INFOTEC-NLP at MiSonGyny 2025: Misogynistic Content Identification in Spanish Song Lyrics Through Splitting and Augmenting Data

Daniela Carmona[1], Mario Graff[1,2,*], Mireya Paredes[1] and Eric Sadit Tellez[1,2]

[1] *INFOTEC Centro Público de Investigación en Tecnologías de la Información y Comunicación, 112 Circuito Tecnopolo Sur, Parque Industrial Tecnopolo 2, Aguascalientes, 20326, México.*

[2] *Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI), 1582 Insurgentes Sur 1582, Crédito Constructor, Ciudad de México, 03940 México*

CEUR-WS.org/Vol-4098/MiSonGyny2025_paper2.pdf

Abstract

Misogyny, as discrimination or prejudice against women, is a global problem present in personal interactions, societal structures, and popular culture. Music lyrics often perpetuate negative stereotypes, objectifying or demeaning women, which influences societal norms and reinforces misogynistic attitudes. This manuscript describes our participation in the MiSongyny shared task at IberLEF2025 [1], centered on the detection of misogynistic content in Spanish song lyrics. Our methodology employs a classifier based on the BETO model (`dccuchile/bert-base-spanish-wwm-uncased`), a Spanish variant of BERT, complemented by diverse lyric processing techniques. For example, we addressed the constraint of transformer models' limited sequence length by enhancing the number of training examples using a chunking strategy with overlapping token windows. Additionally, to alleviate the significant class imbalance in the dataset, we utilized a Paraphrasing Large Language Model (LLM) to augment instances of minority classes. Our system achieved competitive results: an F1-score of **0.7876** for Subtask 1 (binary classification of misogynistic content), and a macro F1-score of **0.4929** for Subtask 2 (multiclass categorization), both surpassing the baseline models provided by the organizers.

Keywords

misogyny identification, large language model, data augmentation

1. Introduction

Misogyny is a deep-seated prejudice against women, rooted in beliefs of their inferiority and linked to patriarchal systems that sustain male dominance. It significantly affects women's mental health by increasing stress, anxiety, and depression, and limiting opportunities [2]. Misogyny is also linked to male violence and extends to online spaces, media, and institutions, appearing as stereotypes, dismissal, mockery, and abuse. Among these spaces, music plays a key role in the normalization of misogynistic discourse, as sexist lyrics pervade popular songs by objectifying women through hypersexualized and commanding expressions. Spanish-speaking genres such as *reggaeton* and *corridos tumbados* are particularly involved, although this issue transcends modern music and appears in various musical styles.

In response to growing concerns over the presence of misogyny in digital content, several computational initiatives have emerged. One notable effort is AMI (Automatic Misogyny Identification), a shared task presented at IberEval [3] and Evalita [4] in 2018, which marked a key starting point in automatic misogyny detection. The challenge involved identifying misogynistic content in tweets, classifying its type (e.g., objectification, dominance), and determining whether it targeted individuals or groups. Participants used models such as Support Vector Machines, Logistic Regression, ensemble

IberLEF 2025, September 2025, Zaragoza, Spain

[*] Corresponding author.

dani.leon.carmona@gmail.com (D. Carmona); mario.graff@infotec.edu.mx (M. Graff); mireya.paredes@gmail.com (M. Paredes); eric.tellez@infotec.edu.mx (E. S. Tellez)

https://mgraffg.github.io/ (M. Graff); https://sadit.github.io/ (E. S. Tellez)

0009-0006-8221-4900 (D. Carmona); 0000-0001-6573-4142 (M. Graff); 0000-0002-7149-4592 (M. Paredes); 0000-0001-5804-9868 (E. S. Tellez)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.10 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-12-10

methods, and basic neural networks trained on features like TF-IDF, character n-grams, and word embeddings [4]. The highest accuracy in the binary classification task reached 0.84 using a TF-IDF + SVD + Boosting approach for Italian tweets, while the best English system combined embeddings and logistic regression to achieve 0.70. These results, while promising, revealed the limitations of traditional models in capturing the nuanced and contextual nature of misogynistic language. This, in turn, motivated the adoption of transformer-based architectures, which are capable of deeper semantic understanding.

In recent years, natural language processing (NLP) has been increasingly applied to the detection of hate speech and discrimination in digital content [5]. While much research has focused on social media, less attention has been paid to content from *song lyrics*, where figurative and poetic language poses unique challenges for automatic detection.

In the context of Spanish NLP, a particularly relevant resource is BETO, the first BERT-based model pre-trained exclusively on Spanish corpora [6]. Developed by researchers at the University of Chile, BETO was trained on a 3 billion-word dataset composed of Wikipedia and OPUS sources such as TED talks and news articles. It follows the BERT-base architecture, but the RoBERTa training recipe [7], i.e., it uses very large batches and removes the next sentence inference task in the training. The authors also introduced GLUES, a benchmark suite for Spanish NLP tasks, including NLI, NER, POS tagging, and QA. When evaluated on these benchmarks, BETO outperformed multilingual BERT, i.e. the original approach to tackle Spanish tasks, in most tasks under the same training conditions, and even achieved state-of-the-art results in POS tagging and document classification. These outcomes support the use of domain-specific and language-specific models like BETO over general-purpose multilingual ones.

Cañete et al. [8] present DistilBETO, a lightweight model based on the full BETO, maintaining similar performance and learning capabilities while requiring significantly fewer computational resources. This model is based on the DistilBERT distillation technique presented by Sanh et al. [9], designed to transfer the knowledge of a large model to a small one, by reducing the number of layers, among other subtle modifications, and retraining with a distillation loss based on cross-entropy.

Building on the intersection of music and hate speech detection, Calderón-Suárez et al. [10] proposed a novel approach to detect misogyny on social networks using transfer learning from song lyrics. They automatically compiled corpora of misogynistic and non-misogynistic content in both English and Spanish, extracting high-quality phrases from lyrics using seed-based heuristics. These phrases were used to augment training datasets for supervised classifiers. A filtering mechanism based on cosine similarity and the Rocchio algorithm ensured the inclusion of the most linguistically relevant patterns. Their experiments showed performance improvements, particularly in Spanish, using models such as BETO and DistilBERT. The method was also extended to the multimodal detection of misogyny in memes, confirming the value of musical content as a source of sociolinguistic features.

A key reference in this line of research is *Task 3 of the shared task HOMO-MEX 2024*, which focused on the detection of homophobic content in Spanish song lyrics [11]. The organizers introduced a new dataset, HOMO-LYRICS, and showed that even state-of-the-art models such as XLM-RoBERTa struggled with this task, which was framed as a binary classification problem. The best system achieved a macro-F1 score of around 0.57. These findings reinforce the idea that the detection of hate speech in musical texts presents unique linguistic and cultural challenges. Addressing this gap, our work focuses on the detection of misogynistic content *in Spanish song lyrics*, which, like homophobia, is often expressed in subtle or normalized ways in widely consumed cultural products.

In this paper, we present our system developed for the MiSongyny shared task at IberLEF 2025 [12]. Our approach is based on the BETO model, a transformer pre-trained exclusively on Spanish, fine-tuned for misogyny detection in song lyrics. To address the challenges posed by input length constraints and class imbalance, we implemented a chunking strategy to segment lyrics into overlapping strophes. Additionally, we applied data augmentation using a Spanish paraphrasing LLM.

This manuscript is organized as follows. Section 2 details our proposed solution for the MiSongyny challenge, while Section 3 presents our experimental results. Section 4 presents our final discussion and conclusions.

2. System description

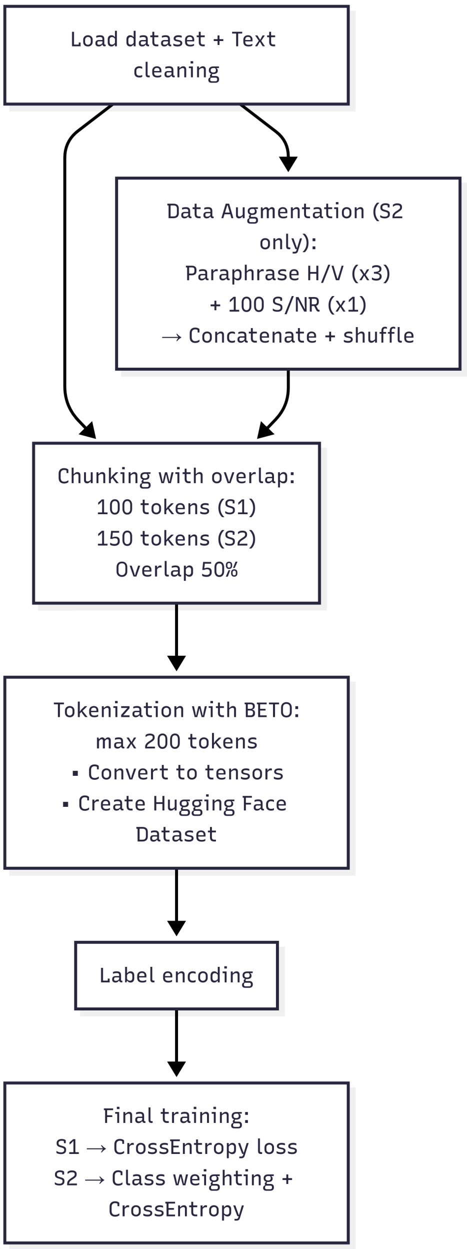

Figure 1: Vertical processing pipeline for both subtasks (S1 and S2). Shared components include dataset loading, text cleaning, chunking with 50% overlap (100 tokens in S1, 150 in S2), tokenization with BETO, and label encoding. Subtask S2 includes additional data augmentation and class weighting before training. The final training step uses standard CrossEntropy loss for S1, and a weighted variant for S2, as detailed in the final block.

The proposed system for detecting misogyny in Spanish song lyrics utilizes a fine-tuned transformer specifically pretrained for Spanish. We employed BETO[1], a monolingual BERT model developed on a comprehensive Spanish text corpus [13], chosen for its effectiveness in Spanish NLP tasks. The system handles the following problems:

- Subtask 1: predict whether a song is misogynistic; a binary classification problem with labels *Not misogynic (NM)* and *Misogynic (M)*.

[1] https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased

- Subtask 2: classify the type of misogynistic language in a song; it considers the following labels: *Sexualization (S)*, *Violence (V)*, *Hate (H)*, or *Not Related (NR)*.

To better understand the dataset composition, Table 1 presents the number of annotated instances per class in the training sets for both subtasks. While Subtask 1 is a binary classification problem, Subtask 2 involves a four-class setup. Both tasks exhibit class imbalance, particularly Subtask 2, where the Hate and Violence categories are underrepresented.

Table 1
Class distribution in the training sets for Subtask 1 and Subtask 2

Class	Subtask 1	Subtask 2
Not Misogynistic (NM)	1,462	–
Misogynistic (M)	642	–
Not Related (NR)	–	526
Sexualization (S)	–	435
Violence (V)	–	129
Hate (H)	–	78
Total	**2,104**	**1,168**

To adapt the model to these tasks, we designed a multistage pipeline that includes data cleaning, overlapping chunk generation for long texts, label encoding, and fine-tuning using the Hugging Face Transformers library with a PyTorch back-end. For Subtask 2 only, we further applied paraphrasing-based data augmentation for minority classes and class rebalancing through computed loss weights. These additional steps aimed to mitigate the effects of class imbalance and improve generalization in the multiclass setting (see Figure 1).

2.1. Data Augmentation via Paraphrasing (Subtask 2 only)

This strategy was applied exclusively to **Subtask 2**, the multiclass classification task involving the labels *Sexualization (S)*, *Violence (V)*, *Hate (H)*, and *Not Related (NR)*. Given the substantial class imbalance, particularly the underrepresentation of the categories *Hate (H)* and *Violence (V)*, we implemented a targeted data augmentation strategy based on paraphrasing. This approach aimed to expand the training set of minority classes while preserving the semantic meaning of the original examples.

We used a Spanish-language paraphrasing model[2] to increase the number of messages in those classes with fewer examples. This model is a sequence-to-sequence model based on T5 [14] that is specifically tuned to paraphrase messages in Spanish. The model generates semantically equivalent variants for a given text. Paraphrasing has been widely used as a data augmentation strategy in natural language processing to improve performance on unbalanced classification tasks [15].

Each instance from the *H* and *V* categories was augmented by generating three paraphrased versions, thus tripling their representation. For the majority classes (*Sexualization (S)* and *Not Related (NR)*), we randomly sampled 100 examples per class and generated one paraphrased version per instance to minimally enhance the diversity of these classes without introducing further imbalance.

The following generation parameters were used for paraphrasing:

- `num_beams = 5` (beam search for better quality)
- `max_length = 256`
- `num_return_sequences = 1` for most cases (or 3 for minority class augmentation)

All paraphrased examples were integrated into the training set, and the combined dataset was shuffled to promote balanced samples in mini-batches during training.

[2] https://huggingface.co/milyiyo/paraphraser-spanish-t5-small

2.2. Chunking with Overlap

Given that song lyrics often exceed the token limit accepted by transformer-based models such as BETO, we implemented a splitting strategy to divide and multiply each song into smaller, manageable segments. To preserve contextual continuity and avoid loss of semantic information across chunks, we adopted an overlapping sliding window approach with a stride equal to 50% of the size of the chunk.

We applied this strategy to both subtasks, but with different chunk sizes:

- *Subtask 1 (Binary Classification)*: Chunks of 100 tokens with 50-token overlap.
- *Subtask 2 (Multiclass Classification)*: Chunks of 150 tokens with 75-token overlap.

Each song was tokenized using the tokenizer associated with BETO. The resulting token sequence was segmented into overlapping chunks based on the configuration of each task. This approach ensures that critical contextual information is preserved and reconsidered within the text boundaries in subsequent segments.

For songs where the final chunk was shorter than the defined chunk size, the remaining tokens were appended to the previous chunk to form a complete sequence and avoid discarding information. Each resulting chunk inherited the label of the original song and was treated as an independent training instance. The chunked dataset was then used as input for model fine-tuning.

During inference, multiple predictions for a single song were generated, one per chunk, and aggregated by computing the statistical mode for all chunk-level predictions associated with the same song ID.

2.3. Label Encoding and Class Balancing

Class balancing using loss weighting was applied exclusively to Subtask 2 (multiclass classification), where class imbalance was significant in the four categories: *Sexualization (S)*, *Violence (V)*, *Hate (H)*, and *Not Related (NR)*. To mitigate the impact of imbalance, we calculated a set of normalized class weights, assigning higher penalties to minority classes during training, as recommended in previous work [16].

The weight for class i was defined as:

$$w_i = \frac{N}{n_i} \cdot \frac{C}{\sum_{j=1}^{C} \frac{N}{n_j}},$$

where N is the total number of training examples, n_i is the number of examples in class i, and C is the total number of classes. These weights were normalized and passed to the `CrossEntropyLoss` function in PyTorch. This rebalancing strategy was designed to reduce model bias toward majority classes and improve performance on underrepresented ones.

In contrast, no class weighting was used in Subtask 1 (binary classification), as the label distribution between misogynistic and non-misogynistic samples was relatively balanced.

2.4. Tokenization and Input Formatting

We used the tokenizer associated with the BETO model to tokenize the input text. The tokenizer employs WordPiece tokenization [17], which splits words into subword units, helping to manage vocabulary size and effectively handle out-of-vocabulary terms.

Each chunked text was tokenized with a maximum sequence length of 200 tokens. Sequences shorter than this limit were padded with a special padding token, while longer sequences were truncated. This ensured uniform input dimensions across all examples and compatibility with the fixed input size expected by the transformer model. After tokenization, the dataset was converted to the Hugging Face `Dataset` format [18] and split into training and validation subsets using a stratified split 95% - 5%.

2.5. Training Details

We used the Hugging Face Transformers library [18] for pretraining and processing, using the uncased BETO model. The classification head was configured to output logits corresponding to the number of target classes in each task (two for Subtask 1, four for Subtask 2), using a softmax activation over the final hidden layer. The training process was conducted using PyTorch in a GPU-enabled environment when available, with CPU fallback otherwise.

Subtask 1 (Binary Classification)

In Subtask 1, the same base model and tokenizer were used, but the training pipeline did not include class reweighting or data augmentation, as the dataset was relatively balanced. Therefore, the standard Hugging Face `Trainer` class was used without modifying the loss function.

Training was performed over 10 epochs with the same learning rate of 1×10^{-4} and evaluation strategy based on training steps. Chunked inputs of 100 tokens (with 50% overlap) were used, and the final predictions per song were obtained by majority voting in the corresponding fragments. The same macro-averaged evaluation metrics were tracked during training and validation.

Subtask 2 (Multiclass Classification)

To address class imbalance in Subtask 2, we implemented a custom trainer by subclassing the `Trainer` class and overriding the `compute_loss` method. The model minimized the weighted `CrossEntropyLoss`, where class weights were calculated based on the inverse class frequencies and normalized for numerical stability.

The model was trained using the `AdamW` optimizer [19] with a learning rate of 1×10^{-4} for 8 epochs. A batch size of 8 was used for both training and evaluation to prevent memory overflow. The evaluation strategy was set to `steps`, allowing intermediate evaluations. Performance in the validation set was monitored using macro-averaged metrics (accuracy, precision, recall, and F1), calculated using the `Scikit-learn` library [20].

3. Results

The performance of the system was evaluated using the official test sets provided by the competition organizers on the Codabench platform. Both subtasks involved submitting predictions for unseen, unlabeled data, with the evaluation performed externally using hidden gold standard labels.

Before performing the official submissions, we conducted extensive internal experiments on the training data for Task 2. The dataset was split into training and validation subsets to evaluate various configurations. Several Spanish transformer models were tested, such as `dccuchile/bert-base-spanish-wwm-uncased`, BETO-emotion, and a contextualized variant for hate speech. We experimented with chunk sizes, epochs, class balancing techniques (undersampling, oversampling), class weighting, and paraphrasing (Table 2).

Table 2
Internal Experimental Results (Validation Split)

Model	Epochs	Paraphrasing	Oversampling	Weights	Accuracy	Macro F1	Macro Recall	Notes
BETO (dccuchile)	10	No	No	No	0.6880	0.4760	0.4681	Baseline, no rebalancing
BETO (dccuchile)	10	No	No	Undersampling	0.4872	0.4119	0.4598	Reduced all classes to minority size
BETO (dccuchile)	10	No	Yes	No	0.6709	0.4030	0.4198	Simple oversampling
BETO (dccuchile)	5	No	No	Yes (auto)	**0.7179**	**0.4989**	0.4934	Best internal result
BETO (dccuchile)	5	No	No	Yes (manual)	0.7137	0.4552	0.4644	Manual weights for H and V
BETO (dccuchile)	5	No	Yes (H×3, V×2)	No	0.6624	0.4334	0.4374	Manual oversampling
BETO (dccuchile)	5	Yes	No	No	0.6624	0.4685	0.4697	Paraphrasing only
BETO-emotion	5	No	No	Yes	0.6795	0.4336	0.4430	Lower than BETO base
BETO-emotion	5	No	No	No	0.6838	0.4794	**0.4789**	Strong alternative model
BETO-contextual	5	No	No	No	0.3675	0.1344	0.2500	Discarded due to poor results

The best internal configuration was BETO (dccuchile), trained for 5 epochs using class weights calculated from the inverse class frequency, achieving a macro F1 score of 0.4989 on the validation set. This setup was used in our official submission.

3.1. Official Codabench Submissions

Table 3 summarizes all the official submissions made to the Codabench platform for both subtasks. Each configuration used the BETO model from `dccuchile`, with variations in the number of training epochs, use of paraphrasing, chunk size, and class balancing strategies.

For Subtask 1 (binary classification), the best result was achieved with 10 epochs, no paraphrasing, and 100-token chunks, confirming the robustness of the default BETO configuration in this simpler setup.

For Subtask 2 (multiclass classification), the highest macro F1 score was obtained by combining class weighting, paraphrasing applied to all classes, and longer 150-token chunks. This indicates that both data augmentation and tailored input segmentation played a key role in handling class imbalance and improving overall generalization.

These results are consistent with our internal validation experiments and informed the final design of our submitted models.

Table 3
Official Codabench Submissions and Evaluation Metrics

ID	Task	Epochs	Paraphrasing	Chunks/Overlap	Weights	Macro F1	Precision	Recall	Notes
289081	Task 2	8	Yes (all classes)	150 / 50%	Yes	**0.4929**	0.5157	0.4889	Best Task 2 result
289034	Task 1	10	No	100 / 50%	No	**0.7876**	0.8035	0.7765	Best Task 1 result
288285	Task 1	8	No	100 / 50%	No	0.7820	0.7782	**0.7865**	–
288243	Task 2	8	No	100 / 50%	Yes	0.4250	0.4527	0.4279	No augmentation
288151	Task 2	8	Yes (only majority class)	150 / 50%	Yes	0.4523	0.4572	0.4531	Paraphrased only one class
286987	Task 2	7	No	100 / 50%	Yes	0.4155	0.4425	0.4248	–
286985	Task 1	7	No	100 / 50%	No	0.7735	0.7717	0.7754	–
286841	Task 1	6	No	100 / 50%	No	0.7684	0.7632	0.7752	–
285881	Task 2	6	No	100 / 50%	Yes	0.4182	0.4334	0.4290	–

3.2. Final Results for Subtask 1: Binary Classification

This subtask is dedicated to identifying whether a song lyric contains misogynistic content or not. A simpler pipeline was adopted without data augmentation or class reweighting. Predictions were generated using 100-token chunks with 50% overlap.

The best-performing configuration used 10 training epochs and no balancing, leveraging the strong generalization ability of the BETO model. Table 4 includes a comparison with the official baseline provided by the organizers.

Table 4
Evaluation results for Subtask 1 (Binary Classification)

System	Precision	Recall	F1-score
Our Model (Best)	0.8035	0.7765	**0.7876**
Baseline	–	–	0.7434

3.3. Final Results for Subtask 2: Multiclass Classification

Table 5 presents the final evaluation results for Subtask 2. The system achieved moderate performance in the four target categories: *Sexualization (S)*, *Violence (V)*, *Hate (H)*, and *Not Related (NR)*.

Submissions that used paraphrasing for all classes and larger chunk sizes (150 tokens, 50% overlap) yielded the best performance. Class weights were included in all multiclass experiments to mitigate imbalance. Data augmentation through paraphrasing significantly improved macro metrics, especially in minority classes.

Table 5 also includes the macro F1-score obtained by the official baseline system provided by the organizers for reference.

Table 5
Evaluation results for Subtask 2 (Multiclass Classification)

System	Macro Precision	Macro Recall	Macro F1-score
Our Model (Best)	0.5157	0.4889	**0.4929**
Baseline	–	–	0.4151

3.4. Summary of Experimental Strategy

Overall, the results highlight the effectiveness of the BETO-based architecture and the chunking-based strategy, particularly in the binary setting. Performance in the multiclass setting, while lower, reflects the added difficulty of fine-grained classification. Our internal validation results informed the final submission strategy, showing that combining paraphrasing, adjusted chunk sizes, and class weighting improved performance in the more complex Task 2.

Future improvements could include testing hierarchical models, advanced sampling strategies, or exploring multilingual pretraining for more robust representations.

4. Conclusions

This paper describes our approach for the MiSongyny shared task at IberLEF 2025, focusing on binary and multiclass subtasks designed to identify misogyny in Spanish song lyrics at varying levels of granularity. Unlike social media, where misogynistic content is often explicit and filled with harsh language, song lyrics frequently employ metaphorical and poetic expressions, creating unique challenges for automatic detection.

Our solutions leverage the BETO model, a Spanish-specific monolingual BERT variant, which we fine-tuned for the task of misogyny identification in song lyrics. To tackle the issues of input length limitations and class imbalance, we utilized a sliding window approach to divide the lyrics into overlapping stanzas and employed data augmentation with a Spanish paraphrasing large language model to increase the representation of the minority class.

The final result of our model for Subtask 1 (F1-score = 0.7876) clearly surpasses the official baseline (F1 = 0.7434), as shown in Table 4. The simplicity of the binary classification task allowed the base BETO model to perform strongly even without data augmentation or class reweighting. Our decision to train for 10 epochs and use a clean chunking strategy (100 tokens, 50% overlap) was sufficient to generalize well to the test set. These results confirm that BETO is well-suited for detecting misogynistic content in a binary setting without heavy customization.

For Subtask 2, our best submission reached a macro F1-score of 0.4929, significantly outperforming the baseline score of 0.4151, as shown in Table 5. The multiclass setting introduced additional challenges such as class imbalance and fine-grained distinctions between types of misogyny. The improvement was mainly due to the use of paraphrasing for all classes, chunk length adjustment (150 tokens), and class-weighted loss. These techniques helped the model better capture minority categories such as *Violence (V)* and *Hate (H)*, which tend to be underrepresented in training data.

Overall, both subtasks benefited from careful fine-tuning of training strategies. However, Subtask 2 demanded additional effort in terms of data processing and class balancing due to its higher complexity.

The results emphasize the effectiveness of employing a BETO model alongside a chunking strategy and data augmentation in detecting misogynistic content in song lyrics, which are often lengthy and rich in figurative language.

Declaration on Generative AI

During the preparation of this work, we applied Grammarly's model for grammar and spelling checks. After using these services, we reviewed and edited the content as needed and assume full responsibility for the content of the publication.

References

[1] T. Alcántara, M. Soto, C. Macias, O. Garcia-Vazquez, A. Espinosa-Juarez, H. Calvo, J. E. Valdez-Rodríguez, E. Felipe-Riveron, Overview of MiSonGyny at IberLEF 2025: Misogyny Speech Detection in Spanish Language Song Lyrics, Procesamiento del Lenguaje Natural 75 (2025).

[2] W. H. Organization, Violence against women prevalence estimates, 2018: Global, regional and national prevalence estimates for intimate partner violence against women and global and regional prevalence estimates for non-partner sexual violence against women, 2021. URL: https://www.who.int/publications/i/item/9789240022256, accessed: 2024-05-29.

[3] E. Fersini, M. Anzovino, P. Rosso, Overview of the task on automatic misogyny identification at ibereval 2018, in: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval), 2018, pp. 214–228.

[4] E. Fersini, D. Nozza, P. Rosso, Overview of the evalita 2018 task on automatic misogyny identification (ami), in: Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, volume 2263 of *CEUR Workshop Proceedings*, CEUR-WS, 2018. URL: http://ceur-ws.org/Vol-2263/paper009.pdf.

[5] P. Fortuna, S. Nunes, A survey on automatic detection of hate speech in text, ACM Computing Surveys (CSUR) 51 (2018) 1–30.

[6] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, arXiv preprint arXiv:2308.02976 (2023). URL: https://arxiv.org/abs/2308.02976, accepted at PML4DC, ICLR 2020.

[7] 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, arXiv preprint arXiv:1907.11692 (2019).

[8] J. Cañete, S. Donoso, F. Bravo-Marquez, A. Carvallo, V. Araujo, Albeto and distilbeto: Lightweight spanish language models, in: Proceedings of the 13th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022.

[9] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL: https://arxiv.org/abs/1910.01108. arXiv:1910.01108.

[10] R. Calderón-Suárez, R. M. Ortega-Mendoza, M. A. Márquez-Vera, F. A. Castro-Espinoza, Identificación automática de contenido misógino en redes sociales: Un enfoque basado en transferencia de conocimiento proveniente de canciones, Computación y Sistemas 28 (2024) 283–299. doi:10.13053/CyS-28-1-4896.

[11] H. Gómez-Adorno, G. Bel-Enguix, H. Calvo, S. Ojeda-Trueba, S. T. Andersen, J. Vásquez, S. Ojeda-Trueba, T. Alcántara, M. Soto, C. Macías, Overview of HOMO-MEX at IberLEF 2024: Hate Speech Detection Towards the Mexican Spanish speaking LGBT+ Population, Procesamiento del Lenguaje Natural 73 (2024) 393–405.

[12] 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.

[13] J. Cañete, G. Chaperon, R. Fuentes, J. Pérez, Beto, the spanish bert, in: Proceedings of the Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberLEF), 2020.

[14] 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, Journal of Machine Learning Research 21 (2020) 1–67.

[15] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, E. Hovy, A survey of data augmentation approaches for nlp, arXiv preprint arXiv:2105.03075 (2021). Accepted to Findings of ACL 2021.

[16] M. Buda, A. Maki, M. A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural Networks 106 (2018) 249–259.

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2019).

[18] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.

[19] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2019).

[20] 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.