Classifying YouTube Comments At SSD-2026: Enhancing RoBERTa With An XLSTM-like Head

Axel Francisco Leon-Paloma[1], Jaime Stack-Sánchez[1]

[1]Mathematics Research Center (CIMAT), Jalisco S/N Valenciana, 36023 Guanajuato, GTO México.

Abstract

This paper presents the model used in the SSD (Social Support Detection)-2026 challenge to classify text from social media sources. This competition featured three different tasks and two different datasets: one in English and another in Spanish, both composed of YouTube comments. To address these tasks, we proposed a model composed of two stages. First, we trained a Siamese RoBERTa-based model to obtain high-quality hidden representations for the text. Then, in the second stage, we added an xLSTM-based model to obtain more accurate results by leveraging the capabilities of the mLSTM and sLSTM cells from xLSTM. Using this procedure, we conclude that adding the second stage and using a sufficiently large training dataset leads to better results in most tasks with the provided data. These improvements enabled our approach to achieve the best Macro F1 score for Task 3 and the third-best Macro F1 score for Task 1 on the Spanish dataset.

Keywords

Social Support Detection, Text Classification, RoBERTa, xLSTM, Focal Loss, Social Media Analysis

1. Introduction

The emergence of social networks has brought the benefit of instant communication and, moreover, has provided people with a platform to share their ideas and opinions. In some contexts, this also creates opportunities to discuss these ideas, expressing support for or disapproval of different topics. At first glance, this appears entirely beneficial; however, just as people engaged in heated discussions and delivered speeches with particular connotations before the advent of social networks, the same phenomenon now occurs within online platforms, except that it reaches a much broader audience. Nowadays, comments on social media matter, as they can be used as tools of persuasion and can mobilize public opinion in support of one cause or another. A clear example can be observed during election periods [1, 2], when politicians are highly concerned with their image on social media and with comments regarding their character, work, and values. In this and many other contexts, monitoring the nature of comments on social media platforms becomes important, and this task can be assisted by powerful tools such as neural networks, leveraging advances developed in the field of natural language processing for text classification.

With the introduction of the Transformer architecture [3], deep learning significantly enhanced its ability to process and learn from data, particularly sequential data. The Transformer is an encoder–decoder architecture designed to process sequential data in parallel. For specific tasks, several improvements to the original architecture have been proposed, and one of the most influential models was BERT [4], introduced in 2018. The BERT model exploits the capabilities of the Transformer encoder to generate accurate representations of texts, which are particularly useful for discriminative tasks. One of the most important contributions of BERT was demonstrating the effectiveness of pre-trained models, which are trained on large-scale datasets and can subsequently be adapted to downstream tasks through fine-tuning, often achieving state-of-the-art performance.

The BERT model, in turn, gave rise to a wide set of open-source models oriented toward improving its performance, sometimes on specific tasks. RoBERTa [5] was one of the models that emerged to improve the performance of BERT, and its motivation was the belief that BERT was undertrained, a

IberLEF 2026, September 2026, León, Spain

axel.leon@cimat.mx (A. F. Leon-Paloma); jaime.stack@cimat.mx (J. Stack-Sánchez)

https://github.com/axelflp/ (A. F. Leon-Paloma); https://github.com/JaimeStack (J. Stack-Sánchez)

hypothesis that was supported when RoBERTa surpassed BERT, with both models using a maximum input length of 512 tokens.

Given that the tasks proposed in SSD-2026[6][7][8][9][10][11] consist of classifying brief texts, namely YouTube comments, we selected the RoBERTa architecture for the first stage of our model to obtain accurate embeddings for the YouTube comments in the provided dataset.

For the second stage, we added a head composed of sLSTM and mLSTM cells, which are part of the xLSTM model [12], a model designed to outperform the classic LSTM [13] model and narrow the performance gap with Transformers. This model focuses on capturing and exploiting local dependencies and global context in sequential data. In this work, we view these capabilities and those of the RoBERTa architecture not as competing alternatives but as complementary, using the cells of xLSTM as a tool to decode the information contained in the embeddings generated by the RoBERTa architecture in order to make more accurate predictions.

However, despite having a powerful architecture, it is equally important to employ an appropriate loss function to minimize prediction error. In the context of classification, the most common loss function is Binary Cross-Entropy (BCE), with specific variants depending on the type of classification problem (binary, multiclass, or multilabel). Although BCE has proven effective across a wide range of applications, it may encounter difficulties when dealing with challenging data characteristics, such as class imbalance or noisy labels. To address these limitations, numerous extensions and improvements have been proposed over the years, including Focal Loss [14], which was designed to mitigate the bias caused by class imbalance, and label smoothing, which enhances the generalization capability of the model.

The model presented in this work leverages the capabilities of RoBERTuito, a pre-trained RoBERTa-based model[1], to encode text using an arrangement similar to the Siamese architecture proposed in [15]. We fine-tune this model on the training dataset provided by the SSD-2026 challenge [6] to obtain high-quality hidden representations of the text. Subsequently, we use these representations to train a model based on xLSTM cells [12].

The main contributions of the present work are:

- We propose a two-stage architecture that combines Siamese RoBERTa-based embeddings with an xLSTM-based model, allowing the refinement of contextual representations obtained from pre-trained language models.
- We employ simple aggregation mechanisms and lightweight classification heads, encouraging the hidden representations to become sufficiently discriminative so that class separation can be achieved without relying on complex prediction layers.
- We modified the Focal Loss formulation proposed in [16] by introducing a simplified update rule for β that incorporates a momentum factor and removes the square-root operation, improving numerical stability.

2. Related Work

BERT-based models. Just as the Transformer model gave rise to numerous variants and extensions, BERT-based models also emerged and demonstrated improved performance over the original architecture. One of the most notable examples is RoBERTa [5], which introduced significant modifications to the training procedure while making only minor changes to the architecture itself. BERT is trained using two objectives, Next Sentence Prediction (NSP) and Masked Language Modeling (MLM), whereas RoBERTa relies exclusively on MLM with dynamic masking. In addition, it employs byte-level BPE and increases the amount of training data, the number of training steps, the batch size, and the vocabulary size, based on the observation that BERT was undertrained. As a result, RoBERTa became a highly effective pre-trained model and is now widely used throughout the research community.

[1]Throughout the paper, we use the name RoBERTa because it is the name of the architecture behind RoBERTuito.

Embedded BERT models. One approach to enhance the capabilities of BERT models is to integrate them into larger architectures. In this context, we can mention some techniques that have been explored over the years:

- **Integration into Retrieval-Augmented Generation (RAG) systems**, where BERT models serve as the retrieval component within a broader generation pipeline. This includes:
  - *Dual-encoder retrievers*, in which two independent BERT instances encode queries and passages for similarity-based retrieval [17],
  - *Retriever-generator architectures*, where BERT-derived encoders are integrated into end-to-end retrieval-augmented systems for knowledge-intensive tasks [18].
- **Integration into encoder-decoder architectures**, where BERT occupies the encoder role within a sequence-to-sequence generation pipeline. This includes:
  - *Encoder initialization from BERT checkpoints*, pairing BERT-based encoders with generative decoders for sequence generation tasks [19],
  - *Unified encoder-decoder pretraining*, where BERT-inspired representations are integrated into sequence-to-sequence architectures such as BART [20],
  - *Text-to-text frameworks*, where pre-trained encoder-decoder architectures generalize BERT-style pretraining to a wide range of generation tasks [21].
- **Integration into hybrid architectures with specialized modules**, where BERT is combined with other network types to enrich representations. This includes:
  - *BERT with Graph Neural Networks (GNNs)*, where contextual representations generated by BERT are further processed through graph-based reasoning layers for text classification and knowledge-enhanced learning [22, 23],
  - *BERT with recurrent or convolutional modules* (e.g., BERT+BiLSTM, BERT+CNN), commonly employed in text classification and sequence labeling tasks [24],
  - *BERT integrated with domain-specific architectures*, where BERT embeddings are combined with specialized neural components to address tasks in areas such as biomedical and molecular interaction prediction [25].

xLSTM. Given that the Transformer architecture largely surpassed recurrent models such as LSTMs, creating a substantial gap between the two approaches and decreasing interest in research on older recurrent models, a new architecture called Extended Long Short-Term Memory (xLSTM) was introduced in 2024 [12]. The authors' goal was to enhance the traditional LSTM architecture to enable it to compete with Transformers and State Space Models. To this end, the authors proposed a model whose main contributions can be summarized as follows:

- Two new cells, sLSTM and mLSTM, are introduced. Both make use of exponential gating and are stacked sequentially to form the xLSTM architecture.
- The sLSTM cell includes a memory-mixing mechanism that combines information from different heads. This cell operates recurrently over time.
- The mLSTM cell employs a matrix memory capable of storing multiple states instead of a single state, as is the case with vector memory. Additionally, the matrix memory enables parallelized training, and its update rule has a statistical interpretation related to covariance updates.
- The sLSTM cell captures local dependencies, whereas the mLSTM cell captures global dependencies.

Related tasks/research. Recent works have explored similar problems by analyzing textual content from online communities. For example, [26] released a dataset composed of annotated YouTube comments. We mention this paper because the task proposed with the release of this dataset consists of detecting a positive aspect of YouTube comments, namely hope, which is similar to the task we

address in this work. This is less common, given that most research on social media content focuses on detecting negative aspects.

Sentiment analysis of textual content from social media is a useful tool for a variety of interests. Another example we can mention is [2], where the authors analyzed social media posts during the 2020 U.S. presidential election to infer political alignment, candidate support, and user polarization.

3. Methodology

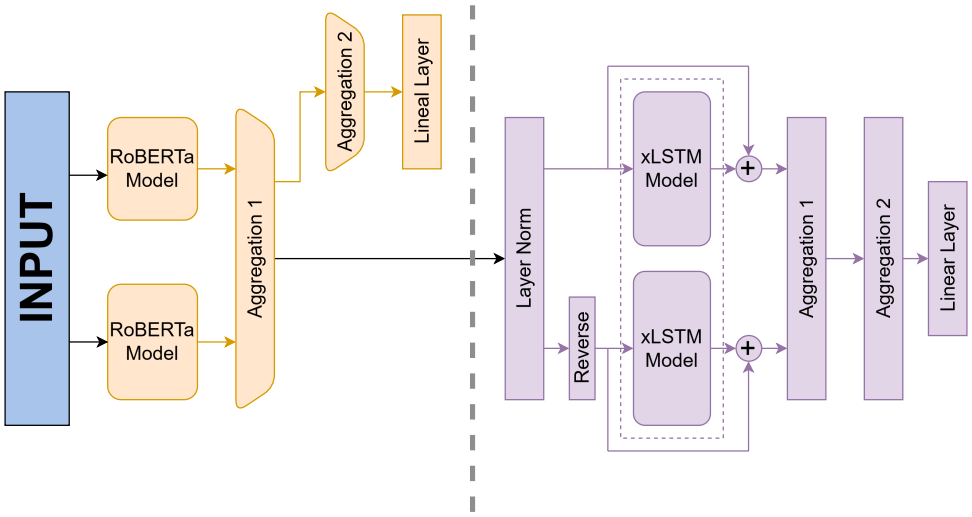

Figure 1: Overview of the proposed two-stage architecture. Left (orange): first-stage model, consisting of a Siamese RoBERTa-based model followed by hidden-state aggregation and classification. Right (purple): second-stage model, consisting of input normalization, sequence reversal, bidirectional xLSTM-based processing, hidden-state aggregation, and classification. The dashed line separates the two training stages.

In this section, we present the proposed model developed to classify text from social media sources and address the tasks of the challenge.

3.1. Pipeline

In Figure 1, we present the two stages, separated by a dashed line. In the first stage, we pass the input independently, a tokenized YouTube comment, through two RoBERTa models and then aggregate the outputs with two aggregation layers, Aggregation 1 and Aggregation 2, in order to obtain a single representation vector for each sentence. We divide the aggregation process into two steps because, in the first step, we aggregate only the pairs of vectors coming from the RoBERTa models while preserving the sequence-length dimension. In the second step, we collapse this dimension by aggregating all the hidden vectors corresponding to one comment into a single vector. This last vector is then passed through a linear layer (classification layer), which converts the vectors from dimension 768 to a scalar for binary classification or into a vector whose dimension equals the number of classes for multiclass classification. This first stage defines a complete model that we train and use to make predictions, but to improve its performance, we propose a second stage.

Once we have trained the model described in the first stage, we freeze its weights and use the output up to its first aggregation as the embedding input to train the xLSTM-based model that defines the second stage. This model begins by normalizing the input and creating a copy of the resulting output. We reverse the order of the copied representation along the sequence-length dimension for those vectors that are not padding vectors (zero vectors). After that, we pass these vectors through two xLSTM models, one for the original representation and one for the reversed representation[2], and add skip

[2]This configuration can be regarded as a bidirectional xLSTM.

connections. After this, we pass the vectors through two aggregation layers and one linear layer, using the same configuration as the first stage but without sharing weights.

This second stage defines the final proposed model used to make predictions.

3.2. Siamese RoBERTa-based model for embeddings

The first stage involves fine-tuning a Siamese pre-trained RoBERTa network. Each RoBERTa architecture receives the same tokenized YouTube comment as input so that, despite sharing the same configuration, each model can capture different aspects of the information contained in the text.

Throughout the entire model, covering both stages, we repeatedly apply the masking process to ensure that padding tokens remain represented as zero vectors. This prevents operations such as bias addition and normalization from altering these vectors and consequently introducing noise during training. We incorporate this masking step between the layers of the RoBERTa models, and it constitutes the only modification made to their original architecture.

Once the outputs of the two RoBERTa models are obtained, we aggregate them in a first step as follows. Given x_1 and x_2, the outputs of each RoBERTa model:

$$\begin{aligned} x_3 &= \text{concat}[x_1; x_2], \\ x_4 &= \text{LayerNorm}(x_3), \\ x_5 &= \text{LinearLayer}(x_4), \\ x_6 &= \text{ReLU}(x_5) \end{aligned}$$

For the second training stage, the output of this first aggregation serves as the input to the xLSTM-based model, preserving both the sentence-length dimension and the padding vectors.

Continuing with the first stage, we perform a second aggregation by applying the operation

$$\log(\sum_i \exp(\cdot_i))$$

along the sequence-length dimension to obtain a single vector representation for each sentence.

3.3. xLSTM-based model

Our xLSTM-based model is a simplified version consisting of a simple arrangement of the sLSTM and mLSTM cells rather than the complete model proposed in [12], which is considerably more complex, stacking several sLSTM and mLSTM cells and adding more transformations.

In the xLSTM model, we feed the vectors into an sLSTM cell, and its output is subsequently passed through an mLSTM cell. Within both cells, the operations are performed using four heads. For the sLSTM and mLSTM, we adopt the stabilized versions proposed in [12]. These stabilized versions avoid overflows caused by large values resulting from the application of the exponential function.

For the second training stage, we use the weights of the trained linear layer from the first stage to initialize the weights of the final classification layer.

3.4. Loss Function

As the loss function, we use an improved version of the Focal Loss function [16], which in its original version is defined as:

$$\mathcal{FL} = \alpha \mathcal{CE}(1-p)^{\gamma}.$$

The improved version we use adds a dynamic term β as the exponent of $(1-p)$, but the update rule we use differs from the one presented in [16], simplifying it to provide greater numerical stability. The improved loss function is:

$$\mathcal{FL} = \alpha\mathcal{CE}(1-p)^{\gamma\beta},$$

and the update rule for β proposed in this work follows the operations below:

$$adjustment = (1 - momentum)\,\frac{\mathcal{L}_t - \mathcal{L}_{t-1}}{\mathcal{L}_{t-1}},$$
$$\beta = \beta\,(1 - adjustment)\,,$$

where $momentum$ is a hyperparameter.
β is updated according to the following rationale. The term

$$\frac{\mathcal{L}_t - \mathcal{L}_{t-1}}{\mathcal{L}_{t-1}}$$

measures the relative change in the loss function between consecutive iterations. Multiplying this quantity by $(1 - momentum)$ modulates the impact of that change, preventing the model from overreacting to abrupt variations and reducing the risk of making large updates due to anomalous training signals.

3.5. Training

In total, for each task, we trained the model for 25 epochs, of which 15 correspond to the first training stage and 10 to the second. We used the following configuration: a batch size of 128, gradient accumulation of 4, mixed-precision training, and customized learning-rate schedules for each of the two training stages. The learning rate starts with a linear warmup phase and then follows a CosineAnnealingLR schedule for the remaining iterations, reaching a maximum value of $1e-4$.

For the first training stage, we used the pre-trained model called RoBERTuito [27], which was trained on approximately 500M Spanish tweets.

For the loss function, we used $\gamma = 2.0$ and $momentum = 0.8$, while β was initialized to 0.6.

4. Data and Tasks

The dataset provided in the Social Support Detection at IberLEF 2026 (SSD-2026) challenge [6] is divided into Spanish and English subsets, both containing YouTube comments. In the training set, each instance is annotated with three labels, one for each task.

This challenge defines three tasks where, given a comment, we have to classify it according to the following criteria:

- **Task 1** (Binary): Does the comment express any type of support
- **Task 2** (Binary): Does the support shown apply to a group or, instead, to an individual?
- **Task 3** (Multiclass): Which social group benefits from this support (Black Community, LGBTQ, Nation, Religion, Women, Other)?

> Comment: “Podrían hacer un draw my life de Elon Musk”
> Label 1: Non Support, Label 2: NO, Label 3: NO.

Figure 2: Example instance selected from the Spanish training dataset.

For each language, the class distributions of the datasets are shown in Tables 2 and 1. The label NO identifies instances that are not classified for the corresponding task; therefore, such instances are ignored for that task.

The valid labels for Task 3 correspond to the following class mapping:

Table 1
SSD-2026 spanish training dataset, 2600 instances.

Task 1		Task 2			Task 3						
Non Support	Support	Group	Individual	NO	0	1	2	3	4	5	NO
2010	590	485	134	1981	24	224	47	82	60	48	2115
2600 valid instances		619 valid instances			485 valid instances						

Table 2
SSD-2026 english training dataset, 7998 instances.

Task 1		Task 2			Task 3						
Non-Supportive	Supportive	Group	Individual	NO	0	1	2	3	4	5	NO
6203	1795	1450	345	6203	91	123	786	416	15	19	6548
7998 valid instances		1795 valid instances			1450 valid instances						

- Black Community → 0
- LGBTQ → 1
- Nation → 2
- Other → 3
- Religion → 4
- Women → 5

Additionally, from this point onward, for Task 1 and Task 2, we will use the following class mapping:

- Non-Support/Non-Supportive, Group → 0
- Support/Supportive, Individual → 1

For Task 1 and Task 2, which are binary classification problems, the α value is defined as:

$$\alpha = \frac{negative\_instances}{valid\_instances},$$

where the negative instances correspond to the Non-Supportive/Non Support and Group classes. For Task 3, α_i is defined for each class i as follows:

$$\alpha_i = \frac{i\_instances}{valid\_instances}.$$

For preprocessing, we employed the procedure proposed in [27], which is designed to transform social media language, including emojis and slang, into plain text without losing the information conveyed in the original text.

It is important to mention two things, first some Spanish comments contain emojis, whereas all English comments are plain text. Second, in the Spanish dataset, there are instances that, although labeled as "Non Support" for Task 1, have valid labels for Task 2. Therefore, Table 1 may give the impression that there is a discrepancy between the number of instances labeled as "Support" for Task 1 and the number of valid instances for Task 2.

5. Results

The model and its training procedure were designed to maximize the Macro F1 score, although accuracy is also reported. The global results obtained by the proposed model on the SSD-2026 test dataset [6] are

shown in Tables 3 and 4. Based on these results, we can conclude that adding the second stage improves the results in most cases according to the Macro F1 and accuracy metrics.

It is important to remark that although the accuracy values follow the same trend as the Macro F1 score and increase when the second stage is added, accuracy is not very informative because our datasets are not balanced, as shown in Tables 2 and 1. In this case, the best option to use as a reference for drawing accurate conclusions is the Macro F1 score, which addresses the challenge of imbalanced data.

Our model shows better performance on the Spanish dataset than on the English dataset, and this discrepancy can be attributed to several factors. One relevant difference between the datasets is that the Spanish dataset contains emojis within its instances, which can contribute to making the comments more informative, whereas the English dataset contains only plain text. In addition to the above, it should also be noted that we used the pre-trained model RoBERTuito, a model trained on Spanish tweets, which can introduce a bias in favor of Spanish texts.

One aspect that stands out in Table 4 is the abrupt decrease in the model's performance from the first to the second stage for Task 3. However, if we inspect Table 5, we can observe that this decrease on the English dataset is mainly caused by the lower performance of the second-stage model on class 4[3], which affects the global metrics. The origin of this behavior can be traced back to the distribution of the English training dataset shown in Table 2, where class 4 is the most underrepresented among all classes. The instances of class 4 represent only 1.034% of the valid Task 3 instances. Therefore, due to this underrepresentation, the data from this class may introduce noise during the second stage instead of reinforcing the model's knowledge.

Table 3
Global metrics on the Spanish dataset.

Stage	Task 1		Task 2		Task 3	
	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy
First	0.881	0.917	0.901	0.936	0.943	0.962
Second	**0.883**	**0.918**	**0.906**	**0.936**	**0.950**	**0.966**

Table 4
Global metrics on the English dataset.

Stage	Task 1		Task 2		Task 3	
	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy
First	0.822	0.878	**0.859**	**0.913**	**0.815**	**0.848**
Second	**0.828**	**0.881**	0.851	0.911	0.738	0.826

Following the class-wise analysis presented in Table 5, we observe that, for Task 1 and Task 2, the highest scores correspond to the predominant class in each dataset. In contrast, for Task 3 on the Spanish dataset, the scores do not exhibit abrupt performance drops across classes. Although there appears to be a preference for some classes, particularly those for which perfect scores are achieved, in general, there are no substantial differences in the scores across classes. On the other hand, on the English dataset, our model shows considerably larger performance gaps across classes in Task 3. In particular, for some classes, we obtain low scores despite these classes not being underrepresented. For example, Class 3 is the second most frequent class in the English dataset, yet we obtain the second-lowest scores in Task 3 for this class. This behavior may be attributed to the apparent bias of the model toward the Spanish dataset, discussed previously, together with the lack of potentially informative features in the English dataset, such as emojis, resulting in less consistent performance.

[3]Highlighted in red.

Table 5
Class-wise metrics. For each task–metric pair, the highest score is highlighted in bold.

Class	Task 1				Task 2				Task 3			
	F1-Score		Recall		F1-Score		Recall		F1-Score		Recall	
	1st	**2nd**	**1st**	**2nd**	**1st**	**2nd**	**1st**	**2nd**	**1st**	**2nd**	**1st**	**2nd**
Spanish dataset												
0	0.946	**0.947**	0.950	**0.952**	**0.960**	0.959	**0.991**	0.975	0.857	0.923	**1.0**	**1.0**
1	0.816	0.819	0.805	0.805	0.843	0.852	0.750	0.805	0.990	0.990	0.982	0.982
2									0.869	0.869	0.833	0.833
3									0.974	0.950	0.950	0.950
4									0.967	0.967	**1.0**	**1.0**
5									**1.0**	**1.0**	**1.0**	**1.0**
English dataset												
0	0.922	**0.923**	**0.924**	0.921	**0.946**	0.945	0.933	**0.939**	0.826	0.772	0.826	0.739
1	0.722	0.733	0.716	0.739	0.771	0.757	0.820	0.782	**0.952**	**0.952**	0.967	0.967
2									0.888	0.868	0.872	0.826
3									0.761	0.752	0.769	0.817
4									0.666	0.250	**1.0**	0.250
5									0.8	0.833	0.8	**1.0**

Since these tasks are not associated with an application-specific context that requires prioritizing some classes over others, the global evaluation is based on Macro F1-score and Accuracy. For the class-wise analysis, we report the F1-score and Recall for each class and stage. Recall is included because, although it is not equivalent to Accuracy, it provides a similar class-oriented interpretation by measuring the proportion of correctly identified instances within each actual class.

5.1. Comparison with a Standard Loss Function

To show that the choice of the improved Focal Loss is appropriate, we replicate the same training process described above, but replace the loss function with Cross-Entropy, one of the most commonly used loss functions for classification. Table 6 shows that, in most cases, except for Task 1 on the Spanish dataset, the improved Focal Loss function performs better. In particular, our model shows a more notable improvement in Task 3, where the Macro F1 score presents a gap of 0.121 and 0.053 on the Spanish and English datasets, respectively, between our proposed model and the best score obtained by the model trained with Cross-Entropy Loss. On the other hand, the accuracy score follows the same tendency as the Macro F1 score.

To demonstrate the benefit of the improved Focal Loss, Table 7 presents the class-wise scores obtained using Cross Entropy Loss, following the same format as Table 5. A comparison between both tables shows that, for Task 1 and Task 2, the scores for Class 0, the majority class in both datasets, remain almost unchanged. In contrast, the performance for Class 1, the minority class, drops considerably when Cross Entropy Loss is used instead of the improved Focal Loss.

For Task 3, the difference in performance is even more evident. In the Spanish dataset, Cross Entropy Loss yields an F1-score and a recall of 0.0 in the first stage for Class 0, the most underrepresented class, together with substantially lower scores in the second stage. In the English dataset, Class 4, which is the most underrepresented class, remains the most difficult to classify. On the other hand, the performance of the majority classes (Class 1 in the Spanish dataset and Class 2 in the English dataset) decreases only

Table 6
Metrics of our model trained with the Cross Entropy loss function against the best scores of our original model.

Stage	Task 1		Task 2		Task 3	
	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy
Spanish dataset						
First	0.890	0.923	0.890	0.929	0.632	0.826
Second	**0.887**	**0.923**	0.879	0.923	0.829	0.892
Our model (Best)	0.883	0.918	**0.906**	**0.936**	**0.950**	**0.966**
English dataset						
First	0.800	0.864	0.847	0.909	0.757	0.823
Second	0.801	0.870	0.848	0.909	0.762	0.826
Our model (Best)	**0.828**	**0.881**	**0.859**	**0.913**	**0.815**	**0.848**

Table 7
Class-wise metrics by language, using Cross Entropy Loss. For each task–metric pair, the highest score is highlighted in bold.

Class	Task 1				Task 2				Task 3			
	F1-Score		Recall		F1-Score		Recall		F1-Score		Recall	
	1st	**2nd**	**1st**	**2nd**	**1st**	**2nd**	**1st**	**2nd**	**1st**	**2nd**	**1st**	**2nd**
Spanish dataset												
0	**0.950**	**0.950**	0.952	**0.962**	**0.956**	0.952	**0.991**	**0.991**	0.0	0.571	0.0	0.666
1	0.831	0.825	0.825	0.791	0.825	0.806	0.722	0.694	0.955	0.964	0.964	0.964
2									0.285	0.769	0.166	0.833
3									0.679	0.894	0.9	0.85
4									0.875	0.866	0.933	0.866
5									**1.0**	0.909	**1.0**	0.833
English dataset												
0	0.913	**0.918**	0.917	**0.935**	**0.944**	**0.944**	**0.939**	0.936	0.808	0.717	0.826	0.608
1	0.687	0.685	0.675	0.639	0.75	0.753	0.769	0.782	**0.937**	**0.937**	**0.967**	**0.967**
2									0.864	0. 875	0.826	0.846
3									0.737	0.743	0.798	0.807
4									0.4	0.5	0.25	0.5
5									0.8	0.8	0.8	0.8

slightly when Cross Entropy Loss is employed.

These results indicate that the improved Focal Loss consistently enhances performance across all tasks and datasets, with the largest gains observed for minority classes.

5.2. Baseline for the English Dataset

The pre-trained model RoBERTuito is particularly well suited to the Spanish dataset because it was pre-trained on a large Spanish dataset composed of tweets. However, for the English dataset, the

question remains: why not use a model pre-trained on an English dataset? To address this issue, we trained our model by replacing RoBERTuito with cardiffnlp/twitter-roberta-base, a pre-trained model available on Hugging Face that was also trained on a dataset composed of tweets, but in this case in English. Apart from the training dataset, there is another difference between both models: whereas RoBERTuito was trained from scratch following the RoBERTa guidelines, cardiffnlp/twitter-roberta-base was trained on top of the original RoBERTa-base checkpoint.

Table 8
Comparison of scores on the English dataset between RoBERTuito and cardiffnlp/twitter-roberta-base as backbone models, with cardiffnlp/twitter-roberta-base used as the baseline.

Stage/Backbone model	Task 1		Task 2		Task 3	
	Macro F1	Accuracy	Macro F1	Accuracy	Macro F1	Accuracy
First (RoBERTuito)	0.822	0.878	**0.859**	**0.913**	**0.815**	**0.848**
Second (RoBERTuito)	**0.828**	**0.881**	0.851	0.911	0.738	0.826
First (baseline)	0.823	0.878	0.841	0.900	0.657	0.837
Second (baseline)	0.822	0.8785	0.835	0.897	0.666	0.840

As shown in Table 8, using RoBERTuito leads to higher scores, despite the baseline having the advantage of being trained on an English dataset. This may be due to one important difference between the two models: RoBERTuito was trained on 500 million tweets, whereas the baseline was trained on only approximately 58 million. This represents a substantial difference in the training dataset size, which may explain why, despite not being trained on an English dataset, RoBERTuito still has a stronger ability to understand short texts coming from social media.

Another aspect to highlight from Table 8 is the gap between the scores. To explain this, it is important to recall that, according to Table 2, the English training dataset contains 7998, 1795, and 1450 valid instances for Task 1, Task 2, and Task 3, respectively. The effect of this decrease in dataset size can be observed in Table 9, where we report the maximum F1 score for each backbone model. In particular, the smaller the training dataset size, the larger the gap in favor of the RoBERTuito model. We use only the F1 score for this analysis because, as mentioned above, accuracy is not reliable enough in the presence of unbalanced data.

Table 9
Comparison of the best Macro F1 scores obtained on the English dataset using RoBERTuito and cardiffnlp/twitter-roberta-base as backbone models. The gap is computed as the best Macro F1 score obtained with RoBERTuito minus the best Macro F1 score obtained with the baseline model.

Backbone model	Task 1	Task 2	Task 3
	Macro F1	Macro F1	Macro F1
RoBERTuito	0.828	0.859	0.815
Baseline	0.823	0.841	0.666
Gap	0.005	0.018	0.149

6. Conclusions

In this work, we presented a two-stage architecture for social support detection in social media comments. The proposed approach combines the representational capabilities of Siamese RoBERTa-based models with an xLSTM-based module designed to further refine the hidden representations obtained from the pre-trained language model.

Table 10
Ranking table of the SSD-2026 [6] challenge for Task 1 and Task 3 on the Spanish dataset.

Rank	Spanish Task 1		Spanish Task 3	
	username	Macro F1 Score	username	Macro F1 Score
1	davidsotoesime	0.889	**axelflp**	0.9502
2	0287289	0.886	osornio_up	0.933
3	**axelflp**	0.883	adriglup	0.928

A key aspect of the proposed methodology is the use of simple aggregation mechanisms and lightweight classification heads. This design allows the learning process to focus on generating robust hidden representations rather than relying on highly complex classifiers.

Additionally, we simplified the β update rule, thereby providing additional stability during training while preserving the advantages of the improved Focal Loss formulation [16].

The obtained results demonstrate the effectiveness of the proposed approach for the SSD-2026 challenge, achieving competitive performance across the evaluated tasks and obtaining the best Macro F1 score for Task 3 and the third-best Macro F1 score for Task 1 on the Spanish dataset, as shown in Table 10. These results suggest that combining contextual embeddings with xLSTM-based representation refinement constitutes a promising direction for social media text classification, with better performance on Spanish inputs. This latter observation may be due to different factors, such as the pre-trained model used, the training dataset size, and the fact that some Spanish comments contain emojis.

7. Future work

In the proposed model, we use RoBERTuito because of the similarity between the dataset on which it was trained and the dataset provided for the SSD-2026 challenge. However, there is still a wide range of models that can be used to replace it in the process of generating our embeddings in the first stage, starting with other BERT-based models.

On the other hand, we mentioned that we only used specific cells of the xLSTM model. However, the full xLSTM is a richer model, and it is possible to incorporate more ideas from the original architecture and arrange these cells in a way that improves performance.

Acknowledgments

The authors thank CIMAT Bajio Supercomputing Laboratory (#300832) for the computational resources provided for this project.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5.5 to create the tables included in this paper, in order to obtain the desired arrangement of the data while preserving the formatting style. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] L. Belcastro, F. Branda, R. Cantini, F. Marozzo, D. Talia, P. Trunfio, Analyzing voter behavior on social media during the 2020 us presidential election campaign, Social Network Analysis and Mining 12 (2022) 83.

[2] L. Belcastro, R. Cantini, F. Marozzo, D. Talia, P. Trunfio, Learning political polarization on social media using neural networks, IEEE access 8 (2020) 47177–47187.
[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems 30 (2017).
[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: https://arxiv.org/abs/1810.04805. arXiv:1810.04805.
[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.
[6] M. Shahiki Tash, L. Ramos, A. Zahra, A. Y. Barrera Animas, F. Balouchzahi, O. Kolesnikova, G. Sidorov, A. Gelbukh, R. Monroy, F. Vargas, Overview of SSD-2026 at IberLEF 2026: Social Support Detection, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2026), co-located with the 42nd Conference of the Spanish Society for Natural Language Processing (SEPLN 2026), CEUR-WS. org, 2026.
[7] A. Bonet-Jover, J. Á. González-Barba, L. Chiruzzo, Overview of IberLEF 2026: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2026), co-located with the 42nd Conference of the Spanish Society for Natural Language Processing (SEPLN 2026), CEUR-WS. org, 2026.
[8] Z. Ahani, M. Shahiki Tash, F. Balouchzahi, L. Ramos, G. Sidorov, A. Gelbukh, Social support detection from social media texts, PLOS ONE (2026) 1–25. URL: https://doi.org/10.1371/journal.pone.0337476.
[9] O. Kolesnikova, M. Shahiki Tash, Z. Ahani, A. Agrawal, R. Monroy, G. Sidorov, Advanced machine learning techniques for social support detection on social media, Heliyon 11 (2025). URL: https://doi.org/10.1016/j.heliyon.2025.e43437.
[10] M. S. Tash, L. Ramos, Z. Ahani, R. Monroy, O. kolesnikova, H. Calvo, G. Sidorov, Online social support detection in Spanish social media texts, 2025. URL: https://arxiv.org/abs/2502.09640. arXiv:2502.09640.
[11] M. Shahiki-Tash, Z. Ahani, L. Ramos, O. Kolesnikova, A. Y. Barrera-Animas, Linguistic Analysis in Different Types of Support using LIWC for Spanish and English online communities (2026). URL: https://doi.org/10.21203/rs.3.rs-9557287/v1.
[12] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter, S. Hochreiter, xlstm: Extended long short-term memory, 2024. URL: https://arxiv.org/abs/2405.04517. arXiv:2405.04517.
[13] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.
[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, 2018. URL: https://arxiv.org/abs/1708.02002. arXiv:1708.02002.
[15] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL: https://arxiv.org/abs/1908.10084. arXiv:1908.10084.
[16] X. Mao, Z. Li, Q. Li, S. Zhang, Bert-dxlma: Enhanced representation learning and generalization model for english text classification, Neurocomputing 622 (2025) 129325. URL: https://api.semanticscholar.org/CorpusID:275387488.
[17] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W. tau Yih, Dense passage retrieval for open-domain question answering, 2020. URL: https://arxiv.org/abs/2004.04906. arXiv:2004.04906.
[18] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL: https://arxiv.org/abs/2005.11401. arXiv:2005.11401.
[19] S. Rothe, S. Narayan, A. Severyn, Leveraging pre-trained checkpoints for sequence generation tasks, Transactions of the Association for Computational Linguistics 8 (2020) 264–280. URL: http://dx.doi.org/10.1162/tacl_a_00313. doi:10.1162/tacl_a_00313.
[20] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer,

BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL: https://aclanthology.org/2020.acl-main.703/. doi:10.18653/v1/2020.acl-main.703.

[21] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL: https://arxiv.org/abs/1910.10683. arXiv:1910.10683.

[22] Y. Lin, Y. Meng, X. Sun, Q. Han, K. Kuang, J. Li, F. Wu, BertGCN: Transductive text classification by combining GNN and BERT, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 1456–1462. URL: https://aclanthology.org/2021.findings-acl.126/. doi:10.18653/v1/2021.findings-acl.126.

[23] L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, 2018. URL: https://arxiv.org/abs/1809.05679. arXiv:1809.05679.

[24] C. Sun, L. Huang, X. Qiu, Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 380–385. URL: https://aclanthology.org/N19-1035/. doi:10.18653/v1/N19-1035.

[25] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2019) 1234–1240. URL: http://dx.doi.org/10.1093/bioinformatics/btz682. doi:10.1093/bioinformatics/btz682.

[26] B. R. Chakravarthi, HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion, in: M. Nissim, V. Patti, B. Plank, E. Durmus (Eds.), Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, Association for Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 41–53. URL: https://aclanthology.org/2020.peoples-1.5/.

[27] J. M. Pérez, D. A. Furman, L. A. Alemany, F. Luque, Robertuito: a pre-trained language model for social media text in spanish, 2022. URL: https://arxiv.org/abs/2111.09453. arXiv:2111.09453.