Synthetic Data and Machine Learning for Toxic Dynamics Detection in Italian Conversation

Fabio Cirullo

Università degli Studi di Bari Aldo Moro

Abstract

Cyber Intimate Partner Violence (CIPV) is a significant social issue, and its automatic identification from text is not only a complicated task but also a crucial step for online safety. This document is about creating a system to recognize CIPV in Italian dialogues, dealing with the presence/absence of violence as well as with categorization by violent dynamics. One of the challenges was that only violent examples were in the original dataset; hence, to make the binary classification possible, a balanced collection of non-violent conversations was created using a Large Language Model (LLM). After that, multiple text classification models were developed and tested to find out to what extent the synthetic data could be used in CIPV detection and what strategy could be employed as a practical one.

Keywords

CIPV, toxic conversation, LLM, classification, fine-tuning

1. Introduction and Motivations

The spread of digital communication has unfortunately also created new opportunities for abuse within relationships. A serious example is **Cyber Intimate Partner Violence** (CIPV), where harmful behaviors are carried out through text messages, social media, or other online platforms. Recognizing this kind of content is not only a technical challenge, but also an important step toward protecting users.

This project focuses on detecting CIPV in Italian conversations, through two main tasks:

- **Binary Classification:** separating conversations that contain CIPV from those that do not;
- **Multi-class Classification:** identifying which of ten possible toxic dynamics is present (e.g., *Manipulator–Emotionally Dependent*, *Dominant–Emotional Slave*, *Jealous–Obsessive–Submissive*).

To tackle these tasks, I used Transformer-based models. One of the main difficulties was the lack of non-violent data, which I addressed by generating synthetic dialogues with a Large Language Model (LLM). For the multi-class task, particular attention was given to keeping the ten toxic categories evenly represented.

2. Related Work

The detection of harmful online content has been a central topic in NLP for many years. The first approaches relied on handcrafted features combined with classical models such as SVMs and Logistic Regression. With time, these methods were overtaken by neural architectures like LSTMs and CNNs, and more recently by Transformer-based models such as BERT [1], which have set new standards in tasks like hate speech, cyberbullying, and abusive language detection [2, 3].

CIPV, however, brings different challenges compared to other forms of online abuse. Unlike public hate speech, it takes place in private conversations and is closely tied to the dynamics between partners and the surrounding context. To my knowledge, no previous study has specifically addressed CIPV detection in Italian.

Another relevant area is data augmentation in low-resource settings. Earlier techniques made use of paraphrasing, while more recent work has turned to Large Language Models (LLMs) to generate synthetic examples [4]. In most cases, these methods are used to rebalance existing datasets. In this project, instead, synthetic data was crucial to build an entirely new negative class (non-violent dialogues), making it possible to approach the binary classification task. This aspect sets the work apart from previous research.

3. Proposed Approach

The approach adopted in this project is organized into several phases:

- **Dataset Analysis:** A preliminary exploration of the dataset was carried out to understand its characteristics.
- **Data Augmentation:** A Large Language Model (LLM) was used to generate the missing non-toxic conversations required for the binary classification task, as well as additional test examples to support a more robust evaluation.
- **Binary Classification:** Traditional machine learning models were implemented and evaluated using classic text representations.
- **Multi-class Classification:** A comparative study was conducted between a baseline model and a fine-tuned Transformer to assess performance on a multi-class task.

3.1. Implementation Details

The project was implemented in Python, making extensive use of libraries for data manipulation, analysis, and machine learning. `pandas` and `NumPy` were used for data handling and numerical operations, while `scikit-learn` was used for the development of traditional machine learning models.

f.cirullo@studenti.uniba.it (F. Cirullo)

For deep learning, particularly Transformer-based architectures, `Transformers` and `PyTorch` were employed. `NLTK` was used for processing tasks, while data visualization was performed with `Matplotlib` and `Seaborn`.

3.2. Dataset Analysis

The analysis starts with the provided dataset, a CSV file containing 1000 Italian conversations. An initial examination of its structure identified key columns, including `conversation` (the dialogue text) and `person_couple` (the multi-class label). This corpus constituted the 'positive' or 'toxic' class. For the multi-class task, the dataset was perfectly balanced: the 1000 conversations were equally distributed across the 10 toxic couple dynamic categories (shown in Table 1), with exactly 100 samples for each.

Table 1
Distribution of the 10 toxic couple dynamics categories.

Category	Examples
Psicopatico e Adulatrice	100
Manipolatore e Dipendente emotiva	100
Persona violenta e Succube	100
Narcisista e Succube	100
Sadico-Crudele e Masochista	100
Perfezionista Critico e Insicura Cronica	100
Vittimista e Crocerossina	100
Dominante e Schiavo emotivo	100
Geloso-Ossessivo e Sottomessa	100
Controllore e Isolata	100

Then, a linguistic analysis was performed to understand the dataset's characteristics. This involved computing the distribution of conversation lengths and extracting the most frequent unigrams, bigrams, and trigrams to identify common linguistic patterns associated with toxic dialogues. The n-gram analysis revealed recurring phrases indicative of psychological abuse or provocation.

Finally, the dataset was converted from CSV to JSON format to simplify its use in the subsequent modeling pipelines. The insights gained from this analysis guided the next step: the generation of non-toxic conversations to complete the dataset for the binary classification task.

3.3. Generation of Non-Toxic Conversations

To address the absence of a negative class, which was essential for the binary classification task, I employed a synthetic data generation strategy. This process was carried out using the Google AI Studio API [5], specifically using the Gemini 2.5 Pro model [6]. The core of the methodology was the development of a detailed prompt designed to generate a total of 1000 non-toxic conversations.

The prompt engineering process involved several key instructions to ensure the quality and consistency of the generated data. The model was instructed to:

- Produce output in a CSV text format with a specified column order.
- Generate conversations in Italian with a turn length similar to the toxic examples, ensuring structural consistency between the two classes.
- Include a unique batch number for each generation run to promote diversity in the output.

A few-shot prompting technique was adopted by providing a small set of toxic conversations from the original dataset: this gave the model a clear template of the conversational format, even as the main instruction was to create dialogues entirely free of toxic dynamics. To maintain high-quality output and avoid repetition, the generation was performed in five batches of 200 conversations each.

This procedure resulted in a balanced binary classification dataset, comprising the 1000 original toxic samples and 1000 synthetic non-toxic samples, which served as the foundation for the subsequent classification experiments.

3.4. Test Set Enhancement

To ensure a more comprehensive evaluation of the models' performance, I added a small number of synthetically generated examples to the original test set with. This process was again carried out using the Gemini Pro model with a few-shot prompting methodology.

The model was instructed to produce the output directly in JSON format, with required data structure specified in the prompt. The generation strategy was adapted to each classification task:

- **Binary classification:** 10 new conversations were generated (5 toxic and 5 non-toxic). The few-shot prompt included 10 examples (5 per class), ensuring the model captured the semantic differences between the two categories.

- **Multi-class classification:** 10 new conversations were generated, one for each of the 10 toxic categories. The prompt contained a representative example for all classes, guiding the model in producing coherent dialogues for every category.

The number of generated samples was intentionally kept limited because the goal was not to extend the dataset, but rather to introduce independent, unseen instances that could act as an external validation. Importantly, these synthetic conversations were never used for training, but only for testing, in order to provide a more reliable evaluation of the models' generalization capabilities.

3.5. Binary Classification

The first modeling phase of the project focused on the binary classification task, i.e., distinguishing between toxic and non-toxic conversations. For this purpose, I adopted a set of traditional machine learning algorithms:

- Logistic Regression
- Support Vector Machine (SVM)
- Multinomial Naive Bayes

Each classifier was evaluated in combination with two text vectorization strategies:

- Bag-of-Words (BoW)
- Term Frequency–Inverse Document Frequency (TF-IDF)

This design resulted in six distinct model–vectorizer configurations.

A preprocessing pipeline was applied: conversations were segmented into individual turns, from which structural artifacts (e.g., speaker names, colons, quotation marks) were removed. The cleaned turns were then concatenated with a custom separator token, preserving the turn-based structure of the dialogue—an essential feature for modeling toxic couple dynamics, as it maintains the identity of distinct speakers.

To study the impact of text normalization, I carried out two experimental settings for each model–vectorizer pair. In the first, a more aggressive normalization was performed, including stopword removal and lemmatization. In the second, these two steps were omitted to obtain a more natural linguistic variety.

For training and evaluation, the dataset was split into 80% training and 20% testing. Hyperparameter optimization was carried out through an exhaustive Grid Search with 5-fold cross-validation on the training data. The search space included the principal parameters of both classifiers and vectorizers, as shown in Table 2.

Table 2
Hyperparameter space explored during Grid Search for binary classification models and vectorizers.

Component	Hyperparameters Tuned
Vectorizers (BoW & TF-IDF)	
ngram_range	(1, 1), (1, 2), (1, 3)
min_df	(1, 2, 5)
Models	
Logistic Regression	C: [0.01, 0.1, 1, 10, 100] penalty: ['l1', 'l2'] solver: ['saga', 'liblinear']
SVM	C: [0.01, 0.1, 1, 10, 100] penalty: ['l1', 'l2']
Multinomial Naive Bayes	alpha: [0.01, 0.1, 1, 10, 100] fit prior: [True, False]

3.6. Multi-class Classification

For the multi-class classification task, which aimed to identify one of the ten toxic couple categories, I carried out a comparative study to understand the impact of different input representations on model performance. Three distinct approaches (shown in Table 3) were implemented, each varying in the conversation segmentation level:

- **Full conversation:** the entire dialogue is treated as a single input sequence, as in the binary task.
- **Individual turns:** each turn of the dialogue is classified independently.
- **Sliding windows:** overlapping windows of three consecutive turns are generated. From a dialogue of N turns, this yields:

$$\{(t_1, t_2, t_3), (t_2, t_3, t_4), \ldots, (t_{N-2}, t_{N-1}, t_N)\}$$

  To preserve speaker identity, explicit tokens (`[SPEAKER1]`, `[SPEAKER2]`) are added at the beginning of each turn. This representation ensures that the model can correctly interpret the speaker sequence, even when a window begins mid-conversation.

For every approach, I compared a baseline model, Logistic Regression with TF-IDF vectorization, against a fine-tuned Transformer model, specifically the Italian BERT variant `dbmdz/bert-base-italian-cased` [7].

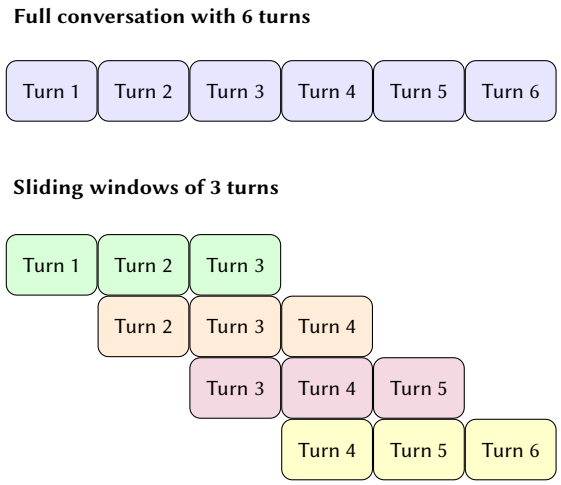

Figure 1: Example of window generation from a 6-turn conversation. Each window contains 3 consecutive turns, producing a total of $N - 2$ sequences.

For all three input strategies, the baseline model reused the hyperparameter definition previously used in the binary task. The primary focus, however, was on fine-tuning the Transformer model. For this process, the dataset was split into training (70%), validation (10%), and test (20%) sets. The model was trained for up to 50 epochs with an *early stopping* mechanism monitoring f1 score. This prevented overfitting by stopping training when no improvement was observed for several consecutive epochs.

4. Evaluation

The purpose of this chapter is to present the results of the evaluation process applied to the proposed models. The assessment was carried out on both the binary and multi-class classification tasks, using the datasets described in the previous chapter. Standard metrics such as accuracy, precision, recall, and f1-score were employed to provide a comprehensive description of model performance.

The chapter is structured as follows: first, the evaluation setup and chosen metrics are described. Subsequently, the results for the binary classification task are reported, followed by the analysis of the multi-class classification experiments. Finally, a comparative discussion highlights the key findings and limitations that emerged from the evaluation.

Table 3
Overview of the three input representation strategies for multi-class classification.

Approach	Description	Key Features
Conversation-level	Entire dialogue treated as a single input sequence	Preserves the full conversational context, but the input can become excessively long
Turn-level	Each turn is treated as an independent input	Highlights local utterances, but loses inter-turn dynamics
Sliding window	Overlapping windows of 3 turns with speaker tokens ([SPEAKER1], [SPEAKER2])	Captures short-range interactions and preserves speaker alternation

4.1. Experimental Setup

The performance of all models was evaluated using standard classification metrics:

- **Accuracy**
- **Precision**
- **Recall**
- **F1-Score**

The F1-Score, defined as the harmonic mean of Precision and Recall, is given by:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

For the multi-class task, the **Macro-F1** score was adopted. This metric computes F1 independently for each class and averages the results, treating all classes equally: a suitable choice given the dataset's balanced distribution. In addition, **Confusion Matrices** are reported in the results section to allow for a more detailed analysis of errors.

The evaluation was conducted on a dedicated test set (20% of the original data). To complement this, each model was also tested on a small set of synthetically generated conversations (10 for the binary task and 10 for the multi-class task), showing how well they can handle data outside the original dataset.

4.2. Binary Classification Results

This section presents the performance of traditional machine learning models on the binary classification task. The complete results for all 12 experimental configurations on the test set are summarized in Table 4. Overall, performance was nearly flawless, with most configurations achieving perfect scores (1.0000) or slightly below (0.9975). This indicates that even simple classifiers, when paired with standard vectorization methods, could effectively separate toxic from non-toxic conversations in the dataset.

A clearer picture emerges when evaluating on the synthetically generated test conversations. In this setting, the impact of preprocessing becomes more evident:

- With stopword removal and lemmatization: all three models correctly classified 6–7 out of 10 examples.
- Without stopword removal and lemmatization: Logistic Regression and SVM correctly classified 8–9 out of 10 examples, while Multinomial Naive Bayes remained at 6–7 correct examples depending on the vectorization method.

These results suggest that synthetic conversations represented a greater challenge due to lexical variations not present in the original dataset. They also highlight that excessive preprocessing can remove useful textual cues, slightly limiting the models' ability to generalize to external data.

4.3. Multi-class Classification Results

This section reports the results of the multi-class classification task, where conversations were categorized into one of ten toxic couple categories. A baseline Logistic Regression model was compared with a fine-tuned BERT architecture across three input representation strategies (see Table 5).

The main findings can be summarized as follows:

- **Impact of input representation:** The sliding window of three consecutive turns clearly outperformed both the full-conversation and single-turn approaches. Short contexts captured interactional dynamics more effectively.
- **Model comparison:** Logistic Regression with sliding windows achieved the best performance, reaching a Macro F1-Score of 0.9632. In contrast, the fine-tuned BERT model in the same configuration reached 0.9000. This indicates that TF-IDF features on short windows were particularly well-suited for linear classification, while BERT struggled with short conversational fragments.
- **Generalization to synthetic data:** On the synthetically generated set of conversations, results dropped. Logistic Regression with sliding windows, despite excelling on the original dataset, correctly classified only 5 out of 10 synthetic examples, reflecting the greater variability of out-of-distribution inputs.

4.4. Discussion and Summary

The evaluation highlights two contrasting scenarios. In the **binary classification task**, traditional models achieved near-perfect performance across all configurations. This suggests that distinguishing toxic from non-toxic conversations was relatively straightforward. The supplementary evaluation on generated dialogues confirmed this behavior, with the best models maintaining robust performance.

The **multi-class classification task**, instead, proved far more challenging. Performance was strongly influenced by the input representation: the sliding window of three turns resulted as the most effective strategy. The Logistic Regression baseline with TF-IDF outperformed

Table 4
Complete results for the binary classification task. Each model and vectorizer pair was tested with and without Stopword removal and Lemmatization (S&L).

Model	Vectorizer	Preprocessing	Accuracy	Precision	Recall	F1-Score
Logistic Regression	TF-IDF	With S&L	1.0000	1.0000	1.0000	1.0000
		Without S&L	0.9975	0.9975	0.9975	0.9975
	Bag-of-Words	With S&L	1.0000	1.0000	1.0000	1.0000
		Without S&L	0.9975	0.9975	0.9975	0.9975
Support Vector Machine	TF-IDF	With S&L	1.0000	1.0000	1.0000	1.0000
		Without S&L	0.9975	0.9975	0.9975	0.9975
	Bag-of-Words	With S&L	0.9975	0.9975	0.9975	0.9975
		Without S&L	0.9975	0.9975	0.9975	0.9975
Multinomial Naive Bayes	TF-IDF	With S&L	0.9975	0.9975	0.9975	0.9975
		Without S&L	1.0000	1.0000	1.0000	1.0000
	Bag-of-Words	With S&L	0.9975	0.9975	0.9975	0.9975
		Without S&L	1.0000	1.0000	1.0000	1.0000

Table 5
Complete comparative results for the multi-class classification task, detailing multiple performance metrics.

	Baseline (LogReg + TF-IDF)				Fine-tuned BERT			
Input Representation	**Acc.**	**Prec.**	**Recall**	**F1**	**Acc.**	**Prec.**	**Recall**	**F1**
Full Conversation	0.7050	0.7317	0.7050	0.7040	0.6700	0.6800	0.6700	0.6700
Single Turn	0.5662	0.5702	0.5662	0.5630	0.5500	0.5600	0.5500	0.5500
Sliding Window (3 turns)	0.9633	0.9642	0.9633	0.9632	0.9000	0.9000	0.9000	0.9000

the fine-tuned BERT model, showing that a simple linear approach can surpass a more complex architecture. However, tests on generated conversations revealed clear limitations in generalization, highlighting difficulties with out-of-distribution inputs.

Overall, the experiments emphasize three key points:

- binary classification can be effectively addressed with standard techniques;
- multi-class classification requires more careful design, with input granularity playing a central role;
- robustness and generalization remain open challenges.

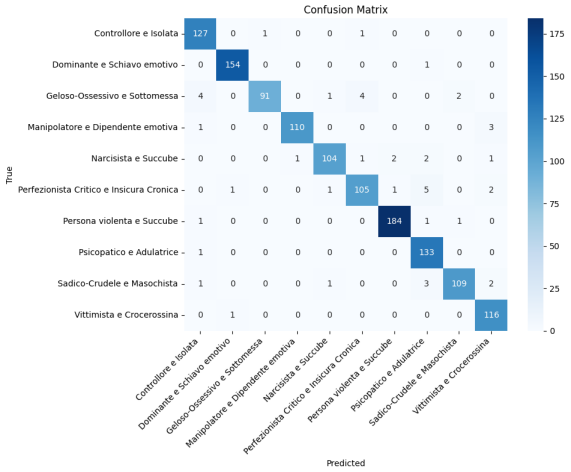

Figure 2: Confusion matrix for the best-performing multi-class model (Logistic Regression with sliding windows).

5. Conclusions and Limitations

This project addressed the automatic detection of Cyber Intimate Partner Violence (CIPV) in Italian conversations, through both a binary task (toxic vs. non-toxic) and a multi-class task with ten toxic dynamics. Traditional machine learning models were compared with a fine-tuned Transformer, exploring different input representations and using a Large Language Model to generate additional data.

Binary classification turned out to be easy to handle: Logistic Regression and SVM achieved near-perfect results, due to the clear separation between toxic and synthetically generated non-toxic dialogues. The multi-class task was more difficult, with performance strongly influenced by input representation. Sliding windows of turns consistently outperformed other strategies, and Logistic Regression with TF-IDF surpassed BERT, showing the effectiveness of linear models in this setting.

However, the study faces limitations. The use of a curated dataset and synthetic negative class reduced task complexity, while tests on generated dialogues revealed weaker generalization, especially in the multi-class case. Perhaps, models struggled with phenomena requiring deeper contextual understanding, such as sarcasm or implicit aggression.

Future work should test advanced Transformer ar-

chitectures on sliding-window inputs, validate models on real annotated data, explore hybrid approaches, and experiment with alternative vectorization strategies in combination with baseline models, such as word embeddings. Overall, the project illustrates both the potential and the limits of computational methods for detecting toxic dynamics, where the complexity of language and context continues to pose difficulties.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018). arXiv:1810.04805.

[2] P. Mishra, H. Yannakoudakis, E. Shutova, Tackling online abuse: A survey of automated abuse detection methods (2019). arXiv:1908.06024.

[3] Z. Waseem, T. Davidson, D. Warmsley, I. Weber, Understanding abuse: A typology of abusive language detection subtasks (2017). arXiv:1705.09899.

[4] V. Kumar, A. Choudhary, K. Passi, Data Augmentation Using Pre-trained Transformer Models (2020). arXiv:2003.02245.

[5] Google, Google ai studio, https://ai.google.dev/aistudio, accessed: 2025-08-19 (2024).

[6] G. T. et al., Gemini: A family of highly capable multimodal models (2023). arXiv:2312.11805.

[7] W. Trost, Others, Italian bert models (dbmdz/bert-base-italian-cased), https://huggingface.co/dbmdz/bert-base-italian-cased (2019).