GRUPPETTOZZO at MultiPRIDE 2026: Detecting LGBTQ+ Reclamatory Intent via Context-Aware Transformers

Federico Traina[1,†], Alessandro Santoro[1,†], Gabriele Greco[1,†], Irene Siragusa[1,*] and Roberto Pirrone[1]

[1] *Department of Engineering, University of Palermo, Viale delle Scienze, Edificio 6, Palermo, 90128, Sicily, Italy*

Abstract

In this report, we present the proposed approach of the GRUPPETTOZZO team for the detection of reclamatory usage of terms related to LGBTQ+. In the context of the multilingual MultiPRIDE challenge in the EVALITA 2026 campaign, we proposed a transformer-based encoder-only method to classify the textual content of the given tweets (task A) and in conjunction with the contextual information provided by the user biography (task B). To handle the unbalance in the given data set, we explored the usage of both weighted and focal loss functions and two data augmentation strategies, to better generalize the performance of developed models in the target multilingual context. We submitted two runs for each task, one using a target monolingual Language Model for each specific language, and one using a multilingual one. Both proposed approaches, overcame the baseline 6 times out of 10 and reached the 1st place in subtask B2.

Keywords

Context-Aware Transformers, LGBTQ+ Slurs Reclamation, Data Augmentation, Language Models

1. Warning

This paper contains examples of explicitly offensive content.

2. Introduction

Language within the LGBTQ+ context is characterized by complex phenomena such as reclamation, whereby terms originally intended as slurs are repurposed by community members for positive and/or identity-affirming purposes. In the context of social networks and hate speech detection applications, an important challenge lies in distinguishing the correct usage of such slurs that may have both a reclamation and a denigratory intent. MultiPRIDE challenge [1] attempts to address this complex phenomenon proposing two multilingual binary classification tasks. In particular, the objective is to determine whether a given tweet, uses a term related to LGBTQ+ context with a reclamatory intent (task A) in Italian (subtask A1), Spanish (subtask A2) and English (subtask A3). In addition, organizers add the user biography (user bio) when available as extra context for the classification (task B) in Italian (subtask B1) and Spanish (subtask B2), as well as encouraging cross-lingual applications (subtask A-multi and B-multi).

This paper describes the system developed by the GRUPPETTOZZO team for tasks A and B of the MultiPRIDE challenge in the context of the EVALITA 2026 campaign [2]. Our approach consists in fine-tuning encored-only Language Models (LMs) to address task A and, for task B, we integrated textual biographies as pragmatic context into the model's input. Moreover, we handle tasks A and B in both monolingual and multilingual configuration. Our approach can be summarized in five stages,

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

[*] Corresponding author.

[†] These authors contributed equally.

federico.traina01@community.unipa.it (F. Traina); alessandro.santoro04@community.unipa.it (A. Santoro); gabriele.greco07@community.unipa.it (G. Greco); irene.siragusa02@unipa.it (I. Siragusa); roberto.pirrone@unipa.it (R. Pirrone)

0009-0005-8434-8729 (I. Siragusa); 0000-0001-9453-510X (R. Pirrone)

as can be seen in Figure 1. Firstly we perform a data augmentation phase to enrich the original data set, following two different strategies. Textual contents are pre-processed through a data cleaning procedure before being injected into the developed Context-Aware Architecture, made up of a LM backbone, a custom pooling strategy and a classification head that performs the final prediction.

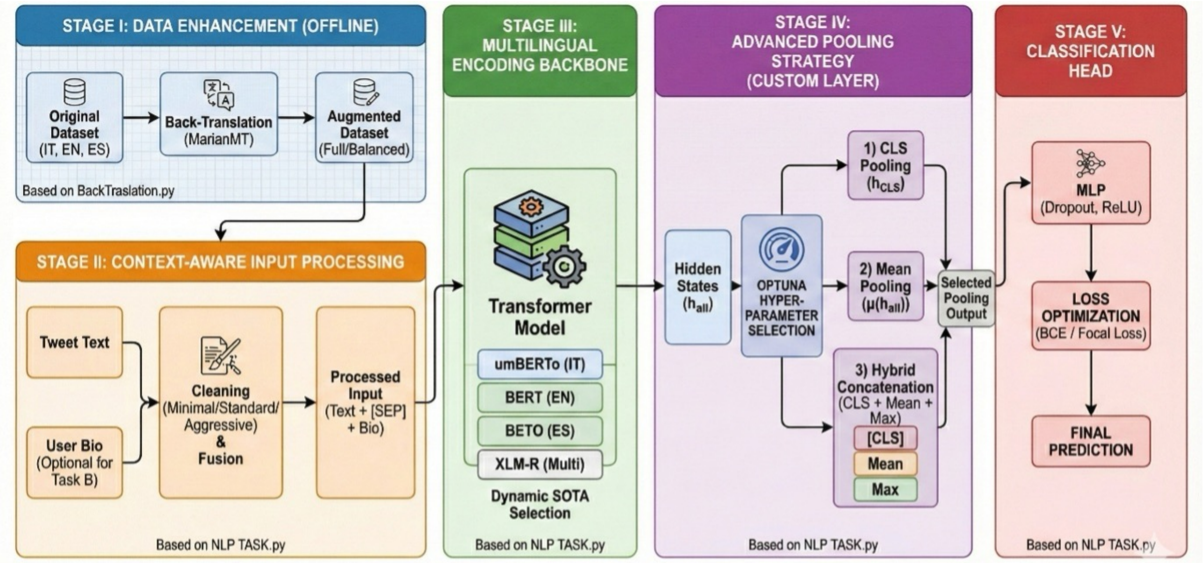
STAGE I: DATA ENHANCEMENT (OFFLINE)
Original Dataset (IT, EN, ES)
Back-Translation (MarianMT)
Augmented Dataset (Full/Balanced)
Based on BackTraslation.py
STAGE II: CONTEXT-AWARE INPUT PROCESSING
Tweet Text
User Bio (Optional for Task B)
Cleaning (Minimal/Standard/Aggressive) & Fusion
Processed Input (Text + [SEP] + Bio)
Based on NLP TASK.py
STAGE III: MULTILINGUAL ENCODING BACKBONE
Transformer Model
umBERTo (IT)
BERT (EN)
BETO (ES)
XLM-R (Multi)
Dynamic SOTA Selection
Based on NLP TASK.py
STAGE IV: ADVANCED POOLING STRATEGY (CUSTOM LAYER)
Hidden States (h_{all})
OPTUNA HYPER-PARAMETER SELECTION
1) CLS Pooling (h_{CLS})
2) Mean Pooling ($\mu(h_{all})$)
3) Hybrid Concatenation (CLS + Mean + Max)
[CLS]
Mean
Max
Selected Pooling Output
Based on NLP TASK.py
STAGE V: CLASSIFICATION HEAD
MLP (Dropout, ReLU)
LOSS OPTIMIZATION (BCE / Focal Loss)
FINAL PREDICTION
Based on NLP TASK.py

Figure 1: Overview of the proposed approach.

We submitted two runs for each subtask using a monolingual and a multilingual LM as backbone. The architecture developed was able to exceed the baseline 6 times out of 10, and reached the 1st place in subtask B2.

This report is structured as follows: data augmentation and pre-processing procedures, and the proposed system are described in Section 3, while results and related discussion are presented in Sections 4 and 5. Section 6 contains final remarks.

3. Description of the System

MultiPRIDE data set is arranged in three different splits, one for each language, comprehensive of 1,000 labeled tweets, where positive classes are associated to usage of slurs in a reclamatory manner. For Spanish and Italian splits, the user bio field was provided to address task B, if available. For the development phase, we split the data set using an 80-20 ratio and a stratified strategy.

The system processes data in two distinct structural modalities designed to evaluate the impact of external context on classification, depending exclusively on the target task and not on the language:

- **Text-Only** the input to the model is exclusively the textual content of the tweet (subtasks A1, A2, A3, A-multi);
- **Context-Aware** an enriched representation is generated by concatenating the original tweet with the user bio (subtasks B1, B2, B-multi). The input follows the pattern:

  ```
  TWEET [SEP] CONTESTO: BIO
  ```

  where the [SEP] token serves as a segment delimiter, thus enabling the model to distinguish between the two distinct information sources within the same input sequence. Whenever the user bio is not available, it is set to an empty string.

3.1. Data Augmentation and Pre-processing

Given the unbalanced distribution of positive labels and the limited number of samples in the training set, we implemented a back-translation pipeline based on MarianMT [3] and OPUS-MT [4, 5] models. This technique involves translating the text from the source language to a pivot language and subsequently re-translating it back into the original language. This process generates a paraphrase that preserves the core semantic content while introducing lexical and syntactic variance. We used English as pivot language for Italian and Spanish splits, while Spanish as pivot language for English split.

We experimented with two augmentation configurations as shown in Figure 2:

- **Full Augmentation** A paraphrase is generated for *every sample* in the data set. On doing this, we double the training set size, exposing the model to greater linguistic variability and helping to prevent overfitting on specific syntactic patterns.
- **Balanced Augmentation** Augmentation is applied *exclusively* to the minority class, i.e. the positive labeled samples. This technique reduces the class imbalance ratio, enabling the model to learn more robust features for the positive class without resorting to naive oversampling, which would simply duplicate existing examples.

After the data augmentation phases, a check over the generated samples have been conducted and eventually duplicated samples, which may result if the back-translation system outputs the original text, have been removed.

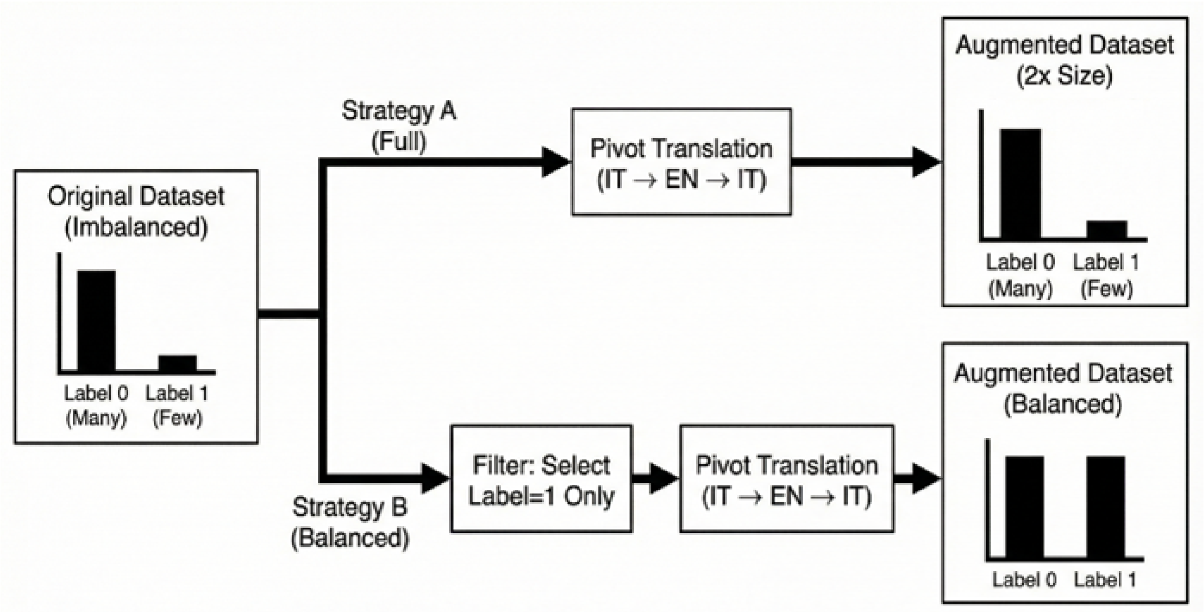

Figure 2: Overview of the data augmentation pipeline with full and balanced augmentation strategies.

Given the social media nature of the data set, we formalized three different strategies of data pre-processing, and we investigated the impact of each strategy:

- **Minimal** the original text is converted in lowercase and no further data cleaning operation are performed, thus the textual content is left unchanged and are kept both the total amount of information and any noisy information.
- **Standard** a demojization phase is performed over the lowercased input text, and any present emoji is converted in a textual string, e.g. 🏳️‍🌈 →`:rainbow_flag:`.
- **Aggressive** both emoji and stopwords are removed and textual content is lowercased.

The choice to lowercase the input in all the pre-processing strategies is a strategic decision for our task. On social media, capitalization is often irregular or used exclusively for emphasis. Normalizing the text helps the model focus on semantic content rather than graphical variations, thus enhancing its generalization capabilities.

Emojis are dense semantic and sentiment information. Removing them eliminates any potential signals of LGBTQ+ community affiliation (e.g., colored hearts, pride flags) or references to the sentiment of the given tweet. The demojization projects these symbols into the semantic space of the BERT tokenizer [6], allowing the model to proper interpret those meta-linguist features. In the aggressive strategy, the stopwords removal can be detrimental in pragmatic tasks, as functional particles (negations or pronouns) which can radically alter the sentence's meaning.

3.2. Context-Aware Architecture

The proposed Context-Aware Architecture departs from standard classification models by integrating modular components designed to maximize semantic feature extraction and enhance model robustness in the presence of imbalanced data. It is made up of a LM backbone, followed by a neural module which further processes the dense representation obtained from the LM with a custom pooling strategy and a classification head that performs the final prediction.

3.2.1. Language Model Backbones

The proposed model leverages on fine-tuning of pre-trained transformer models [7], selected to maximize the semantic representation in each target languages.

For the Italian split, we used lupobricco/umBERTo_fine-tuned_hate_offensivity[1] (umBERTo). This model has been obtained after a domain-specific fine-tuning of UmBERTo [8], which is an Italian LM based on RoBERTa model [9], and trained over a vast Italian corpus using the Whole Word Masking (WWM) technique. Specifically, the WWM requires the model to predict entire words even when only a sub-word unit is masked. This results to be a crucial feature in the context of Romance languages (Italian, Spanish), which are characterized by an abundance of suffixes and complex conjugations. In addition, UmBERTo more effectively handles the complex morphology of the Italian language by utilizing a SentencePiece-based tokenizer, which better preserves the semantic roots of words. The choice of umBERTo provides a critical advantage in Task A. This specific version has been further fine-tuned for Hate Speech Detection, thus resulting (i) already optimized to recognize patterns of toxicity and (ii) the need of an extensive training from scratch to identify aggressive intent is significantly reduced.

For the English split we adopted the standard BERT model in its base and uncased version [6], which represent a state-of-the-art LM.

As for the Spanish data set, we used BETO [10], a BERT-based model trained with the WWM technique over Spanish data. Analogously for the chosen Italian LM, WWM allow a better modeling of the Spanish language and the model itself is forced to learn deeper syntactic and semantic representations, consistently outperforming standard Multilingual BERT (mBERT) [6].

We also considered a multilingual backbone for the cross-lingual classification task. We opt for XLM-RoBERTa [11], which scales cross-lingual pre-training utilizing 2.5 TB of filtered CommonCrawl text across 100 languages. Unlike mBERT, XLM-RoBERTa omits the Next Sentence Prediction (NSP) objective and is trained with significantly larger batches and longer durations. This architecture facilitates the creation of a shared semantic space where semantically equivalent terms in different languages (e.g. 'gay' in English and 'gay' in Italian) are mapped to proximal vectors. This cross-lingual alignment capability is fundamental to our system, enabling the transfer of knowledge from high-resource examples in English and Spanish to improve predictions even for underrepresented classes [12]

3.2.2. Neural Module Architecture

Despite the different LM backbone used, the output of the last layer $H \in \mathbb{R}^{l \times d}$, where l is the length of the given sequence and d the dimension of the hidden state, serves as input to a neural model.

[1] https://huggingface.co/lupobricco/umBERTo_fine-tuned_hate_offensivity

From this vector representation, three pooling strategies are defined:

- **CLS Pooling** $\mathbf{v}_{cls}$ which selects the embedding of the `[CLS]` token;
- **Mean Pooling** $\mathbf{v}_{mean}$ across the target sentence removing any padding tokens;
- **Hybrid Pooling** is the proposed advanced concatenation strategy, which consists in concatenating the vector representation obtained by the previous pooling strategies and a Max Pooling $\mathbf{v}_{max}$ over the dense representations for each sample, removing any padding token. The final representation is obtained as $\mathbf{v}_{final} = \mathbf{v}_{cls} \oplus \mathbf{v}_{mean} \oplus \mathbf{v}_{max}$, with dimension equal to $3 \times d = 2304$.

While $\mathbf{v}_{cls}$ is optimized during LM training, $\mathbf{v}_{mean}$ captures the global context, and $\mathbf{v}_{max}$ highlights salient signals (e.g., a single offensive term). The concatenation of these vectors provides a more comprehensive overview of the input data to the final classification.

The classification head applies, over the obtained data after pooling, a dropout layer to prevent the overfitting of the network, followed by a hidden dense layer, a ReLU activation function and the classification layer.

3.3. Training Details

For the training phase, we adopted an hyperparameter optimization strategy, including architectural details such as the pooling strategy, dropout value, the size of the hidden MLP layer, and training details as the number of layers to freeze in the LM backbone, loss function, optimizer and learning rate, described below.

To balance learning capacity with the stability of pre-trained weights in the LM backbones, we implemented a selective freezing mechanism. The initial embeddings and the firsts n encoder layers can be frozen in order to (i) force the model to leverage its acquired foundational linguistic knowledge and (ii) focus the training process exclusively on the higher-level layers. Such high-level features, then combined with the neural model architecture, can deeply focus toward the target task during its training phase.

We considered two different options as for the loss function: a weighted Binary Cross-Entropy (BCE) loss with the inverse class ratio as weight, and a Focal Loss [13], parametrized with γ, optimized to down-weight easy samples, and compelling the model to focus on more complex, hard-to-classify instances.

As optimizer during the training phase, we both evaluated AdamW [14] and EvoLved Sign Momentum (Lion) [15]. Unlike AdamW, which computes the first and second moments (mean and variance) of the gradients, the Lion optimizer utilizes only the sign of the gradient for weight updates. This approach simplifies the optimization process, reduces memory overhead, and provides a more robust regularization effect, specifically in fine-tuning on small data sets such as the MultiPRIDE one.

We used Optuna [16], a bayesian optimization to determine the optimal hyperparameter configuration for each task and subtask, and a comprehensive overview of all the considered hyperparameters reported Table 1.

Table 1
Overview of all the considered hyperparameters.

Hyperparameter	Search space / possible values
Pooling strategy	CLS Pooling, Mean Pooling, Hybrid Pooling
# frozen layers LM	1,2,3,4,5,6
MLP size	128, 256, 623
Dropout	from 0.1 to 0.5
Loss Function	Weighted BCE, Focal Loss
Gamma (Focal Loss)	from 0.5 to 3
Learning Rate	from 1e-5 to 5e-5
Optimizer	AdamW, Lion

The developed system and the training pipeline have been implemented in PyTorch [17] and with HuggingFace Transformers [18]. To optimize the computational efficiency we adopted a Mixed Precision Training (FP16) technique. We trained the models up to a maximum of 8 epochs and implemented an Early Stopping strategy over the macro F1-score in our validation set with a patience value equal to 3. Experiments have been conducted over Google Colab instances equipped with one GPU Tesla T4 and Kaggle instances with one NVIDIA TESLA P100 GPU.

4. Results

For each subtask diverse training configuration have been explored, namely the diverse augmentation strategies (without, full and balanced), data cleaning techniques (minimal, standard, aggressive), combined with the best hyperparameter obtained with OPTUNA for each configuration (Table 1). In this section we report the configurations which obtained the best results in terms of macro F1-score over our validation set and the official test set for both tasks.

In Table 2 are reported our development results for Task A, computed on an internal validation set obtained via a stratified 80/20 split of the official training data.

Table 2
Results over internal validation set for Task A. Bold results refer to the highest ones per language split.

Subtask	Data Cleaning	Data Augmentation	Pooling	# frozen LM layers	MLP size	Optimizer	Loss	Macro F1-score
A1 (IT)	Standard	None	CLS	2	128	Lion	BCE	0.9014
A2 (ES)	Minimal	None	Mean	1	128	AdamW	BCE	0.7320
A3 (EN)	Minimal	None	CLS	4	128	Lion	Focal	0.6288
A-Multi	Aggressive	None	Hybrid	5	128	AdamW	Focal	0.7898
A1 (IT)	**Minimal**	**Full**	**Hybrid**	**4**	**128**	**Lion**	**BCE**	**0.9702**
A2 (ES)	**Minimal**	**Full**	**CLS**	**6**	**128**	**Lion**	**Focal**	**0.9047**
A3 (EN)	Minimal	Full	Mean	4	128	AdamW	Focal	0.8849
A-Multi	Standard	Full	Mean	3	128	Lion	Focal	0.8816
A1 (IT)	Minimal	Balanced	Hybrid	0	128	Lion	Focal	0.9478
A2 (ES)	Minimal	Balanced	Mean	2	128	Lion	Focal	0.8987
A3 (EN)	**Standard**	**Balanced**	**Hybrid**	**4**	**256**	**Lion**	**Focal**	**0.9190**
A-Multi	**Standard**	**Balanced**	**Mean**	**5**	**128**	**Lion**	**Focal**	**0.8871**

Lowest results are obtained without data augmentation, suggesting an intrinsic complexity of the task, especially in contest of data scarcity. Both strategies of full and balanced data augmentation led to higher performances, almost over 0.90 for all configurations, confirming that the initial poor generalization capabilities are mainly addressed to the reduced number of training samples.

The Italian split (subtask A1) shows the overall best results, even without data augmentation and increasing with both full and balanced augmentation. This behavior may suggest an minor linguistic ambiguity in Italian tweets, when compared with other languages. While both Italian and Spanish data benefits from the full augmentation strategy, English and multilingual splits highly benefit from the balanced augmentation.

As for data cleaning, the minimal strategy, which essentially left the input data unchanged, resulted to be the overall best choice, thus assessing that punctuation elements and emojis are semantically essential for the given task. Multilingual setup, on the contrary, needs a standard or aggressive data pre-processing strategy. These performances show that a normalization step helps cross-lingual models to better map concepts in a common latent and embedding space.

Regarding the pooling strategies, results obtained with the concatenation of the three pooling operations (hybrid pooling) are the best one for Italian and English splits, while Spanish split benefits from the usage of the simple `[CLS]` token, while the mean pooling was preferred for the multilingual setup.

The conjunct use of the focal loss with Lion optimizer resulted overall the best choice to focus the learning towards the more complex samples and the unbalanced class. Furthermore, the optimal size for the MLP layer was equal to 128 and overall best results are obtained though a soft training of the LM backbone, that is with a higher lever of transformer layers kept frozen during the training.

In Table 4 are reported the obtained results for Task B computed on an internal validation set obtained via a stratified 80/20 split of the official training data, within the different augmentation strategies for each language and with the multilingual configuration. In this task, the additional information provided by the user bio, introduces identity elements about the author of the target tweet, which can used by the model as suggestion for the final classification.

Table 3
Results over validation set for Task B. Bold results refer to the highest ones per language split.

Subtask	Data Cleaning	Data Augmentation	Pooling	# frozen LM layers	MLP size	Optimizer	Loss	Macro F1-score
B1 (IT)	Aggressive	None	CLS	1	256	AdamW	Focal	0.9043
B2 (ES)	Standard	None	Hybrid	1	128	AdamW	Focal	0.7064
B-Multi	Aggressive	None	Hybrid	1	256	AdamW	Focal	0.8340
B1 (IT)	**Standard**	**Full**	**Mean**	**4**	**128**	**AdamW**	**Focal**	**0.9667**
B2 (ES)	**Standard**	**Full**	**CLS**	**0**	**256**	**AdamW**	**Focal**	**0.9541**
B-Multi	Standard	Full	Hybrid	3	128	Lion	BCE	0.9015
B1 (IT)	Standard	Balanced	Hybrid	3	128	Lion	BCE	0.9602
B2 (ES)	Aggressive	Balanced	Hybrid	5	128	AdamW	Focal	0.9010
B-Multi	**Standard**	**Balanced**	**Hybrid**	**0**	**128**	**Lion**	**Focal**	**0.9041**

Also for this case, the full augmentation results the best data augmentation option, with the only exception for the multilingual case, which slightly benefits from the balanced augmentation. Overall, performances of the developed models without data augmentation are considerably lower compared to the ones with data augmentation.

Despite the additional information provided by the user bio, this did not result sufficient to compensate both the unbalanced labels distribution and the poor quantity of samples, making them comparable to the results obtained with the task A. More specifically, Italian and Multilingual splits benefits from the additional information of the user bio, compared to the Spanish one, whose performances decrease. We hypothesize that the model can effectively benefits of this additional context for the cases in which a high coherence is present between the user bio and the target tweet, suggesting that in the Italian split this correlation is higher than in the Spanish one. However, the overall best performances are obtained with the full data augmentation strategy and implementing the standard data cleaning method and a training based on Focal Loss and AdamW as optimizer.

For each task, we submitted the prediction obtained with both monolingual (run 1) and multilingual (run 2) model reaching the best performances over our validation set (Tables 2 and 3), and in Table 4 are collected the obtained results over the test set.

Obtained results confirms the excellent performance of the fine-tuned umBERTo model. Specifically, in subtask B1 we obtained a macro F1-score of 0.8979 with the monolingual proposed architecture. On the other hand, the integration of user biographies did not lead to an improvement: the macro-F1 went from 0.8735 (Task A) to 0.8681 (Task B) in the proposed multilingual model.

A drastic performance decline was observed for the English split in subtask A3, compared to the result observed during the development phase, where we reached an macro F1-score of only 0.5979 with the multilingual backbone. This suggests either significant overfitting on the training data or a semantic distributional shift in the test set compared with the training one. The different nature of the data may led the model to struggled to differentiate between reclaimed usage and actual hate speech.

For Spanish subtasks, the monolingual system achieved a macro F1-score of 0.7289 in subtask A2 and

Table 4
Results over test set for Task A-B. Official score for ranking is the Macro F1-Score

Subtask	Run	Rank	Macro Precision	Macro Recall	Macro F1-Score
A1 (IT)	1	6	0.9144	0.8594	0.8834
A1 (IT)	2	9	0.8725	0.8745	0.8735
A2 (ES)	1	8	0.7387	0.7205	0.7289
A2 (ES)	2	5	0.7273	0.7882	0.7506
A3 (EN)	1	14	0.5516	0.5248	0.5285
A3 (EN)	2	6	0.5866	0.6184	0.5979
B1 (IT)	1	3	0.9061	0.8903	0.8979
B1 (IT)	2	6	0.9247	0.8313	0.8681
B2 (ES)	**1**	**1**	**0.7293**	**0.7315**	**0.7304**
B2 (ES)	2	8	0.7506	0.6560	0.6856

0.7304 in subtask B2, in which placed in the first spot of the ranking. This corresponds to an absolute gain of 0.0015 macro F1 points for A2 over B2. This suggests that, for the Spanish test set, biographical information may have introduced contextual noise rather than clarifying the intent of the speaker. Moreover, the usage of a monolingual model was crucial compared with a multilingual one, for which a severe downgrade in performances with the additional bio information can be seen.

5. Discussion

A deeper analysis of the results obtained allows us to better analyze the complexity of the task and compare the effectiveness of the proposed techniques.

A counterintuitive finding from our experiments is the superiority of less invasive data cleaning strategies (minimal and standard) compared to the aggressive one. In traditional NLP tasks, stopwords and punctuation are often discarded as noise. However, within the LGBTQ+ linguistic context, the use of emojis (e.g., 🏳️‍🌈, ✨) and pronouns frequently serves as a crucial indicator of the author's identity and its possible reclaimed intent. The removal of these elements, in the aggressive strategy, deprives the model of fundamental pragmatic signals.

The implementation of a augmentation strategy with Back-Translation proved to be pivotal. Models trained exclusively on the original data set exhibited clear signs of data scarcity and struggled with generalization. The introduction of augmented variants stabilized the training process, enabling the multilingual model (XLM-RoBERTa) to achieve performance levels comparable to those of monolingual models, reaching a macro F1-score higher than 0.90 in development phase in tasks B. This demonstrates a significant cross-lingual transfer learning capability of the model, where the model successfully leverages augmented patterns across the three languages.

An outperforming pooling strategy cannot be uniquely identified, but the overall highest performance in both development and testing phases are addressed to the proposed hybrid pooling. This suggests that the salient information is not merely concentrated in the `[CLS]` token of the sequence, but is distributed across the tweet and the biography (mean and max pooling).

Moreover, the usage of the focal loss during the training guaranteed a more robust and generalization capabilities in the model itself which performances did not deteriorated significantly during the testing phase, with the only exception for subtask A3.

5.1. Error Analysis

A detailed examination of misclassified instances reveals systematic patterns that illuminate the intrinsic challenges of reclamation detection across the three languages examined.

Misclassification patterns reveal distinct difficulties across the three languages. In Italian, the model frequently misinterprets meta-linguistic discussions about slur usage as reclamation attempts. For instance the following tweet

> *"Ah perché dire 'sei troppo femminile' è omofobia ma coglione, frocio e compagnia bella no?"*

discusses the inconsistency of considering certain terms offensive while accepting others. Overall it results as a reflexive commentary on language norms rather than genuine reclamation. Similarly, expressions like

> *"Prima che le haters mi inizino a dire eH mA qUeStA è OmOfObIa, volevo comunicarvi che frocia lo sono pure io"*

represent preemptive self-identification that the model struggles to contextualize correctly.

In Spanish, the presence of explicit LGBTQ+ pride hashtags (e.g., #OrgulloLGTBI, #LGTBI) creates ambiguity. The model tends to overpredict reclamation when these markers appear alongside slurs, even in non-reclamatory contexts. For example,

> *"Gracias @USER por #Mansos y por #MaricónPerdido. Gracias a las personas que no se callan. Gracias a las personas que escuchan. Gracias a las personas que leen. Cómo nos gusta leer con orgullo! Feliz día ⊠ #OrgulloLGTBI #OrgulloLGTBIQ+ #Orgullo2021 "*

expresses gratitude for LGBTQ+ literature (referencing book titles #Mansos and #MaricónPerdido) within a celebratory pride context. Despite the positive framing and absence of reclamatory intent, the co-occurrence of pride-related terminology and the slur "maricón" in the book title confounds the classifier.

English presents the most severe challenges, with the majority of errors involving subtle pragmatic distinctions. The data set contains numerous instances of meta-discourse about reclamation itself, such as

> *"I don't like the word faggot and don't like the word queer, either. I see them both as insults"*

where community members explicitly reject reclamation. The model fail to correct detect this particular nuance, misclassify it as reclamation. Additionally, ironic or sarcastic usages like

> *"Honestly though faggot is such a fun word to say and all of my lgbt friends and I call each other that like 24/7"*

are systematically misclassified as reclamatory due to the positive framing and in-group reference, despite lacking genuine empowerment intent.

Certain reclamatory strategies prove particularly challenging across all languages. Empowerment through defiance—exemplified by the English tweet

> *"I enjoy being the wrong faggot to fuck with. Plus, sometimes it's just fun to fight"*

which was annotated as reclamatory, requires understanding implicit self-empowerment signals that extend beyond explicit pride markers. Similarly, the Italian expression

> *"BUONGIORNO AMICI BUON MESE DELLA FROCIAGGINE, LESBAGGINE, BISESSUALISMO, TRANSGENDERESIMO QUEER"*

employs creative neologisms and celebratory tone that the model struggles to distinguish from hate speech.

The biography context, despite being incorporated into the model, provides insufficient disambiguation in many cases. Users who identify as LGBTQ+ in their bios may discuss homophobic incidents they experienced without reclaiming the slurs used against them, yet the model systematically misinterprets author identity as sufficient evidence for reclamation.

6. Conclusion

In this paper we reported the architecture proposed by the GRUPPETTOZZO team for MultiPRIDE tasks A and B promoted at the EVALITA 2026 campaign. Our experimental results demonstrate that the identification of reclamation intent in slurs' usage is not a mere lexical problem, but a semantic one. Developed system won in subtask B2 reaching a macro F1-score of 0.7304, while reaching competitive results in the other subtasks and the baseline 6 times out of 10. Obtained results shows that preserving the non-verbal pragmatic markers (emojis) while adopting a suitable data augmentation strategy and a training procedure based on a focal loss, were our core features to increment model performance over the two proposed tasks. The concatenation of the textual content and the user bio for tasks B, allowed a performance increment when compared to tasks A for both Italian and Spanish split when a monolingual LM was used as backbone. On the contrary, the opposite behavior is found for multilingual model, which struggles in discriminating the additional user bio information.

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini 3 in order to: Text Translation, Grammar and spelling check, Paraphrase and reword. Further, the author(s) used Gemini 3 for figures 1, 2 and 3 in order to: Generate images. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[2] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

[3] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, et al., Marian: Fast neural machine translation in C++, in: Proceedings of ACL 2018, System Demonstrations, Melbourne, Australia, 2018, pp. 116–121. URL: https://aclanthology.org/P18-4020/. doi:10.18653/v1/P18-4020.

[4] J. Tiedemann, M. Aulamo, D. Bakshandaeva, M. Boggia, S.-A. Grönroos, T. Nieminen, A. Raganato, Y. Scherrer, R. Vazquez, S. Virpioja, Democratizing neural machine translation with OPUS-MT, Language Resources and Evaluation (2023) 713–755. doi:10.1007/s10579-023-09704-w.

[5] J. Tiedemann, S. Thottingal, OPUS-MT — Building open translation services for the World, in: Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT), Lisbon, Portugal, 2020.

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[8] L. Parisi, S. Francia, P. Magnani, Umberto: an italian language model trained with whole word masking, https://github.com/musixmatchresearch/umberto, 2020.

[9] 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).

[10] 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.

[11] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, et al., Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.

[12] D. Nozza, Exposing the limits of zero-shot cross-lingual hate speech detection, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Association for Computational Linguistics, Online, 2021, pp. 907–914. URL: https://aclanthology.org/2021.acl-short.114/. doi:10.18653/v1/2021.acl-short.114.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007. doi:10.1109/ICCV.2017.324.

[14] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. URL: https://arxiv.org/abs/1711.05101. arXiv:1711.05101.

[15] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, Q. V. Le, Symbolic discovery of optimization algorithms, in: Proceedings of the 37th International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2023.

[16] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 2019, p. 2623–2631. URL: https://doi.org/10.1145/3292500.3330701. doi:10.1145/3292500.3330701.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: https://proceedings.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[18] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., 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.