Migueluc3m at SpeechMATICS: Speech Multimodal Audio-Text Irony Classification in Spanish

Miguel Sánchez Sánchez

Universidad Carlos III de Madrid, Leganés, 28911, Madrid, Spain

Abstract

This paper describes our system for *SpeechMATICS*, a shared task on irony detection in Spanish at IberLEF 2026. The competition comprises two tasks: Task 1, which addresses irony detection from text only, and Task 2, which introduces a multimodal setting combining text and audio. For both tasks, we adopt transformer-based models, which have become the dominant approach for text and speech classification. For Task 1, we fine-tune and compare three Spanish-oriented text models: BETO, BERTIN, and MrBERT-es, selecting BETO as the best performing model. For Task 2, we evaluate three audio transformer models: Wav2Vec2, HuBERT, and WavLM. Then, we explore three multimodal fusion strategies: a weighted ensemble, a Support Vector Machine (SVM) and a Multilayer Perceptron (MLP). The best performing configuration is a BETO+WavLM fusion using an SVM trained on text logits and audio embeddings, achieving a macro-averaged F1 score of 0.708 on the official test set. Our system achieved first place in Task 2 and second place in Task 1 on the official leaderboard. The results suggest that combining text and audio is beneficial for irony detection in Spanish, with all fusion strategies outperforming the text-only baseline.

Keywords

Irony, SVM, Transformer, Ensemble

1. Introduction

Irony is a form of figurative language in which the intended meaning of an utterance differs from its literal interpretation. Detecting irony automatically is a well-known challenge in Natural Language Processing (NLP), as it requires understanding pragmatic and contextual cues that go beyond the surface meaning of words [1]. This makes irony particularly problematic for downstream tasks such as sentiment analysis and opinion mining where misclassifying an ironic statement can lead to incorrect conclusions about the expressed opinion.

In this paper, we describe our participation for SpeechMATICS [2], a shared task organized within IberLEF [3], focused on irony detection in Spanish. The competition defines two tasks. Task 1 focuses on text-only irony classification, where systems must determine whether a given written transcription is ironic or non-ironic. Task 2 introduces a multimodal setting, where both the text transcription and its corresponding audio recording are available. In this task, systems must classify the audio-text pair as ironic or non-ironic. Our system ranked first in Task 2 (multimodal irony detection) and second in Task 1 (text-only irony detection), achieving the best overall performance in the competition based on the official evaluation criteria.

For Task 1, we fine-tune and compare three Spanish-based transformer models: BETO, BERTIN and MrBERT, selecting the best performing one for submission. For Task 2, we additionally evaluate three audio transformer models: Wav2Vec2, HuBERT, and WavLM and test three fusion strategies to combine the results of both text and audio models: a weighted ensemble, a Support Vector Machine (SVM), and a Multilayer Perceptron (MLP). All systems are evaluated using macro-averaged F1 score.

The rest of the paper is structured as follows. Section 2 reviews related work. Section 3 describes the dataset. Section 4 details our methodology. Section 5 covers the experimental setup. Section 6 and 7 reports results. Section 8 discusses findings and limitations. Section 9 concludes the paper, and Section 10 outlines future work.

IberLEF 2026, September 2026, León, Spain

100429105@alumnos.uc3m.es (M. S. Sánchez)

2. Related Work

Early work on irony detection includes the SemEval-2018 Task 3 proposed by Van Hee et al. (2018) [4], which focused on the classification of English tweets as ironic or non-ironic. Subsequent studies have extended irony detection to other languages, including Spanish and Arabic [5, 6].

Most previous approaches have primarily relied on textual information, despite the fact that irony is often conveyed through additional acoustic and prosodic cues in spoken language. In the multimodal setting, Malik et al. (2023) [7] investigate irony detection in social media posts combining text and images, and evaluate whether visual information provides complementary signals for the task. Their findings suggest that text-only transformer models can perform competitively, raising questions about the actual benefit of additional modalities in some multimodal settings.

In contrast, Barceló Milkova (2025) [8] explores irony detection in Spanish speech and text, proposing a system that integrates textual and acoustic representations. The study introduces a dedicated corpus and evaluates different strategies for fusing speech and text-based models, highlighting the potential of acoustic cues for spoken irony detection in Spanish.

3. Dataset analysis

The competition dataset, MESironic [9], consists of 5,100 Spanish audio-text pairs with a near-balanced class distribution of approximately 50.7% ironic and 49.3% non-ironic samples.

The transcriptions are relatively short, with a mean of approximately 12 words per sample and a maximum of 109 words. After tokenization, the mean number of tokens is 16 and the median is 13. Although the longest sample contains 161 tokens, 90% of the samples contain fewer than 30 tokens, indicating that the dataset consists predominantly of short utterances. Based on this analysis, a maximum sequence length of 48 tokens was selected, which is sufficient to cover the vast majority of samples without significant truncation. A slight difference in mean length is observed between classes, with non-ironic samples being slightly longer (13 vs. 10 words on average).

The total duration of the audio dataset is approximately 296 minutes. Individual recordings are generally short, with a median duration of 2.70 seconds and a mean of 3.52 seconds. The 90th and 95th percentiles are 6.79 and 8.72 seconds respectively, indicating that the majority of recordings are brief utterances, consistent with the short nature of the transcriptions. A slight difference in duration is observed between classes, with non-ironic samples being marginally longer on average (3.79s vs. 3.20s), mirroring the pattern observed in the text analysis.

No relevant noise, errors, or inconsistencies were detected in the text, as the dataset had been previously reviewed manually by the dataset authors.

3.1. On the Use of Data Augmentation

In small datasets such as this one, data augmentation is a commonly used technique to increase the amount of training data and improve model generalization [10]. However, given the highly context-dependent nature of irony and the fact that meaning often relies on the precise wording and tone, its application was considered problematic. Common text augmentation approaches such as synonym replacement or back-translation risk altering the meaning of ironic statements, potentially introducing noise or even flipping the label of an augmented sample. Additionally, standard audio augmentation techniques may alter the prosodic and tonal cues that are essential to the expression of spoken irony. For these reasons, data augmentation was discarded in favor of working directly with the original dataset.

4. Methodology

4.1. Evaluation and fine-tuning of models

For both tasks, an initial set of three candidate models was selected for text and audio respectively. Each model was fine-tuned and evaluated locally using a 80/20 train/validation split, with macro-averaged F1 as the evaluation metric. Due to hardware and time constraints, all models were fine-tuned using the same hyperparameter configuration, varying only the learning rate between 1e-5 and 2e-5. The best performing model in each case was selected for the subsequent fusion experiments.

Hyperparameter	Text Models	Audio Models
Learning rate	1e-5, 2e-5	1e-5, 2e-5
Epochs	5	3
Batch size	32	8
Warmup ratio	0.1	0.1
Weight decay	0.01	0.01
Max grad. norm	-	1.0

Table 1
Hyperparameter configuration used for fine-tuning text and audio models.

4.1.1. Text models

For Task 1, we evaluate three BERT-based transformer models trained on Spanish text: BETO [11], BERTIN [12] and MrBERT-es [13], which we briefly describe below.

1. **BETO**: BERT-based language model trained exclusively on Spanish text, including the Spanish Wikipedia and all Spanish-language sources from the OPUS Project [14]. With 110M parameters, it is one of the earliest and most widely adopted transformer models for Spanish NLP. BETO is available in both cased and uncased versions; given the nature of irony, where capitalisation may carry pragmatic significance, we opted for the cased version.
2. **BERTIN**: RoBERTa-based language model trained from scratch on Spanish text, specifically on a sampled subset of the Spanish portion of the mC4 dataset. A key contribution of BERTIN is the introduction of perplexity sampling, a technique that enables efficient pre-training using approximately half the number of steps and one fifth of the data compared to standard approaches, while achieving comparable performance to existing Spanish language models.
3. **MrBERT-es**: Bilingual Spanish-English encoder derived from MrBERT, a family of ModernBERT-based models pre-trained on 35 languages. MrBERT-es is obtained from the multilingual base model through vocabulary adaptation, reducing the model size to 150M parameters while specializing it for Spanish and English. At the time of this work, it represents one of the most recently released Spanish language models.

4.1.2. Audio models

For Task 2, we evaluate three transformer-based audio models pre-trained on English speech using self-supervised learning: Wav2Vec2 [15], HuBERT [16], and WavLM [17]. Although multilingual variants exist for each of these models, the English base versions were selected for three reasons: hardware constraints, the small size of the dataset, and to ensure a fair comparison across models.

1. **Wav2Vec2**: Learns speech representations directly from raw audio through a combination of a convolutional neural network and a Transformer, and is fine-tuned on labeled data for downstream tasks.
2. **HuBERT**: Addresses the challenge of learning from continuous audio by using an offline clustering step to generate aligned target labels for a BERT-like masked prediction loss. A key

characteristic of the approach is that the prediction loss is applied over masked regions only, forcing the model to learn both acoustic and language representations.

3. **WavLM**: Extends this paradigm by jointly learning masked speech prediction and speech denoising during pre-training, allowing it to capture both speech content and non-speech information such as speaker identity and paralinguistics. It also employs gated relative position bias in its Transformer structure to better capture the ordering of input speech.

4.2. Fusion of text and audio models

To combine the text and audio models for Task 2, three fusion strategies were explored: a weighted ensemble, a Support Vector Machine (SVM), and a Multilayer Perceptron (MLP). Each strategy was applied to the best performing text and audio models, as determined by the local evaluation described in Section 6. The following subsections describe each approach in detail.

4.2.1. Weighted ensemble

The weighted ensemble combines the class probabilities produced by the text and audio models through a weighted sum, defined as:

$$p_{\text{combined}} = w_{\text{text}} \cdot p_{\text{text}} + (1 - w_{\text{text}}) \cdot p_{\text{audio}} \tag{1}$$

where p_{text} and p_{audio} are the softmax probability vectors output by the text and audio models respectively, and $w_{\text{text}} \in [0, 1]$ is the weight assigned to the text model. The final prediction is the class with the highest combined probability.

To study the relative contribution of each modality, three fixed weights were evaluated: 0.25, 0.5 and 0.75. A weight of 0.25 favours the audio model, 0.75 favours the text model, and 0.5 gives equal contribution to both. Additionally, an attempt was made to estimate the optimal weight through local evaluation on the validation split.

4.2.2. SVM

The SVM fusion approach uses the output of the text and audio models as input features to train a Support Vector Machine classifier. Several input feature combinations were explored: text logits concatenated with audio embeddings, text logits concatenated with audio logits, and text embeddings concatenated with audio embeddings. In all cases, features are normalized using standard scaling prior to training, and an RBF kernel is used.

Optimal hyperparameters were selected through a grid search with 5-fold cross-validation on the training split, optimizing for macro-averaged F1. The search covered the following parameter ranges: $C \in \{1, 10, 100, 1000\}$ and $\gamma \in \{\text{scale}, \text{auto}, 0.01, 0.001\}$. The best performing configuration was then evaluated on the held-out validation split to obtain an unbiased estimate of performance. The input and hyperparameter combination that yielded the best results is reported in Section 7.2.

4.2.3. MLP

The MLP fusion approach trains a multilayer perceptron on the concatenated outputs of the text and audio models. Several input feature combinations were explored: text CLS token embeddings concatenated with audio embeddings, text logits concatenated with audio logits, and text logits concatenated with audio embeddings. In all cases, features are normalized using standard scaling prior to training.

Due to hardware and time constraints, most hyperparameters were fixed: AdamW optimizer with a learning rate of 1e-5, weight decay of 0.01, dropout of 0.4, batch size of 32, and 30 training epochs. The hidden dimension was the only hyperparameter varied, with values of 128, 256, and 512 evaluated. The model achieving the highest macro-averaged F1 on the validation split was selected as the best performing configuration. The input combination that yielded the best results is reported in Section 7.3.

5. Experimental setup

All experiments were conducted on a single NVIDIA RTX 3050 GPU with 8GB of VRAM. A fixed random seed of 73918452 was used across all experiments to ensure reproducibility. All transformer models were loaded and fine-tuned using the HuggingFace Transformers library [18], with PyTorch [19] as the underlying deep learning framework. Audio loading and resampling were handled using librosa [20], and SVM training and hyperparameter search were performed using scikit-learn [21]. Data manipulation was handled with NumPy [22] and pandas [23].

6. Local Model Evaluation

6.1. Text Model Evaluation

Table 2 shows the results of the local evaluation of the three text models under the fine-tuning configuration described in Section 6. BETO achieves the highest macro-averaged F1 score across both learning rates, with a learning rate of 1e-5 yielding the best result (F1 = 0.6822). BERTIN consistently underperforms the other two models. MrBERT-es obtains competitive results but performs worse than BETO, despite being the most recently released model Based on these results, BETO with a learning rate of 1e-5 was selected for all subsequent experiments.

Model	LR	Precision	Recall	F1
BETO	1e-5	0.6826	0.6823	0.6822
	2e-5	0.6710	0.6707	0.6705
BERTIN	1e-5	0.6178	0.6176	0.6174
	2e-5	0.6515	0.6493	0.6478
MrBERT-es	1e-5	0.6529	0.6529	0.6529
	2e-5	0.6533	0.6528	0.6526

Table 2
Text model comparison results on the local evaluation set.

6.2. Audio Model Evaluation

Table 3 shows the results of the local evaluation of the three audio models under the fine-tuning configuration described in Section 6. All three models perform best with a learning rate of 2e-5. HuBERT achieves the highest macro-averaged F1 score (F1 = 0.6588), followed closely by WavLM (F1 = 0.6526) and Wav2Vec2 (F1 = 0.6474). While the differences between all three models are small, HuBERT and WavLM were selected for the fusion experiments and Wav2Vec2 was discarded. This decision was motivated by Wav2Vec2 being the worst performing model across both learning rates, as well as time and hardware constraints that limited the number of fusion configurations that could be explored.

Model	LR	Precision	Recall	F1
Wav2Vec2	1e-5	0.6249	0.6208	0.6162
	2e-5	0.6582	0.6524	0.6474
HuBERT	1e-5	0.6546	0.6517	0.6488
	2e-5	0.6592	0.6592	0.6588
WavLM	1e-5	0.6442	0.6416	0.6408
	2e-5	0.6527	0.6526	0.6526

Table 3
Audio model comparison results on the local evaluation set.

7. Competition results

7.1. Weighted ensemble

For both audio models, the best results were obtained with equal weighting $w_{\text{BETO=0.5}}$, suggesting that text and audio contribute similarly to the final prediction. An optimal weight was estimated through local evaluation, yielding $w_{\text{BETO=0.65}}$ for BETO+HuBERT and $w_{\text{BETO=0.55}}$ for BETO+WavLM. However, both locally optimized weights performed worse on the official Codabench test set than the equal weighting, and thus are not presented in the table.

Text Model	Audio Model	Weights	Precision	Recall	F1
BETO	HuBERT	w_{BETO} = 0.25	0.6508	0.6487	0.6468
		w_{BETO} = 0.50	0.6854	0.6848	0.6842
		w_{BETO} = 0.75	0.6690	0.6690	0.6688
BETO	WavLM	w_{BETO} = 0.25	0.6757	0.6751	0.6751
		w_{BETO} = 0.50	0.6966	0.6966	0.6966
		w_{BETO} = 0.75	0.6810	0.6811	0.6810

Table 4
Weighted ensemble fusion results on Codabench (Task 2).

7.2. SVM

Out of all tested input combinations, using the text model logits and audio model embeddings yielded the best results for both BETO+HuBERT and BETO+WavLM, and is therefore the configuration reported in Table 5. The optimal hyperparameters found through GridSearch are also included. In both cases, an RBF kernel with C=1 was selected as the best configuration. BETO+WavLM outperforms BETO+HuBERT, achieving an F1 score of 0.7016 compared to 0.6974, making it the best performing fusion configuration observed across all three strategies.

Text Model	Audio Model	Hyperparameters	Precision	Recall	F1
BETO	WavLM	C=1, γ=scale, kernel=rbf	0.7026	0.7016	0.7016
BETO	HuBERT	C=1, γ=0.001, kernel=rbf	0.6978	0.6974	0.6974

Table 5
SVM fusion results on Codabench (Task 2).

7.3. MLP

Out of all tested input combinations, concatenating the text and audio model embeddings yielded the best results for both BETO+HuBERT and BETO+WavLM, and is therefore the configuration reported in Table 6. Among the neuron sizes evaluated, 256 neurons achieved the best performance for both audio model combinations. As with the previous fusion strategies, BETO+WavLM outperforms BETO+HuBERT, achieving an F1 score of 0.6875 compared to 0.6854. However, the MLP underperforms both the weighted ensemble and the SVM across all configurations, making it the weakest of the three fusion strategies evaluated.

Text Model	Audio Model	Hyperparameters	Precision	Recall	F1
BETO 1e-5	HuBERT 2e-5	256 neurons, wd=0.01, dropout=0.4, lr=1e-5	0.6881	0.6858	0.6854
BETO 1e-5	WavLM 2e-5	256 neurons, wd=0.01, dropout=0.4, lr=1e-5	0.6938	0.6888	0.6875

Table 6
MLP fusion results on Codabench (Task 2).

7.4. Final competition results

For task 1, BETO was used with the hyperparameter configuration described in section 6. For task 2, an SVM combining BETO's logits and WavLM's embeddings was used, as it performed the best out of all fusion strategies evaluated. Additionally, the SVM was retrained with 100% of the data, using the hyperparameter configuration obtained in section 7.2.

Rank	Participant	Precision	Recall	F1
1	carlosdf	0.660	0.659	0.659
2	migueluc3m	0.658	0.658	0.658
3	UC-UCO-Plenitas	0.654	0.646	0.641
4	HammerTeam	0.623	0.623	0.622
-	Baseline	0.602	0.602	0.600

Table 7
Task 1 official leaderboard results on Codabench.

Rank	Participant	Precision	Recall	F1
1	migueluc3m	0.710	0.709	0.708
2	carlosdf	0.697	0.697	0.697
3	UC-UCO-Plenitas	0.669	0.669	0.668
4	HammerTeam	0.662	0.649	0.641
-	Baseline	0.610	0.609	0.609

Table 8
Task 2 official leaderboard results on Codabench.

Tables 7 and 8 show the official leaderboard results for Task 1 and Task 2 respectively. For Task 1, our system ranked second with an F1 score of 0.658, close to the first place result of 0.659. For Task 2, our system achieved first place with an F1 score of 0.708, outperforming the second place system by a margin of 0.011. The larger margin observed in Task 2 suggests that the multimodal approach provided a more decisive advantage over competing systems than the text-only approach.

8. Discussion and limitations

The weighted ensemble results suggest that text and audio contribute similarly to irony detection, as equal weighting ($w_{\text{BETO}} = 0.5$) consistently outperformed configurations that favoured either modality. This is an indication that both modalities carry complementary information for this task, and that neither dominates the other.

Among the fusion strategies, the SVM achieved the best results, outperforming both the ensemble and the MLP. For the SVM, the combination of text logits and audio embeddings consistently outperformed other input combinations. One possible explanation is that the text logits, being a compact 2-dimensional representation of the model's final decision, provide a clean and direct classification signal, while the richer 768-dimensional audio embeddings allow the SVM to exploit finer-grained acoustic information. However, this remains a hypothesis and further analysis would be needed to confirm it.

The MLP consistently underperformed both the ensemble and the SVM across all configurations. This could be attributed to the small size of the dataset, which may be insufficient for training a neural network from scratch, as well as the lack of extensive hyperparameter tuning due to the previously mentioned hardware and time constraints.

In all fusion experiments, BETO+WavLM outperformed BETO+HuBERT, suggesting that WavLM's joint training on masked speech prediction and denoising may provide more useful representations for irony detection than HuBERT's clustering-based approach. However, given the small dataset and limited tuning, this conclusion should be treated with caution.

Additionally, several limitations of the present work should be acknowledged. First, all text and audio models were fine-tuned using the same hyperparameter configuration, varying only the learning rate.

This lack of model-specific hyperparameter tuning means that the relative performance of the evaluated models may not reflect their true potential: a different model could potentially have outperformed BETO or WavLM under a more exhaustive search.

Second, the audio models used are base models pre-trained exclusively on English speech, which may limit their ability to capture Spanish-specific prosodic and acoustic cues relevant to irony. Third, the MLP architecture was not extensively tuned due to hardware and time constraints, and a more systematic search could yield better results.

Finally, the small size of the dataset represents a general limitation for all approaches, as it restricts the ability to train and evaluate models.

9. Conclusion

In this paper, we described our system for *SpeechMATICS*, a shared task on irony detection in Spanish at IberLEF 2026. We evaluated transformer-based models for both text and audio modalities, and explored three multimodal fusion strategies to combine them for Task 2.

For Task 1, BETO was selected as the best performing text model, achieving a macro-averaged F1 score of 0.6822 on the local evaluation set. For Task 2, an SVM fusion of BETO and WavLM, trained on text logits and audio embeddings, achieved the best results across all fusion strategies, obtaining a F1 score of 0.708 on the official Codabench test set. Our system achieved first place in Task 2 and second place in Task 1 on the official leaderboard.

The results suggest that combining text and audio is beneficial for irony detection in Spanish, as all fusion strategies outperformed the individual text model baseline. The weighted ensemble results further indicate that both modalities contribute similarly to the final prediction, with equal weighting consistently yielding the best performance. These findings support the hypothesis that acoustic cues carry meaningful information for irony detection that is not captured by text alone.

10. Future Work

Several directions remain open for future exploration. First, while data augmentation was discarded in this work due to the context-dependent nature of irony, future work could investigate augmentation techniques specifically designed to preserve pragmatic meaning in irony datasets.

Second, the hyperparameter search in this work was limited due to hardware and time constraints. A more exhaustive tuning strategy, for example using a dedicated hyperparameter optimization library such as Optuna [24], could potentially yield better results and provide a fairer comparison across models.

Third, the audio models used in this work are base models pre-trained exclusively on English speech. Exploring larger multilingual audio models could improve performance by better capturing Spanish-specific prosodic and acoustic patterns relevant to irony detection.

Finally, the interpretability of the models remains an open question. Applying explainability techniques such as LIME [25] or SHAP [26] could provide insights into which textual and acoustic features are most informative for irony detection in Spanish.

Declaration on Generative AI

Generative AI tools were used to assist in coding and improve the overall flow of the writing. The integrity of this output has been manually verified. All scientific content, including methodology, results, and conclusions, was produced and manually verified by the author.

References

[1] A. Reyes, P. Rosso, On the difficulty of automatically detecting irony: beyond a simple case of negation, Knowledge and Information Systems 40 (2014) 595–614. URL: https://doi.org/10.1007/s10115-013-0652-8. doi:10.1007/s10115-013-0652-8.

[2] V. Ahuir, A. Barceló-Milkova, A. Casamayor-Segarra, M. J. Castro-Bleda, L. F. Hurtado, Overview of SpeechMATICS at IberLEF 2026: Speech Multimodal Audio-Text Irony Classification in Spanish, Procesamiento del Lenguaje Natural 77 (2026). To appear.

[3] 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.

[4] C. Van Hee, E. Lefever, V. Hoste, SemEval-2018 task 3: Irony detection in English tweets, in: M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard, M. Carpuat (Eds.), Proceedings of the 12th International Workshop on Semantic Evaluation, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 39–50. URL: https://aclanthology.org/S18-1005/. doi:10.18653/v1/S18-1005.

[5] P. D. Ibarbia, Automatic Irony Detection in Spanish Tweets, Master's thesis, University of the Basque Country (UPV/EHU), San Sebastián, Spain, 2022. URL: https://addi.ehu.es/bitstream/handle/10810/61825/TFM_Paula_Diez.pdf, master's Thesis.

[6] R. Abdel-Salam, WANLP 2021 shared-task: Towards irony and sentiment detection in Arabic tweets using multi-headed-LSTM-CNN-GRU and MaRBERT, in: N. Habash, H. Bouamor, H. Hajj, W. Magdy, W. Zaghouani, F. Bougares, N. Tomeh, I. Abu Farha, S. Touileb (Eds.), Proceedings of the Sixth Arabic Natural Language Processing Workshop, Association for Computational Linguistics, Kyiv, Ukraine (Virtual), 2021, pp. 306–311. URL: https://aclanthology.org/2021.wanlp-1.37/.

[7] M. Malik, D. Tomás, P. Rosso, How Challenging is Multimodal Irony Detection?, 2023, pp. 18–32. doi:10.1007/978-3-031-35320-8_2.

[8] A. J. B. Milkova, Más allá de las palabras: Detección de ironía en voz y texto con inteligencia artificial, Master's thesis, Universitat Politècnica de València, 2025. URL: https://riunet.upv.es/handle/10251/228652, master's Thesis.

[9] V. Ahuir, A. J. Barceló-Milkova, A. Casamayor-Segarra, V. González Verdegais, L.-F. Hurtado, C. Periñán-Pascual, M. J. Castro-Bleda, MESironic: A Multimodal Dataset and Benchmark for Irony Detection in Spontaneous Spoken Spanish, Procesamiento del Lenguaje Natural 77 (2026). To appear.

[10] J. Wei, K. Zou, Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019. URL: https://arxiv.org/abs/1901.11196. arXiv:1901.11196.

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

[12] J. D. la Rosa y Eduardo G. Ponferrada y Manu Romero y Paulo Villegas y Pablo González de Prado Salas y María 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/article/view/6403.

[13] D. Tamayo, I. Lacunza, P. Rivera-Hidalgo, S. Da Dalt, J. Aula-Blasco, A. Gonzalez-Agirre, M. Villegas, Mrbert: Modern multilingual encoders via vocabulary, domain, and dimensional adaptation, arXiv preprint arXiv:2602.21379 (2026).

[14] J. Tiedemann, Parallel data, tools and interfaces in OPUS, in: N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA), Istanbul, Turkey, 2012, pp. 2214–2218. URL: https://aclanthology.org/L12-1246/.

[15] A. Baevski, H. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020. URL: https://arxiv.org/abs/2006.11477. arXiv:2006.11477.

[16] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, A. Mohamed, Hubert: Self-supervised speech representation learning by masked prediction of hidden units, 2021. URL: https://arxiv.org/abs/2106.07447. arXiv:2106.07447.

[17] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, F. Wei, Wavlm: Large-scale self-supervised pre-training for full stack speech processing, IEEE Journal of Selected Topics in Signal Processing 16 (2022) 1505–1518. URL: http://dx.doi.org/10.1109/JSTSP.2022.3188113. doi:10.1109/jstsp.2022.3188113.

[18] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, 2019. URL: https://arxiv.org/abs/1912.01703. arXiv:1912.01703.

[20] B. McFee, M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen, B. van Niekirk, D. Lee, F. Cwitkowitz, F. Zalkow, O. Nieto, D. Ellis, J. Mason, K. Lee, B. Steers, E. Halvachs, C. Thomé, F. Robert-Stöter, R. Bittner, Z. Wei, A. Weiss, E. Battenberg, K. Choi, R. Yamamoto, C. Carr, A. Metsai, S. Sullivan, P. Friesch, A. Krishnakumar, S. Hidaka, S. Kowalik, F. Keller, D. Mazur, A. Chabot-Leclerc, C. Hawthorne, C. Ramaprasad, M. Keum, J. Gomez, W. Monroe, V. A. Morozov, K. Eliasi, nullmightybofo, P. Biberstein, N. D. Sergin, R. Hennequin, R. Naktinis, beantowel, T. Kim, J. P. Åsen, J. Lim, A. Malins, D. Hereñú, S. van der Struijk, L. Nickel, J. Wu, Z. Wang, T. Gates, M. Vollrath, A. Sarroff, Xiao-Ming, A. Porter, S. Kranzler, Voodoohop, M. D. Gangi, H. Jinoz, C. Guerrero, A. Mazhar, toddrme2178, Z. Baratz, A. Kostin, X. Zhuang, C. T. Lo, P. Campr, E. Semeniuc, M. Biswal, S. Moura, P. Brossier, H. Lee, W. Pimenta, J. P. Åsen, S. Hyun, I. S, E. Rabinovich, G. Lei, J. Guo, P. S. Skelton, M. Pitkin, A. Mishra, S. Chaunin, BenedictSt, S. VanRavenswaay, D. Südholt, librosa/librosa: 0.11.0, 2025. URL: https://doi.org/10.5281/zenodo.15006942. doi:10.5281/zenodo.15006942.

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

[22] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, Array programming with NumPy, Nature 585 (2020) 357–362. URL: https://doi.org/10.1038/s41586-020-2649-2. doi:10.1038/s41586-020-2649-2.

[23] T. pandas development team, pandas-dev/pandas: Pandas, 2020. URL: https://doi.org/10.5281/zenodo.3509134. doi:10.5281/zenodo.3509134.

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, 2019. URL: https://arxiv.org/abs/1907.10902. arXiv:1907.10902.

[25] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, 2016. URL: https://arxiv.org/abs/1602.04938. arXiv:1602.04938.

[26] S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, 2017. URL: https://arxiv.org/abs/1705.07874. arXiv:1705.07874.