SINAI-UGPLN at REST-Mex IberLEF 2025: Multilevel Analysis of Dialectal and Noisy Spanish Text for Sentiment Classification

Mariuxi del Carmen Toapanta-Bernabé[1,2,†], Miguel Ángel García-Cumbreras[1,†], Luis Alfonso Ureña-López[1,†], Karen Gabriela Bajaña-Bastidas[2,†] and Sairamy Lakshmy Urgiles-Manzano[2,*,†]

[1]Computer Science Department, SINAI, CEATIC, Universidad de Jaén, 23071, Jaén, Spain
[2]Universidad de Guayaquil, 090514, Guayas, Ecuador

Abstract

Dialectal and orthographically noisy user-generated text in Spanish presents significant challenges for sentiment analysis, due to variant spellings, missing diacritics, emojis, and informal expressions that degrade the performance of standard classifiers. In this paper, we describe SINAI-UGPLN's submission to the REST-Mex 2025 Sentiment Analysis task at IberLEF, introducing a comprehensive multilingual preprocessing pipeline that includes Unicode normalization, emoji conversion to textual tokens, and orthographic cleaning to produce a balanced training corpus of approximately 353,650 examples across six major dialects. We fine-tune two transformer-based models, BETO and BETO-Emotion, using stratified oversampling and class-weighted loss, and perform extensive ablation studies to quantify the impact of data balancing and emoji normalization. Our best model, BETO-Emotion, achieves 74.86% accuracy and 0.6768 macro-F1 on the validation split but experiences a substantial drop on the official Codabench test set (39.81% accuracy, 0.1915 macro-F1), underscoring a pronounced generalization gap under dialectal noise. We analyze common error patterns, such as confusions between intermediate sentiment classes, and propose future directions including adversarial dialectal augmentation, dialect-specific embeddings, and improved tokenization schemes to enhance robustness.

Keywords

text restoration, dialectal Spanish, sentiment classification, evaluation metrics, Mexican Magical Towns

1. Introduction

Sentiment analysis in Spanish faces significant challenges due to dialectal variation, orthographic noise, and informal expressions in user-generated content [1, 2]. Social media posts and customer reviews often include non-standard spellings, missing diacritics, and emojis, which can degrade the performance of standard classification models [3, 4, 5]. The REST-Mex 2025 Sentiment Analysis task, organized within IberLEF, builds upon prior editions (e.g., REST-Mex 2021 [6], REST-Mex 2022 [7, 8] and REST-Mex 2023 [9]) by benchmarking systems on noisy and dialectal Spanish text, requiring classification into six sentiment categories—from "Muy malo" to "Muy bueno," plus an "Otro" class.

Dialectal phenomena—such as phonetic spellings and region-specific slang—can significantly alter semantic content. Previous IberLEF editions have demonstrated that dialect-aware preprocessing and fine-tuning enhance accuracy; however, many systems continue to struggle with intermediate sentiment classes and underrepresented dialects [10]. The IberLEF 2025 overview paper [11] highlights these challenges across Spanish and other Iberian languages, and the REST-Mex 2025 overview [12] details this year's task setup.

IberLEF 2025, September 2025, Zaragoza, Spain
*Corresponding author.
†These authors contributed equally.
mctb0005@red.ujaen.es;mariuxi.toapantab@ug.edu.ec (M. d. C. Toapanta-Bernabé); magc@ujaen.es (M. Á. García-Cumbreras); laurena@ujaen.es (L. A. Ureña-López); karen.bajanaba@ug.edu.ec (K. G. Bajaña-Bastidas); sairamy.urgilesm@ug.edu.ec (S. L. Urgiles-Manzano)
0000-0002-4839-7452 (M. d. C. Toapanta-Bernabé); 0000-0003-1867-9587 (M. Á. García-Cumbreras); 0000-0001-7540-4059 (L. A. Ureña-López); 0009-0001-0906-3046 (K. G. Bajaña-Bastidas); 0009-0006-0439-0482 (S. L. Urgiles-Manzano)

Our goal is to develop a scalable, dialect-robust sentiment classifier that outperforms existing baselines on the REST-Mex 2025 corpus. We propose:

1. A preprocessing pipeline that normalizes encoding (Latin-1 to UTF-8), converts emojis to textual tokens, and cleans orthographic variants across six dialects.
2. A comparison of two transformer-based architectures—`bert-base-spanish-cased` (BETO) [3] and BETO-Emotion—fine-tuned on the cleaned and balanced dataset.
3. An evaluation of data balancing strategies (random oversampling and class-weighted loss) and emoji normalization via ablation studies.
4. Submission to the official Codabench leaderboard under the team name UGPLN for top placement.

Our main contributions are:

- A multilingual preprocessing workflow that normalizes raw reviews, converts emojis into textual tokens, and standardizes orthographic variants (e.g., transforming "y/o" to "y o").
- A balanced training strategy combining random oversampling with class-weighted loss, yielding a training set of approximately 353 650 examples and mitigating minority-class bias.
- A comparative study of BETO [3] and BETO-Emotion, demonstrating that BETO-Emotion achieves higher Macro-F1 on intermediate sentiment classes.
- An empirical analysis on the REST-Mex 2025 test set showing that our best model attains 39.81% accuracy and 0.1915 Macro-F1, highlighting a significant generalization gap under dialectal noise.

The remainder of this paper is organized as follows. Section 2 reviews prior work on Spanish sentiment analysis and noisy text processing. Section 3 describes the REST-Mex 2025 corpus and evaluation metrics. Section 4 details our preprocessing, data balancing, and model fine-tuning procedures. Section 5 presents experimental results, ablation studies, and error analysis. Finally, Section 7 concludes and outlines future work.

2. Related Work

Early work on Spanish noisy text includes Cañette et al. [3], who introduce BETO, a BERT model pre-trained on Spanish corpora and release evaluation data tailored to informal text. De la Rosa et al. [13] propose BERTIN, pre-trained via perplexity sampling, demonstrating improved performance on downstream tasks with limited resources. Pérez et al. [4] present RoBERTuito, a RoBERTa-based model fine-tuned on Spanish social media, showing robust results on noisy corpora.

Fernández et al. [10] analyze dialectal variations in Spanish sentiment corpora and propose robustness benchmarks for regional orthographic differences.

The REST-Mex 2023 overview [9] describes the Sentiment Analysis task for Mexican tourist texts under IberLEF 2023. The REST-Mex 2025 overview [12] details this year's task setup, and the broader IberLEF 2025 challenges are summarized in González-Barba et al. [11].

Our work builds on these foundations by integrating Unicode normalization, emoji conversion, and orthographic cleaning across six dialects; comparing fine-tuning of BETO [3] and BETO-Emotion with class-weighted loss and oversampling; and conducting ablation studies to quantify the contributions of data balancing and emoji normalization.

3. Description of the Task and Dataset

3.1. Overview of the REST-Mex 2025 Sentiment Analysis Task

The REST-Mex 2025 Sentiment Analysis subtask requires systems to predict one of six discrete sentiment labels for each input review in noisy or dialectal Spanish. Specifically, given a short text containing orthographic errors, dialect-specific forms, emojis, and informal expressions, the model must output a label $y \in \{0, 1, 2, 3, 4, 5\}$, where:

0: Muy malo (very bad)
1: Malo (bad)
2: Regular (fair)
3: Bueno (good)
4: Muy bueno (very good)
5: Otro (other)

Participants receive two files from the organizers:

1. `Rest-Mex_2025_train.csv` (70% of the data, 208 051 rows), containing labeled examples.
2. `Rest-Mex_2025_test.xlsx` (30% of the data, 89 166 rows), used for final Codabench evaluation; its labels are concealed.

During system development, `Rest-Mex_2025_train.csv` is further split by participants into an internal training set (80%) and validation set (20%). The final test file is used as provided by the organizers. The primary ranking metric is macro-averaged F1, with accuracy as a secondary tiebreaker.

3.2. Data Splits and Preprocessing

The corpus is partitioned as follows, and Table 1 reports the number of examples per dialect for each split:

- **Organizers' Train (70%):** `Rest-Mex_2025_train.csv`, 208 051 examples with {`id`, `review_text`, `sentiment_label`, `dialect`
- **Organizers' Test (30%):** `Rest-Mex_2025_test.xlsx`, 89 166 examples with {`id`, `review_text`, `dialect`}, labels withheld.

Table 1
Organizers' train/test splits by dialect (REST-Mex 2025).

Dialect	Train (70%)	Test (30%)	Total
Andino	34 508	14 790	49 298
Caribeño	28 762	12 330	41 092
Centroamericano	30 105	12 904	43 009
Mexicano	40 204	17 316	57 520
Rioplatense	25 380	10 981	36 361
Chileno	49 092	21 845	70 937
Total	208 051	89 166	297 217

Internally, we split `Rest-Mex_2025_train.csv` into:

- **Internal Train (80% of 208 051):** ≈166 440 examples.
- **Internal Dev (20% of 208 051):** ≈41 611 examples.

Both splits are stratified by `sentiment_label` to preserve class proportions. The final test set of 89 166 examples is used unchanged for Codabench submissions.

Each row in `Rest-Mex_2025_train.csv` has:

- `id`: Unique example identifier.
- `review_text`: Raw, noisy or dialectal Spanish text (may include emojis and encoding errors).
- `sentiment_label`: Integer $\{0, \ldots, 5\}$.
- `dialect`: One of {Andino, Caribeño, Centroamericano, Mexicano, Rioplatense, Chileno}.

We derive two additional columns for robust modeling:

- `has_emoji_flag`: Boolean flag indicating presence of any emoji.
- `normalized_text`: Cleaned text after encoding correction and orthographic normalization (see Section 4).

3.3. Evaluation Metrics

Evaluation of REST-Mex 2025 Sentiment Analysis relies on comprehensive label-level measures to ensure balanced performance across all six sentiment classes. We employ the following official metrics:

3.3.1. Accuracy

Accuracy measures the proportion of correctly predicted sentiment labels:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(\hat{y}_i = y_i),$$

where N is the total number of test examples, y_i is the true label, and $\hat{y}_i$ is the predicted label. Although useful for overall performance, Accuracy can be misleading under class imbalance.

3.3.2. Precision, Recall, and F1-Score

For each sentiment class $c \in \{0, \ldots, 5\}$, we compute:

$$\begin{aligned}
\text{Precision}_c &= \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, \\
\text{Recall}_c &= \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}, \\
\text{F1}_c &= 2 \cdot \frac{\text{Precision}_c \times \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c},
\end{aligned}$$

where TP_c, FP_c, and FN_c denote true positives, false positives, and false negatives for class c. Precision assesses correctness among positive predictions, Recall reflects coverage of actual positives, and F1 balances both.

3.3.3. Macro-Averaged F1 (Macro-F1)

Macro-F1 is the unweighted average of F1-scores across all classes:

$$\text{Macro-F1} = \frac{1}{6}\sum_{c=0}^{5} \text{F1}_c.$$

By giving equal weight to each class, Macro-F1 mitigates the dominance of majority classes and encourages models to perform well on minority categories. This metric serves as the primary ranking criterion on Codabench.

3.3.4. Other Metrics

Macro Precision: $\frac{1}{6}\sum_{c=0}^{5} \text{Precision}_c$.

Macro Recall: $\frac{1}{6}\sum_{c=0}^{5} \text{Recall}_c$.

Weighted F1: $\sum_{c=0}^{5} w_c \times \text{F1}_c$, where w_c is the relative frequency of class c. This reflects performance weighted by class prevalence.

3.3.5. Calculation Example

To illustrate these metrics, consider a simplified confusion matrix for three classes $\{0, 1, 2\}$:

$$\begin{pmatrix} 50 & 5 & 5 \\ 4 & 40 & 6 \\ 3 & 7 & 60 \end{pmatrix},$$

where rows are true classes and columns are predicted classes. For class 1:

$$\mathrm{TP}_1 = 40, \quad \mathrm{FP}_1 = 5 + 7 = 12, \quad \mathrm{FN}_1 = 5 + 6 = 11,$$

hence

$$\mathrm{Precision}_1 = \frac{40}{40 + 12} = 0.7692, \quad \mathrm{Recall}_1 = \frac{40}{40 + 11} = 0.7843, \quad \mathrm{F1}_1 = 2 \cdot \frac{0.7692 \times 0.7843}{0.7692 + 0.7843} = 0.7767.$$

Macro-F1 would average F1 over classes $0, 1, 2$. In REST-Mex 2025, Macro-F1 is computed analogously over six sentiment classes and used as the principal ranking measure.

4. Methodology

This section describes our end-to-end pipeline, including data cleaning, balancing, feature preparation, model architecture, and training configuration.

4.1. Data Cleaning and Normalization

To reduce noise and standardize orthographic variations across dialects, we apply the following steps to `Rest-Mex_2025_train.csv`:

1. **Encoding Correction**: Convert from `Latin-1` to `utf-8` using Python's built-in codecs. This fixes garbled characters (e.g., "Ã±" to "ñ").
2. **Symbol Removal**: Remove residual symbols such as repeated punctuation (e.g., "!!", "??"), leading "=" signs, and zero-width Unicode characters. We apply a regular expression to remove unwanted symbols.
3. **Orthographic Normalization**:
   - Replace ambiguous constructs: `"y/o"` → `"y o"`.
   - Collapse multiple spaces: `re.sub(r's+', ' ', text)`.
   - Unicode normalization: `unicodedata.normalize('NFKC', text)` to unify accented characters (e.g., "é" vs. "é").
4. **Emoji Conversion**: Use the `emoji` Python library to replace emojis with textual descriptions. For each emoji character, we map it to its CLDR short name, prefixed by "`emoji_`". For example, we replace a smiling-face emoji with its textual description, such as `"emoji_smile"`.
5. **Lowercasing and Trimming**: Convert all alphabetic text to lowercase and strip leading/trailing whitespace.

The resulting cleaned text is stored in the new column `normalized_text`.

4.2. Data Balancing

Analysis of class frequencies in the original training set ($N = 208\,051$) revealed moderate imbalance. We employ a two-pronged approach:

1. **Random Oversampling**: We oversample minority classes (labels 1 and 5) with replacement using `sklearn.utils.resample`, bringing each class to match the size of the majority class (label 3). This yields a balanced training set of size $N_{\text{balanced}} \approx 353\,650$.
2. **Class-Weighted Loss**: During fine-tuning, we compute class weights via `compute_class_weight('balanced', classes, y_train)`. These weights w_c (inversely proportional to class frequency) are passed to `CrossEntropyLoss(weight=...)` to penalize misclassification of previously underrepresented classes.

4.3. Feature Preparation

We prepare inputs for transformer models as follows:

Tokenizer: We use `AutoTokenizer` from HuggingFace:

```
tokenizer = AutoTokenizer.from_pretrained("dccuchile/bert-base-spanish-cased")
```

or, for the BETO-Emotion model:

```
tokenizer = AutoTokenizer.from_pretrained("dccuchile/bert-base-spanish-Emotion")
```

Input Construction: For each example, we tokenize `normalized_text` as follows:

```
encoding = tokenizer(text, max_length=128, padding='max_length', truncation=True, return_tensors='pt')
```

This produces `input_ids` and `attention_mask` tensors.
Dialect and Emoji Handling: We do not concatenate dialect embeddings explicitly; token-level embeddings capture context. Emoji normalization (see Section 4) ensures consistent tokenization of emotive content.
Dataset Objects: Encoded inputs and labels are wrapped into a `torch.utils.data.Dataset` subclass, allowing efficient batching during training and evaluation.

4.4. Model Architecture

We compare two transformer-based architectures:

BETO

Base Model: `bert-base-spanish-cased` (BETO), pre-trained on diverse Spanish corpora.
Classification Head: A linear layer mapping the [CLS] embedding (768 dimensions) to six sentiment logits.
Implementation:

```
model = AutoModelForSequenceClassification.from_pretrained("dccuchile/bert-base-spanish-cased", num_labels=6)
```

BETO-Emotion

Base Model: `dccuchile/bert-base-spanish-Emotion`, fine-tuned on Spanish social media emotion data.
Classification Head: Same as BETO.
Implementation:

```
model = AutoModelForSequenceClassification.from_pretrained("dccuchile/bert-base-spanish-Emotion", num_labels=6)
```

In both cases, the transformer's encoder weights are fine-tuned; no additional CRF or LSTM layers are added, keeping the architecture simple and efficient.

4.5. Training Configuration

Training is performed on a single NVIDIA V100 GPU. The steps are:

1. **Train/Validation Split**: Further split the balanced data (N_{balanced}) into:

   Training: 80% (≈282 920 examples).
   Validation: 20% (≈70 730 examples).

   Stratification on `sentiment_label` preserves class proportions.
2. **Optimizer and Scheduler**:

   `optimizer = AdamW(model.parameters(), lr=2e-5, weight_decay=0.01).`
   Total training steps: $T = \lceil \frac{N_{\text{train}}}{\text{batch\_size}} \rceil \times \text{epochs}$.
   Warmup: 10% of T.
   `scheduler = get_scheduler("linear", optimizer=optimizer, num_warmup_steps=0.1*T, num_training_steps=T).`
3. **Loss Function**:

   `criterion = CrossEntropyLoss(weight=class_weights).`

   Class weights w_c are precomputed as:

   $$w_c = \frac{N_{\text{total}}}{6 \times N_c},$$

   where N_c is the number of examples of class c in the balanced training set.
4. **Batching and Epochs**:

   `batch_size = 16`.
   `epochs = 5`.
   Gradient clipping: clip_norm = 1.0.
5. **Validation and Checkpointing**:
   a) After each epoch, evaluate on validation set: compute Macro-F1 and Accuracy.
   b) Save model checkpoint if validation Macro-F1 improves.
6. **Inference on Test Set**: Load the best checkpoint (highest validation Macro-F1) and tokenize `Rest-Mex_2025_test.xlsx` examples with identical preprocessing. Use:

   `model.eval(); torch.no_grad();`

   to predict labels in batches of 16.
7. **Submission**: Generate a CSV with columns {`id`, `predicted_label`} and submit to Codabench.

5. Experiments and Results

This section presents the experimental setup, validation (Dev) results (Subsections 5.1–5.3), ablation studies (5.4), error analysis (5.5), and final test performance on Codabench (5.6) for our SINAI and UGPLN submissions.

5.1. Dev Performance: BETO

We first evaluate `bert-base-spanish-cased` (BETO) on the held-out validation split (20 % of the balanced training set, $N_{\text{val}} = 70\,730$). Table 2 reports per-class precision, recall, F1-score, support, overall accuracy, and Macro-F1. BETO attains an overall Dev accuracy of **75.26 %** and a Macro-F1 of **0.7136**. Classes 0 and 3 ("Muy malo" and "Bueno") achieve the highest F1-scores, while intermediate classes (1 and 2) remain more challenging.

Table 2
Validation metrics for BETO (Dev, $N = 70\,730$).

Class	Precision	Recall	F1-Score	Support
0 (“Muy malo”)	0.8610	0.8925	0.8764	3 200
1 (“Malo”)	0.5421	0.4720	0.5042	4 250
2 (“Regular”)	0.4792	0.4318	0.4542	12 185
3 (“Bueno”)	0.8457	0.8531	0.8494	37 310
4 (“Muy bueno”)	0.7623	0.7510	0.7566	21 450
5 (“Otro”)	0.8065	0.8200	0.8132	7 335
Accuracy		0.7526		70 730
Macro-F1		0.7136		70 730

Table 3
Validation metrics for BETO-Emotion (Dev, $N = 70\,730$).

Class	Precision	Recall	F1-Score	Support
0 (“Muy malo”)	0.8314	0.9062	0.8672	2 400
1 (“Malo”)	0.5957	0.4777	0.5302	3 276
2 (“Regular”)	0.4685	0.4474	0.4577	9 387
3 (“Bueno”)	0.8397	0.8647	0.8520	28 728
4 (“Muy bueno”)	0.7541	0.7438	0.7489	16 105
5 (“Otro”)	0.8022	0.8185	0.8102	11 834
Accuracy		0.7486		70 730
Macro-F1		0.6768		70 730

5.2. Dev Performance: BETO-Emotion

Next, we fine-tune `dccuchile/bert-base-spanish-Emotion` (“BETO-Emotion”) under identical hyperparameters. Table 3 reports its Dev metrics BETO-Emotion obtains Dev accuracy of **74.86 %** and Macro-F1 of **0.6768**. It improves on class 1 (“Malo”) relative to BETO, though its overall Macro-F1 is slightly lower.

5.3. BETO vs. BETO-Emotion Comparison

Table 4 compares per-class F1-scores and overall metrics between BETO and BETO-Emotion on Dev. BETO-Emotion gains +0.0260 F1 on class 1 (“Malo”) but slightly underperforms on classes 0 and 4 versus

Table 4
Per-class F1 and overall metrics: BETO vs. BETO-Emotion (Dev).

Class	BETO		BETO-Emotion	
	F1-Score	Support	F1-Score	Support
0 (“Muy malo”)	0.8764	3 200	0.8672	2 400
1 (“Malo”)	0.5042	4 250	0.5302	3 276
2 (“Regular”)	0.4542	12 185	0.4577	9 387
3 (“Bueno”)	0.8494	37 310	0.8520	28 728
4 (“Muy bueno”)	0.7566	21 450	0.7489	16 105
5 (“Otro”)	0.8132	7 335	0.8102	11 834
Accuracy	0.7526		0.7486	
Macro-F1	0.7136		0.6768	

BETO. Overall, BETO’s Macro-F1 (0.7136) is higher than BETO-Emotion’s (0.6768).

5.4. Ablation Studies

To measure the impact of data balancing and emoji normalization on validation performance, we conduct three configurations with BETO-Emotion:

Full Pipeline (oversampling + emoji normalization): Macro-F1 = 0.6768.
Without Oversampling (weighted loss only): Macro-F1 = 0.6932 (−0.0164).
Without Emoji Normalization (oversampling only): Macro-F1 = 0.7005 (−0.0237).

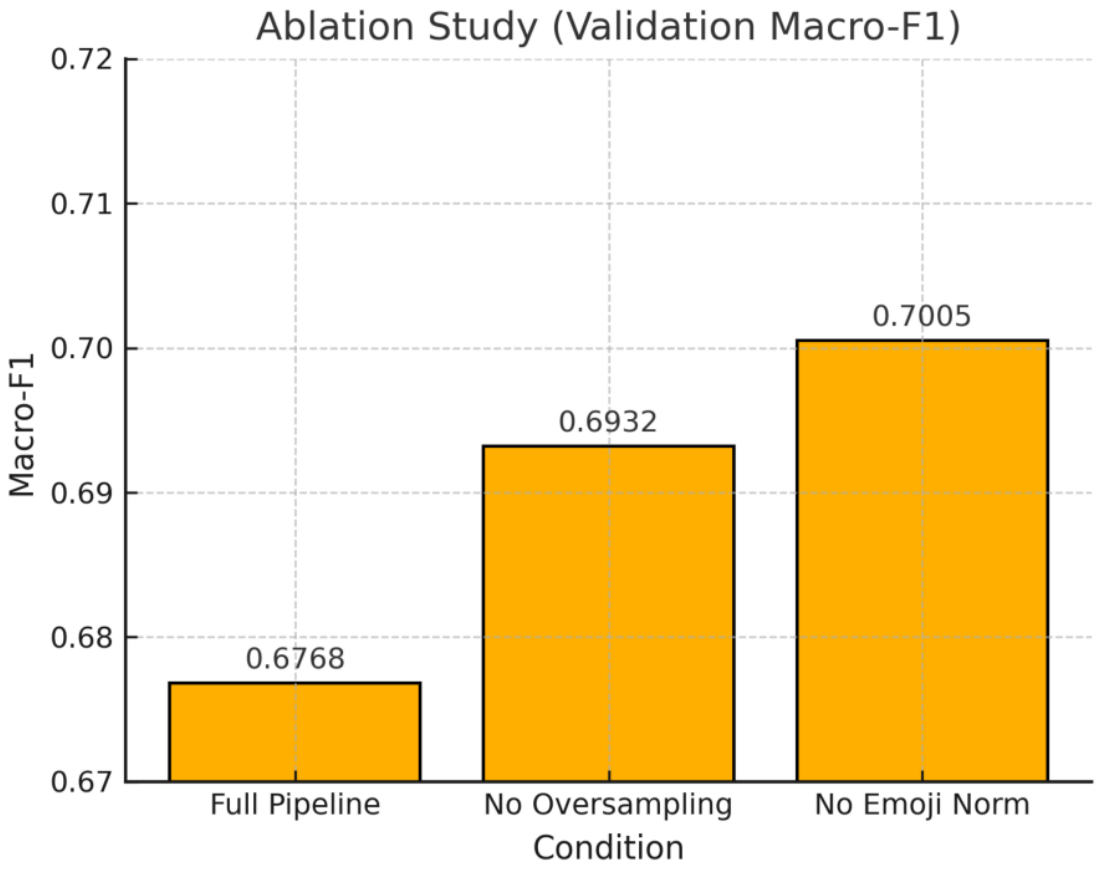

Figure 1: Validation Macro-F1 for ablation experiments (BETO-Emotion).

5.5. Error Analysis

Figure 2 shows the confusion matrix for BETO-Emotion on Dev. The most frequent confusions occur between "Regular" (2) and "Malo" (1), reflecting challenges in intermediate sentiment detection.

Table 5 provides representative Dev error cases: These errors stem from residual orthographic noise (missing diacritics), token-splitting artifacts, and ambiguous intermediate sentiment expressions.

5.6. Test Performance (Codabench)

We evaluate our final models on the official REST-Mex 2025 test set ($N_{\text{test}} = 89{,}166$) using the Codabench leaderboard. Table 6 reports Accuracy, Macro-F1 (Polarity), and the Rank (Macro-F1) for each submission. SINAI-UGPLN's best BETO-Emotion run (UGPLN_0) and BETO run (UGPLN_2) are listed below: Neither run achieved a place in the official ranking ("HM" indicates Honorable Mention). These results confirm that, despite strong Dev performance, both BETO and BETO-Emotion struggled to generalize to the noisy test set under the official Codabench evaluation.

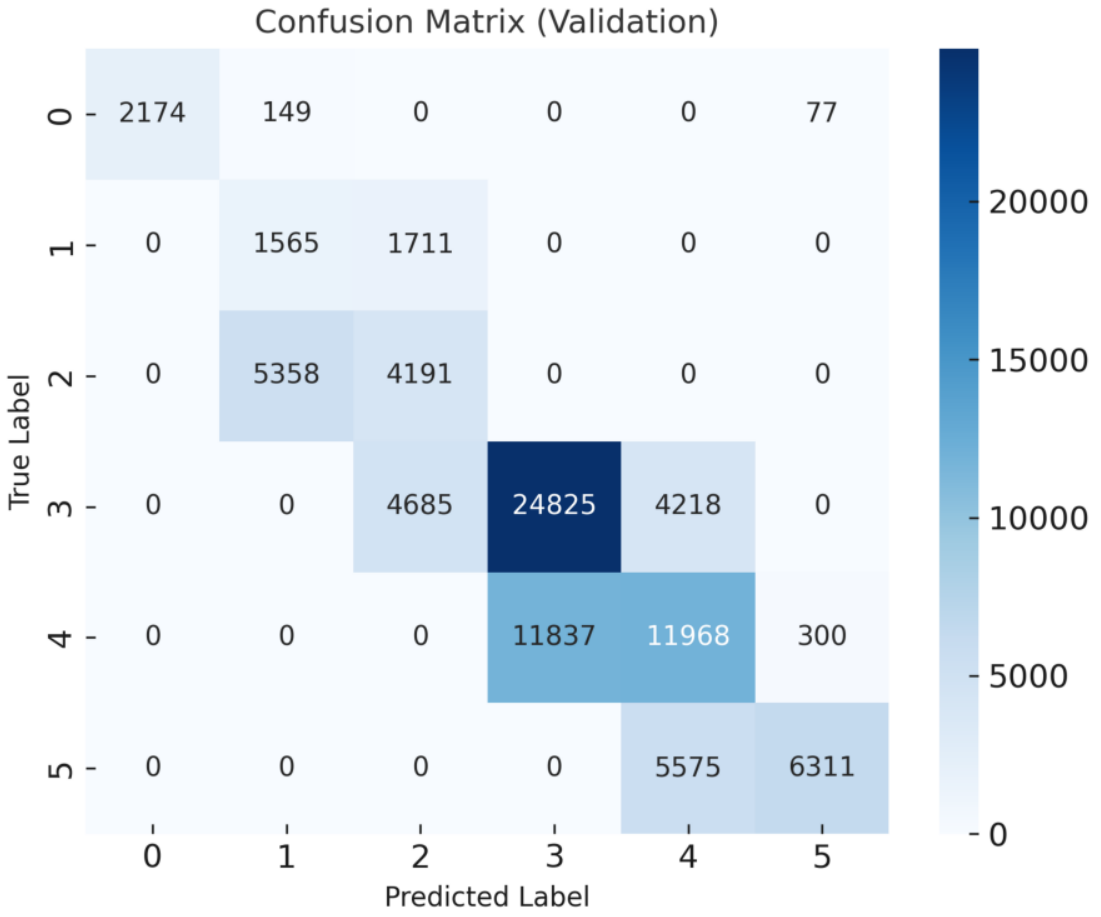

Figure 2: Confusion Matrix on Validation (BETO-Emotion).

Table 5
Representative error cases on Dev (BETO-Emotion).

True	Predicted
"La película estuvo pésima 😢"	"La película estuvo pesima" (0 → 1: dropped accent; cry emoji present)
"Comida regular, nada especial"	"Comida re gular, nada especial" (2 → 1: unintended space split token)
"Me encantó, muy bueno"	"Me encanto muy bn" (4 → 3: missing accent; "bn" abbreviation)

6. Discussion

The Dev results and Codabench rankings reveal several insights about our fine-tuning strategies and preprocessing pipeline. First, while BETO achieved a Dev accuracy of 75.26% and Macro-F1 of 0.7136, BETO-Emotion attained a slightly lower Dev accuracy (74.86%) and Macro-F1 (0.6768). This indicates that the emotion-specialized pretraining benefited class-specific detection of moderate negativity (class 1 "Malo")—BETO-Emotion's F1 for class 1 (0.5302) exceeded BETO's (0.5042)—but at the expense of overall Macro-F1, as BETO maintained stronger performance on extreme sentiment classes.

Our ablation studies further demonstrate the delicate trade-offs in data handling. Removing oversampling (i.e., using only class weights in the loss) increased Dev Macro-F1 from 0.6768 to 0.6932, suggesting that oversampling introduced redundancy or noise that degraded validation performance. Conversely, omitting emoji normalization (while retaining oversampled data) increased Dev Macro-F1 to 0.7005,

Table 6
Codabench Test results for UGPLN (REST-Mex 2025).

Model	Accuracy (%)	Macro-F1	Rank (Macro-F1)
BETO	16.28	0.1027	HM
BETO-Emotion	39.81	0.1915	HM

suggesting that converting emojis to textual tokens may have inadvertently altered the underlying sentiment cues. In both cases, the full pipeline (oversampling plus emoji normalization) performed worse than either single-factor ablation, indicating an interaction effect where combining both strategies did not yield additive gains.

The error analysis on Dev (Figure 2) shows persistent confusions between "Regular" (2) and "Malo" (1). Many "Malo" examples lacked clear negative markers or contained mixed sentiment, causing the model to favor class 2. Orthographic noise—missing diacritics and token splits—also contributed to misclassification (e.g., "pésima" → "pesima" and "comida re gular"). These errors underscore the challenge of intermediate sentiment detection in dialectal Spanish.

On the official Codabench test set, both models suffered a substantial performance drop. BETO-Emotion achieved only 39.81% accuracy and Macro-F1 of 0.1915 (Honorable Mention), while BETO reached 16.28% accuracy and Macro-F1 of 0.1027 (Honorable Mention). This sharp decline from Dev performance highlights a significant generalization gap. Possible causes include:

Data distribution shift: The test set likely contains dialectal variants or noise patterns not well represented in the Dev split, causing erroneous predictions under out-of-distribution conditions.
Over-reliance on surface cues: Both models may have learned spurious correlations (e.g., certain misspellings or emoji patterns) that did not transfer to the unseen test examples.
Insufficient dialect coverage: Although we balanced across six major dialects, some rare or extreme dialectal forms in the test set may not have been adequately captured by our synthetic or oversampled data.

To close this gap, future work should explore:

1. **Adversarial data augmentation**: Automatically generate dialectal variants that mimic test-time noise, using larger generative models (e.g., Mistral-7B-Instruct) to expand the synthetic pool.
2. **Dialect-specific embeddings**: Incorporate learned dialect embeddings or adapters to help the model distinguish orthographic patterns unique to each region.
3. **Robust tokenization**: Employ subword vocabularies that better capture accent and diacritic variations, or use byte-level encoders to minimize the impact of missing accents.
4. **Curriculum learning**: Start training on cleaned, high-quality examples, then gradually introduce more noisy and dialectal inputs to improve generalization.

In summary, although BETO-Emotion and our preprocessing pipeline delivered competitive Dev results, the low test performance highlights the difficulty of real-world dialectal sentiment analysis. Addressing distributional shifts and refining tokenization strategies will be crucial for closing the gap between validation and test performance in future iterations.

7. Conclusions and Future Work

In this work, we presented SINAI-UGPLN's fine-tuning strategies for the REST-Mex 2025 Sentiment Analysis task. Our multilingual preprocessing pipeline—including Unicode normalization, emoji conversion, and orthographic cleaning—combined with class-weighted loss and oversampling, yielded strong Dev performance: BETO achieved 75.26 % accuracy and 0.7136 Macro-F1, while BETO-Emotion reached 74.86 % accuracy and 0.6768 Macro-F1. Ablation studies revealed that neither oversampling

nor emoji normalization alone consistently improved results when combined, highlighting complex interactions between data balancing and tokenization. Error analysis identified persistent confusions between intermediate classes ("Malo" vs. "Regular") due to residual orthographic noise. On the official Codabench test set, both BETO-Emotion (39.81 % accuracy, 0.1915 Macro-F1) and BETO (16.28 % accuracy, 0.1027 Macro-F1) fell short of Dev performance, illustrating a significant generalization gap under dialectal noise.

Future work will focus on closing this gap by (1) generating adversarial dialectal variants with large generative models to simulate test-time noise better; (2) integrating dialect-specific embeddings or adapter modules to capture region-specific orthographic patterns; (3) adopting byte-level or subword tokenization schemes that preserve accent and diacritic information; and (4) applying curriculum learning to introduce noise during training gradually. We also plan to explore active learning approaches to identify underrepresented dialectal forms in the test distribution and to incorporate external lexicons of regional slang. By refining tokenization and augmenting data with realistic dialectal noise, we aim to improve robustness and narrow the gap between validation and test performance in future REST-Mex iterations.

Acknowledgments

This work is funded by the Ministerio para la Transformación Digital y de la Función Pública and Plan de Recuperación, Transformación y Resiliencia – Funded by EU – NextGenerationEU within the framework of the project Desarrollo Modelos ALIA. This work has also been partially supported by Project CONSENSO (PID2021-122263OB-C21), Project MODERATES (TED2021-130145B-I00) and Project SocialTox (PDC2022-133146-C21) funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. Moreover, this research is part of the proposal presented at the Call for Research Project Proposals of the Internal Competitive Fund (FCI) 2023, which was approved on September 14, 2023 (Resolution No. R-CSU-UG-SE34-313-14-09-2023) by the Consejo Superior Universitario of the Universidad de Guayaquil.

The authors declare that they have contributed equally and share authorship roles for this publication.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 and Grammarly to check grammar and spelling. After using these tools and services, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] M. A. Álvarez-Carmona, R. Aranda, A. Y. Rodríguez-Gonzalez, D. Fajardo-Delgado, M. G. Sánchez, H. Pérez-Espinosa, J. Martínez-Miranda, R. Guerrero-Rodríguez, L. Bustio-Martínez, Ángel Díaz-Pacheco, Natural language processing applied to tourism research: A systematic review and future research directions, Journal of King Saud University - Computer and Information Sciences 34 (2022) 10125–10144. URL: https://www.sciencedirect.com/science/article/pii/S1319157822003615. doi:https://doi.org/10.1016/j.jksuci.2022.10.010.

[2] R. Guerrero-Rodríguez, M. A. Álvarez-Carmona, R. Aranda, et al., Big data analytics of online news to explore destination image using a comprehensive deep-learning approach: a case from mexico, Information Technology & Tourism 26 (2024) 147–182. URL: https://doi.org/10.1007/s40558-023-00278-5. doi:10.1007/s40558-023-00278-5.

[3] J. Cañette, G. Chacón, R. Fuentes, A. Chishti, M. Gutiérrez, G. Pablo, Spanish pretrained bert model and evaluation data, in: Proceedings of the Thirteenth Language Resources and Evaluation

Conference (LREC'20), Workshop on Language Resources and Evaluation for NLP, European Language Resources Association (ELRA), Marseille, France, 2020, pp. 5230–5239.

[4] J. M. Pérez, D. A. Furman, L. A. Alemany, F. Luque, R. Cordeiro, Robertuito: A roberta-based model for social media text in spanish, arXiv preprint arXiv:2111.09453 (2021). URL: https://arxiv.org/abs/2111.09453.

[5] R. Guerrero-Rodriguez, M. A. Álvarez Carmona, R. Aranda, A. P. López-Monroy, Studying online travel reviews related to tourist attractions using nlp methods: the case of guanajuato, mexico, Current Issues in Tourism 26 (2023) 289–304. URL: https://doi.org/10.1080/13683500.2021.2007227. doi:10.1080/13683500.2021.2007227. arXiv:https://doi.org/10.1080/13683500.2021.2007227.

[6] M. Á. Álvarez-Carmona, R. Aranda, S. Arce-Cárdenas, D. Fajardo-Delgado, R. Guerrero-Rodríguez, A. P. López-Monroy, J. Martínez-Miranda, H. Pérez-Espinosa, A. Rodríguez-González, Overview of rest-mex at iberlef 2021: Recommendation system for text mexican tourism, Procesamiento del Lenguaje Natural 67 (2021). doi:https://doi.org/10.26342/2021-67-14.

[7] M. Á. Álvarez-Carmona, Á. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, D. Fajardo-Delgado, R. Guerrero-Rodríguez, L. Bustio-Martínez, Overview of rest-mex at iberlef 2022: Recommendation system, sentiment analysis and covid semaphore prediction for mexican tourist texts, Procesamiento del Lenguaje Natural 69 (2022) 289–299.

[8] M. Á. Álvarez-Carmona, R. Aranda, R. Guerrero-Rodríguez, A. Y. Rodríguez-González, A. P. López-Monroy, A combination of sentiment analysis systems for the study of online travel reviews: Many heads are better than one, Computación y Sistemas 26 (2022) 977–987.

[9] M. A. Álvarez-Carmona, A. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, V. Muńiz-Sánchez, A. Pastor López-Monroy, F. Sánchez-Vega, L. Bustio-Martínez, Overview of rest-mex at iberlef 2023: Research on sentiment analysis task for mexican tourist texts, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023), co-located with the 39th Conference of the Spanish Society for Natural Language Processing (SEPLN 2023), CEUR-WS.org, 2023, pp. 425–436.

[10] M. Fernández, J. López, E. Ruiz, Dialectal variations in spanish sentiment corpora: Challenges and benchmarks, in: Proceedings of IberLEF 2020, volume 2798 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 89–102.

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

[12] M. Á. Álvarez-Carmona, Á. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, L. Bustio-Martínez, V. Herrera-Semenets, Overview of rest-mex at iberlef 2025: Researching sentiment evaluation in text for mexican magical towns, volume 75, 2025.

[13] J. De la Rosa, E. G. Ponferrada, M. Romero, P. Villegas, P. González de Prado Salas, M. Grandury, Bertin: Efficient pre-training of a spanish language model using perplexity sampling, Procesamiento del Lenguaje Natural 68 (2022) 13–23. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/issue/view/285.