks, such as question answering or machine reading. He has written more than 80 papers on this field. He has participated in several EU projects (EuroWordNet, CLEF, TrebleCLEF, NEWS, ETB, and Limosine); the Project International Coordinator of EU CHIST-ERA READERS, from 2013 to 2015, and HAMiSoN, from 2023 to 2025; and a Principal Investigator of EU CHIST-ERA LIHLITH-KIQA, from 2018 to 2020. From 2007 to 2015, he was a international coordinator of the European Question Answering Benchmarking and Evaluation Campaigns in multiple European languages with the Cross-Language Evaluation Forum (CLEF QA Track). He received the Award of the Spanish Society for Natural Language Processing. He has chaired the CLEF Conference, in 2012, the EACL 2017 Demonstration Sessions, and several workshops on question answering. He is also a member of several program committees of the main conferences in the area (ACL, EMNLP, COLING, NAACL, and EACL). As a result of this international collaboration, he has coauthored articles with more than 15 foreign researchers.