Toxicity, couple dynamics and most toxic sentence detection in Italian conversations using classical machine learning models and transformer-based models

Davide Cirilli
Università degli Studi di Bari Aldo Moro
Email: d.cirilli2@studenti.uniba.it

***Abstract*—This paper presents a comprehensive study on the detection of toxicity and personality-driven behaviors in Italian conversational data. We developed a system capable of classifying conversations as toxic or non-toxic, analyzing couple dynamics, and identifying or generating the most toxic sentences within a dialogue. Our approach combines traditional machine learning techniques, such as Logistic Regression and Naive Bayes, with advanced transformer-based models like BERT and BART. We evaluated these methods across different tasks, highlighting the strengths of fine-tuned transformer models in capturing subtle and complex patterns of toxicity. The system not only detects harmful interactions but also provides insights into personality-driven behaviors that may underlie toxic exchanges. To support practical use, we created a web application that offers a user-friendly interface for real-time toxicity detection, personality analysis, and the classification or generation of the most toxic sentence in a conversation. The study demonstrates the value of integrating classic and modern machine learning approaches to better understand and address toxic communication in digital spaces.**

I. Introduction and Motivations

In todays digital society, online platforms and social networks have become essential spaces for communication, shaping relationships and social interactions on a global scale. While these platforms offer numerous opportunities for connection and expression, they also create environments where toxic behaviors can spread, often with significant psychological and social consequences. This is especially true in conversations within intimate or emotionally charged contexts, where harmful behaviors may take the form of subtle manipulation, emotional control, or psychological abuse, rather than overtly offensive language.

Detecting and addressing such forms of toxicity is crucial for fostering safer and healthier digital environments. However, traditional approaches that focus on isolated messages or specific keywords often fall short in recognizing the complex dynamics that characterize toxic interactions. In particular, the interplay between conversational context and personality-driven behaviors demands more advanced tools capable of understanding deeper patterns of communication.

Our work aims to contribute to this challenge by exploring systems that combine toxicity detection with personality analysis, offering a more comprehensive view of harmful interactions in online dialogues. Beyond simply identifying toxic messages, these tasks play a vital role in supporting moderation efforts, promoting digital well-being, and informing interventions that can help prevent the escalation of harmful relationships in virtual spaces.

II. Related Work

The detection of toxicity and abusive language in online conversations has been widely explored, especially in English. Early approaches relied on machine learning classifiers such as SVM and Naive Bayes with hand-crafted features (e.g., n-grams, lexicons of offensive terms) to identify hate speech and offensive content [1] [2]. More recent work leverages transformer-based models like BERT [3] and RoBERTa [4], showing significant improvements over traditional methods, particularly in capturing implicit and contextual toxicity [5]. Studies like [6] highlight the importance of conversational context, noting that some messages are perceived as toxic only in relation to previous dialogue turns.

In the Italian language, projects such as EVALITA [7] provided annotated datasets for hate speech detection, mostly focused on Twitter data. These efforts demonstrated the potential of both traditional models and fine-tuned multilingual transformers (e.g., XLM-R) for Italian toxicity detection [8].

For personality detection, initial studies applied linguistic and psychological features (e.g., LIWC categories, stylistic markers) with classifiers like SVM to predict Big Five traits [9] [10]. More recent approaches incorporate deep learning, using RNNs, CNNs, and transformer embeddings to infer personality traits from social media text [11]. While many studies target English data, personality detection in Italian conversational data remains largely underexplored.

Our work extends this literature by addressing both toxicity and personality analysis simultaneously in Italian dialogues, leveraging transformer models to capture conversational context and personality-driven toxic behaviors in a unified system.

III. Dataset preprocessing and Data augmentation

In this section, we describe the dataset construction process, which involved integrating an existing toxic conversation dataset with a newly generated non-toxic conversation dataset. The goal was to create a balanced corpus that captures both toxic and healthy relationship dynamics in Italian conversations, enabling effective training and evaluation of toxicity detection models.

A. Dataset preprocessing

First of all, we applied a preprocessing step to the original dataset containing toxic conversations. The preprocessing steps were as follows:

- **Standardization and cleaning:** We removed incomplete rows with missing names or conversation fields and excluded entries with irrelevant or malformed data. Names of participants were standardized to title case to ensure consistency.
- **Extraction of toxic sentences:** From the explanation field accompanying each conversation, we programmatically extracted the most toxic sentence using regular expressions. If no toxic sentence was present, we marked it as "N/A".
- **Conversation reformatting:** We reformatted conversations using a hierarchical parsing logic. The system first tried to extract messages enclosed in quotation marks. If unavailable, it attempted to split the text using speaker labels (e.g., `Name:`) or speaker names. This aimed to segment and clean the dialogues effectively.
- **Whitespace and symbol normalization:** Excessive spaces and inconsistent symbols (e.g., different types of quotation marks or angle brackets) were normalized to improve text uniformity and facilitate further processing.
- **Final selection and export:** After cleaning, we retained the essential columns — including participant names, cleaned conversation, extracted toxic sentence, and explanation — and saved the resulting dataset in CSV format for subsequent use.

This structured preprocessing pipeline ensured that the data fed to our models was clean, consistently formatted, and suitable for both toxicity detection and personality analysis tasks.

B. Dataset augmentation

To enhance the dataset, we generated a complementary non-toxic conversation dataset using both Google's Gemini API and Google AI Studio. This automated generation process aimed to create realistic and culturally appropriate non-toxic dialogues in Italian, ensuring diversity in conversational patterns while adhering to safety guidelines.

1) Automated generation with Gemini API: The Gemini API was used to generate a large portion of the non-toxic conversation dataset. The model selected was `gemini-2.0-flash-lite`, configured with parameters designed to balance creativity and coherence: temperature was set to 1.8 to promote diverse outputs; `top_p` was set to 1.0 and `top_k` to 0 to allow unrestricted token sampling. Strict safety settings were applied to block inappropriate content, with thresholds set to `BLOCK_MEDIUM_AND_ABOVE` for all categories: harassment, hate speech, sexually explicit content, and dangerous content.

To guide generation, prompts specified role-based dynamics for each dialogue. The `person_couple` classes used were:

- Insicuro e Supportivo
- Propositivo e Collaborativo
- Vulnerabile e Accogliente
- Pentito e Comprensivo
- Grato e Apprezzante

For each sample, a pair was selected randomly, and the API was instructed to generate a 6–10 turn dialogue reflecting the assigned dynamic, along with unique Italian names for the participants.

Postprocessing involved parsing the raw response into structured fields: names, role pair, and dialogue. A custom parser extracted the names and cleaned each dialogue turn by removing speaker prefixes and ensuring proper punctuation and casing. Conversations not adhering to the required format (e.g., missing separator, insufficient turns, incomplete lines) were discarded. The data was incrementally written to CSV in append mode, ensuring that header rows were only written once. After generation, additional validation removed duplicates and normalized name casing, while ensuring that dialogue texts contained no residual formatting artifacts or repeated utterances.

2) Generation with Google AI Studio: Google AI Studio was used to complement the API-generated dataset, following the same parameter configuration as Gemini (temperature = 1.8, `top_p` = 1.0, `top_k` = 0, strict safety thresholds). The platform enabled batch generation with direct control over the response format. Each generated sample followed a predefined schema including:

- `person_couple` class (randomly selected from the same list as above)
- two invented Italian names
- a complete, coherent dialogue
- `most_toxic_phrase` field always set to N/A
- `toxic` label fixed at 0

A key strength of this approach was the high structural consistency of the generated data, which minimized the need for parsing and reduced the risk of formatting errors. Furthermore, the output included explanations of why

each conversation was classified as non-toxic, enhancing the datasets annotation richness.

3) Postprocessing and integration: Once generated, all data (from both sources) underwent additional processing. Conversations were checked for format consistency, proper turn alternation, and minimal length. Names were normalized (first letter uppercase, no trailing spaces), dialogues were cleaned of redundant whitespace, and duplicate conversations were removed. The final augmented dataset was balanced to ensure an even distribution between toxic and non-toxic samples, thereby strengthening model training by preventing bias toward any class.

At the end of this process, we had a comprehensive dataset containing around 950 toxic conversations and approximately 600 non-toxic conversations, each with detailed annotations. This dataset serves as a robust foundation for training and evaluating toxicity detection models, enabling nuanced analysis of conversational dynamics and personality-driven behaviors.

IV. Approach

In this section, we present our approach to toxicity detection, personality classification and most toxic sentence classification or generation in Italian conversational data. Our methodology integrates traditional machine learning techniques with advanced transformer-based models, specifically BERT and BART, to achieve robust performance across multiple tasks. The system is designed to handle both binary toxicity classification and personality-driven behavior analysis, providing a comprehensive solution for understanding toxic interactions in digital conversations.

A. Binary Toxicity Classification

1) Without Text Preprocessing: For the binary toxicity classification task, we first trained traditional machine learning models directly on raw conversational text. The dataset was split into training and test sets using an 80-20 stratified split to preserve class balance. We implemented two pipelines combining `TfidfVectorizer` with either Logistic Regression or Multinomial Naive Bayes classifiers. The `TfidfVectorizer` was configured with a bi-gram range (1,2) to capture both individual tokens and short phrases.

Hyperparameter tuning was performed through grid search with 5-fold stratified cross-validation. For Logistic Regression, we optimized the regularization parameter C and the maximum number of features (3000, 5000, 7000). For Naive Bayes, we adjusted the smoothing parameter α along with the number of features. The best configurations were selected based on weighted f1 measure. Both models achieved high accuracy on the test set, confirming that traditional methods remain effective when combined with strong text representations like TF-IDF, even without preprocessing.

2) With Text Preprocessing: To assess the impact of linguistic normalization, we applied additional preprocessing using the Italian `spaCy` model (`it_core_news_sm`). Each conversation was lowercased, lemmatized, and stripped of stop words, punctuation, and spaces. The resulting text consisted of lemmatized content words, aiming to reduce noise and improve generalization.

We repeated the training and evaluation pipeline using these processed texts. The same classifiers and vectorization settings were used, although the grid search parameter space for max features was slightly reduced (2000, 4000, 6000) due to the more compact vocabulary after preprocessing. The results were comparable to the unprocessed case, with minor variations in precision and recall. This indicates that, while preprocessing provides cleaner input, the models already capture much of the discriminative information through n-gram patterns and TF-IDF weighting.

B. Couple Dynamics Prediction

The couple dynamics prediction task aimed at classifying conversational excerpts according to the type of dyadic interaction, encoded in the `person_couple` field of our dataset. Three distinct modeling strategies were explored: (i) a logistic regression model with Latent Semantic Analysis (LSA), (ii) logistic regression on frozen BERT embeddings, and (iii) a fine-tuned BERT model. Below we describe each method in detail.

1) Logistic Regression with LSA: We applied a pipeline consisting of TF-IDF vectorization, dimensionality reduction via Latent Semantic Analysis, and classification using logistic regression. The main components were:

- **TF-IDF Vectorizer**: Extracted unigrams and bigrams to represent the conversations.
- **TruncatedSVD**: Reduced the feature space to latent topics, with the number of components optimized in {100, 200, 300}.
- **Logistic Regression**: Employed `liblinear` solver and regularization parameter $C \in \{0.1, 1, 10\}$.

Preprocessing was performed using `spaCy` (`it_core_news_sm`), involving lowercasing, lemmatization, and removal of stopwords and punctuation. A grid search with 5-fold stratified cross-validation identified the best combination of latent components and regularization strength. The final model was evaluated on a held-out test set, with accuracy and confusion matrix analyses performed.

2) Logistic Regression on Frozen BERT Embeddings: In this variant, we leveraged the pre-trained Italian BERT model (`dbmdz/bert-base-italian-uncased`) without any fine-tuning. Each conversation was encoded as the `[CLS]` token embedding from BERT's last hidden layer:

- **Tokenization**: Handled via `BertTokenizer` with truncation and padding to a maximum of 512 tokens.

- **Embedding Extraction**: The `last_hidden_state[:, 0, :]` vector was extracted for each sample.

These fixed-size embeddings served as input to a logistic regression classifier with $C = 0.1$ and maximum 2000 iterations. No additional text preprocessing was applied beyond BERT's built-in tokenization. This approach evaluated how well pre-trained contextual representations capture couple dynamics without task-specific adaptation.

3) Fine-Tuned BERT: The final method involved fine-tuning `dbmdz/bert-base-italian-uncased` directly on the classification task:

- **Input Pipeline**: Used `BertTokenizer` with padding, truncation, and max length 512. Data was split into train, validation, and test sets (80/10/10 stratified).
- **Model**: Employed `BertForSequenceClassification` with label mappings derived from the `person_couple` classes.
- **Training Configuration**: Set for up to 7 epochs, batch size 8, Adam optimizer with weight decay, and learning rate warmup. Early stopping with patience of 2 epochs was applied to prevent overfitting.
- **Trainer API**: Used HuggingFace `Trainer`, saving the best model based on validation loss.

Post-training, we plotted loss curves to visualize learning dynamics and assessed the model on the test set via accuracy, classification report, and confusion matrix.

4) Post-Processing and Analysis: For all approaches, confusion matrices were generated to inspect misclassifications. Class labels were decoded using `LabelEncoder`, ensuring consistency across models. In the case of the BERT fine-tuned model, training and validation losses were additionally plotted to analyze convergence behavior.

The classes considered in this task were all those present in the `person_couple` field of the dataset (with both toxic and non-toxic conversations). The classes included:

- Controllore E Isolata
- Dominante E Schiavo Emotivo
- Geloso-Ossessivo E Sottomessa
- Grato e Apprezzante
- Insicuro e Supportivo
- Manipolatore E Dipendente Emotiva
- Narcisista E Succube
- Pentito e Comprensivo
- Perfezionista Critico E Insicura Cronica
- Persona Violenta E Succube
- Propositivo e Collaborativo
- Psicopatico E Adulatrice
- Sadico-Crudele E Masochista
- Vittimista E Croccerossina
- Vulnerabile e Accogliente

This multi-model setup allowed us to compare traditional and transformer-based strategies in predicting conversational dynamics in Italian conversations.

C. Most Toxic Sentence Detection

The task of Most Toxic Phrase Detection focuses on identifying or generating the most toxic sentence within a conversation. We investigated two distinct approaches: (1) toxic phrase classification via fine-tuned BERT, and (2) toxic phrase generation using a BART-based seq2seq model.

1) Most Toxic Phrase Classification with BERT: For the classification approach, conversations were preprocessed to extract individual messages. Each message was paired with its corresponding couple identifier (as additional context) and labeled as toxic (1) if it matched the annotated most toxic sentence, or non-toxic (0) otherwise.

We employed the `dbmdz/bert-base-italian-uncased` model for fine-tuning. Inputs consisted of the message and couple identifier, tokenized using a maximum length of 128 tokens. The model was trained for up to 7 epochs with early stopping (patience = 2 epochs) based on validation loss. The training process used a batch size of 32, a learning rate of 2×10^{-5}, weight decay of 0.01, and the AdamW optimizer as implemented in Huggingface's `Trainer`.

The dataset was stratified and split into 80% training, 10% validation, and 10% test sets. Model evaluation on the test set involved computing accuracy, precision, recall, F1-score, and analyzing the confusion matrix.

At inference time, given a conversation, the model classifies each message independently and the one with the highest predicted probability of being toxic is selected as the most toxic sentence.

2) Toxic Phrase Generation with BART: In the generation-based approach, we framed the task as sequence-to-sequence learning: the input sequence consisted of the entire conversation (prefixed by a task descriptor), and the output was the annotated most toxic sentence.

We used the `facebook/bart-base` model with a maximum input length of 512 tokens and output sequences truncated or padded to a maximum of 64 tokens. Tokenization and training were performed using Huggingface's `Trainer` API with a `DataCollatorForSeq2Seq`.

The dataset was split using `GroupShuffleSplit` to ensure that train, validation, and test partitions (80% / 10% / 10%) did not share structurally similar conversations, minimizing the risk of data leakage. Training employed early stopping (patience = 2 epochs), 7 epochs maximum, learning rate 3×10^{-5}, and a batch size of 4 due to memory constraints with large sequence inputs.

During inference, generation was performed using beam search (num_beams = 4) with early stopping enabled. The generated sentence was evaluated against the reference using BLEU and ROUGE metrics.

3) Evaluation Protocol: Both models were evaluated on held-out test sets using metrics appropriate to their respective tasks.

a) Classification Metrics: For the classification-based model, we computed standard classification metrics: *accu-*

racy, *precision*, *recall*, and *F1-score*. Furthermore, confusion matrices were plotted to provide a detailed view of misclassification patterns across classes.

b) Generation Metrics: For the generation-based model, we employed BLEU and ROUGE metrics to quantitatively evaluate the similarity between the generated toxic sentence and the reference toxic sentence.

BLEU (Bilingual Evaluation Understudy) [12] measures the degree of n-gram overlap between the generated output and the reference text. Specifically, it computes precision over n-grams (where $n = 1, 2, \dots$), penalized by a brevity penalty to discourage overly short outputs. In this work, we computed sentence-level BLEU scores with smoothing to mitigate the problem of zero scores when no n-gram matches are found in short sentences.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [13] is a suite of metrics designed to evaluate text summarization and generation quality. We report three commonly used ROUGE variants:

- **ROUGE-1:** Measures the overlap of unigrams (individual words) between the generated and reference sentence. It reflects how many words of the reference are captured by the output.
- **ROUGE-2:** Measures the overlap of bigrams (sequences of two consecutive words). This provides a stronger indication of fluency and local coherence.
- **ROUGE-L:** Based on the longest common subsequence (LCS), this metric captures the extent to which the generated output preserves the sequence and order of words found in the reference.

For ROUGE, we reported the *F1* measure, which balances precision and recall at the n-gram level.

All metrics were averaged over the test set to provide a global estimate of model performance. The use of both BLEU and ROUGE allowed us to evaluate the generated text along complementary dimensions: BLEU focusing more on precision of n-gram matches, and ROUGE emphasizing recall and sequence fidelity.

D. Web Application for Real-Time Couple Dynamics and Most Toxic Phrase Detection

To facilitate interactive experimentation and demonstration of the proposed models, we developed a web application using the `Gradio` framework. The application integrates three distinct transformer-based components: a couple dynamics classifier, a toxic sentence classifier, and a toxic sentence generator.

1) Architecture and Components:

- **Couple dynamics classification:** The application uses a fine-tuned BERT model (`dbmdz/bert-base-italian-uncased`) to classify the overall interaction pattern between two individuals based on their dialogue history. The classifier outputs one of 15 predefined relationship dynamics labels (e.g., "Narcisista E Succube"). The model produces class probabilities via a softmax layer, from which the most probable label and its confidence score are extracted.
- **Toxic sentence classification:** A second fine-tuned BERT model identifies the most toxic sentence within the conversation. Each message is encoded together with the predicted couple dynamics as context. The model assigns a toxicity probability to each message, and the sentence with the highest probability is selected.
- **Toxic sentence generation:** A fine-tuned BART model (`facebook/bart-base`) generates a toxic sentence summarizing the overall harmful content of the conversation. The model uses a prefix prompt ("Conversation: ...") and generates an output sequence using beam search decoding (4 beams, early stopping, maximum 64 tokens).

2) Inference Workflow: Users interact with the system by submitting individual chat messages. The application maintains conversation history and updates the couple dynamics prediction after each input. When requested, the application:

1) Classifies the most toxic sentence by evaluating each message in the history.
2) Generates a toxic sentence summarization using the BART model.

The inference process is performed in *evaluation mode* with disabled gradient computation to reduce memory usage and ensure efficient execution.

3) User Interface: The interface provides:

- A chat-like component for sequential input and feedback.
- A button to trigger extraction of both the classified and generated toxic sentences.
- A reset button to clear the conversation and results.

4) Implementation Details: All models were loaded in evaluation mode and run on GPU if available. Tokenization is performed dynamically at inference time. The BERT-based components use a maximum input length of 512 tokens (for couple dynamics) and 128 tokens (for toxic sentence classification), with appropriate truncation and padding strategies. The BART generator input is similarly truncated at 512 tokens to fit within the model's architecture limits.

The system provides both the predicted couple dynamics (including a binary toxic/not-toxic indication based on predefined labels) and the toxic sentence analyses, allowing direct comparison between classification-based and generation-based outputs.

V. Results

In this section, we present the results of our experiments on toxicity detection, couple dynamics classification, and most toxic sentence detection in Italian conversational data.

A. Binary Toxicity Classification

The binary toxicity classification task was evaluated using two classical machine learning models: Logistic Regression and Multinomial Naive Bayes. Both were tested under two configurations: without text preprocessing, and with spaCy-based preprocessing (tokenization, lemmatization, stopword and punctuation removal).

1) Hyperparameter Tuning:

- **Without preprocessing:**
  - **Logistic Regression:** $C = 10$, TFIDF max_features = 3000.
  - **Multinomial Naive Bayes:** $\alpha = 0.1$, TFIDF max_features = 5000.
- **With spaCy preprocessing:**
  - **Logistic Regression:** $C = 10$, TFIDF max_features = 2000.
  - **Multinomial Naive Bayes:** $\alpha = 0.1$, TFIDF max_features = 6000.

2) Performance Metrics:

a) Without preprocessing:

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.9968	0.9968	0.9968	0.9968
Multinomial NB	0.9968	0.9968	0.9968	0.9968

Table I: Binary classification results without preprocessing

b) With spaCy preprocessing:

Model	Accuracy	Precision	Recall	F1
Log Reg (spaCy)	0.9936	0.9936	0.9936	0.9936
Multi NB (spaCy)	0.9968	0.9968	0.9968	0.9968

Table II: Binary classification results with spaCy preprocessing

3) Confusion Matrices:

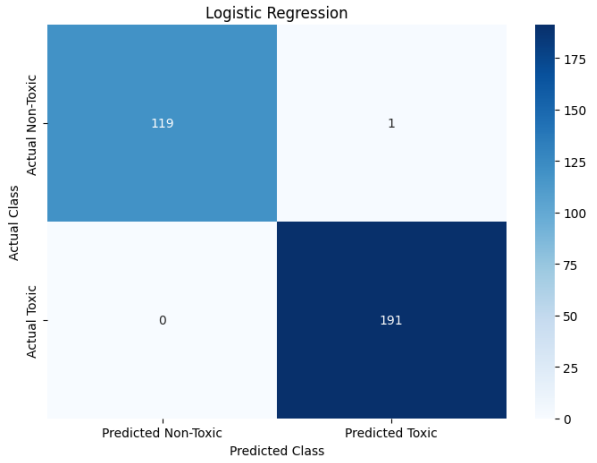

Figure 1: Confusion matrix: Logistic Regression without preprocessing

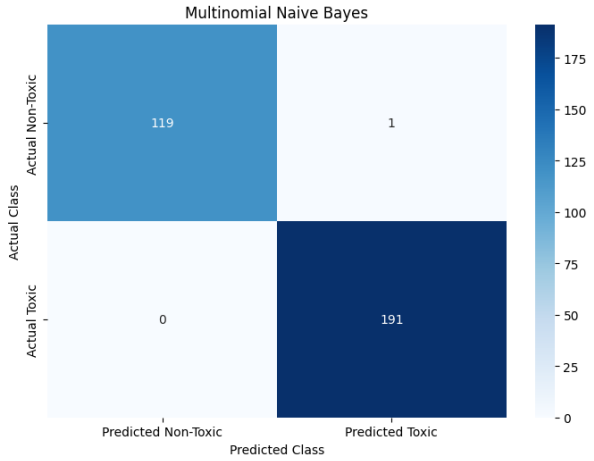

Figure 2: Confusion matrix: Multinomial Naive Bayes without preprocessing

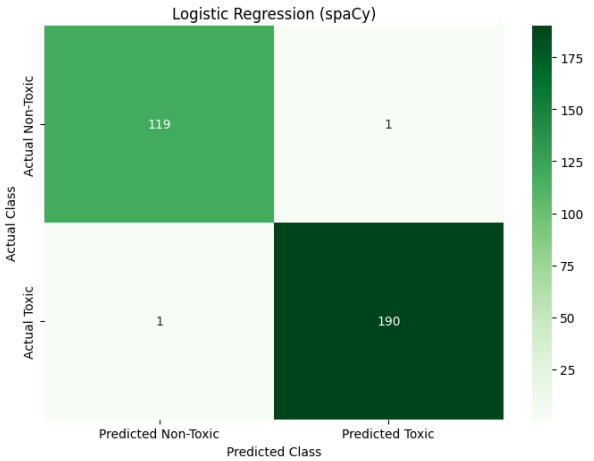

Figure 3: Confusion matrix: Logistic Regression with spaCy preprocessing

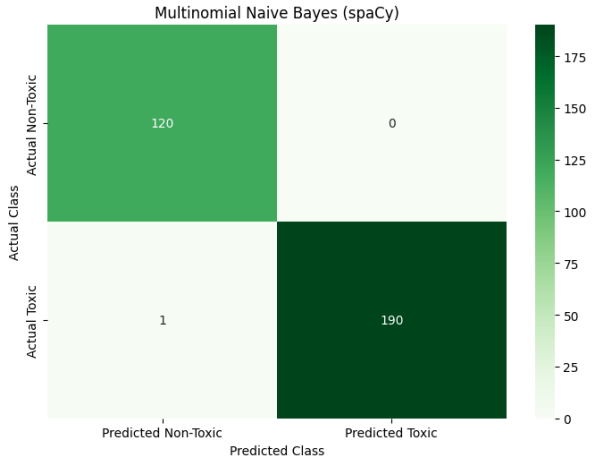

Figure 4: Confusion matrix: Multinomial Naive Bayes with spaCy preprocessing

4) Discussion: Both models achieved near-perfect performance, indicating strong separability between toxic and non-toxic samples. SpaCy preprocessing slightly reduced performance in logistic regression (from 0.9968 to 0.9936), likely due to the removal of informative lexical features. Multinomial Naive Bayes showed robustness across both configurations. These results suggest that simple ML models paired with TFIDF can be highly effective in this binary classification task when sufficient data is available.

B. Couple Dynamics Classification

In this section, we compare the performance of three approaches applied to the couple dynamics classification task: (i) LSA + Logistic Regression, (ii) BERT embeddings with Logistic Regression, and (iii) fine-tuned BERT. Their metrics are summarized in the tables above.

1) LSA + Logistic Regression:

The LSA + Logistic Regression pipeline achieved an accuracy of 73.95% with macro precision, recall, and F1 scores of 0.72, 0.72, and 0.71 respectively. This approach shows reasonable performance, particularly for classes with well-separated semantics (e.g., Grato e Apprezzante, Pentito e Comprensivo, Propositivo e Collaborativo where precision and recall reached or neared 1.00). However, for more subtle or overlapping dynamics (e.g., Sadico-Crudele E Masochista, Geloso-Ossessivo E Sottomessa), both precision and recall were significantly lower. This highlights the limitations of linear models on reduced-dimensional semantic spaces in handling complex emotional contexts.

Metric (Macro)	Accuracy	Precision	Recall	F1
Value	0.7395	0.7233	0.7160	0.7113

Table III: Performance of LSA + Logistic Regression on test set

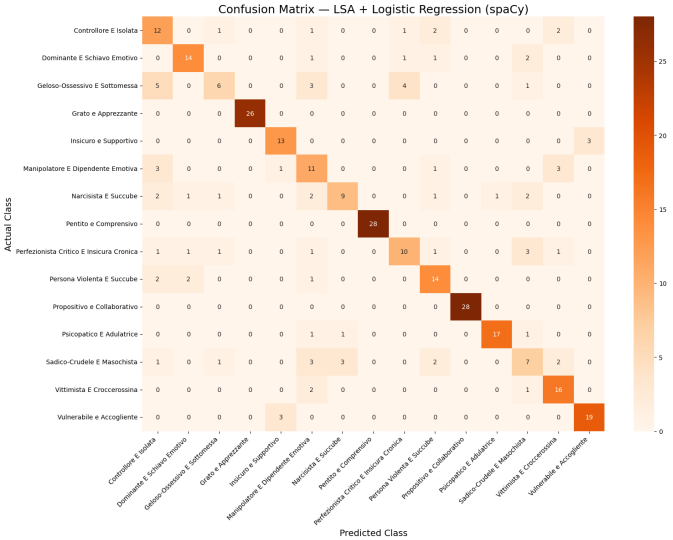

Figure 5: Confusion matrix for LSA + Logistic Regression

2) BERT Embeddings + Logistic Regression:

Using pre-trained BERT embeddings as features yielded slightly lower performance with 70.10% accuracy and macro F1 of 0.67. While BERT embeddings capture richer semantics than LSA, the lack of task-specific fine-tuning likely limited their effectiveness in this multi-class setting. Some classes (Pentito e Comprensivo, Propositivo e Collaborativo) continued to be classified accurately, but the model struggled with more nuanced or less frequent dynamics, suggesting that the embeddings alone were not sufficiently discriminative without adaptation.

Metric (Macro)	Accuracy	Precision	Recall	F1
Value	0.7010	0.6840	0.6767	0.6753

Table IV: Performance of BERT embeddings + Logistic Regression on test set

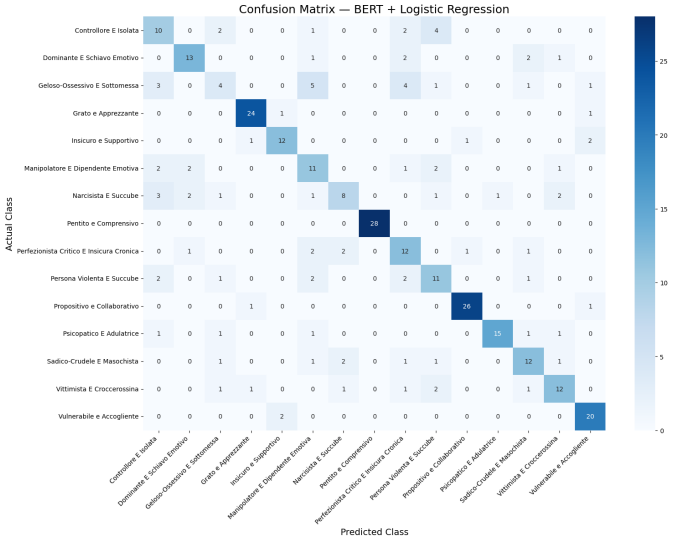

Figure 6: Confusion matrix for BERT embeddings + Logistic Regression

3) Fine-tuned BERT:

The fine-tuned BERT model outperformed both previous approaches, achieving 80.13% accuracy with a macro F1 of 0.78. As shown in the loss curves (Figure 7), the model exhibited good convergence behavior with no significant overfitting. The confusion matrix (Figure 8) further illustrates that this model provided more balanced predictions across most classes, including those that were difficult for previous approaches (e.g., Sadico-Crudele E Masochista, Vittimista E Croccerossina). The fine-tuning phase allowed BERT to specialize its representations to the nuances of couple dynamics, leading to more robust classification.

Metric (Macro)	Accuracy	Precision	Recall	F1
Value	0.8013	0.8093	0.7867	0.7780

Table V: Performance of fine-tuned BERT on test set

Figure 7: Training and validation loss curves for fine-tuned BERT

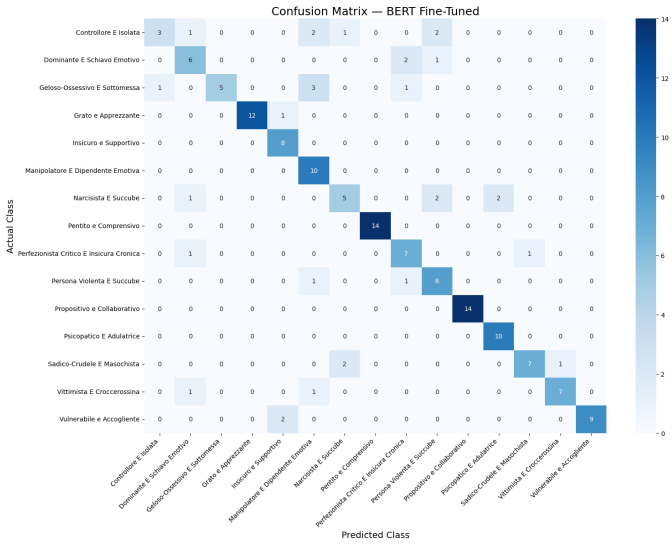

Figure 8: Confusion matrix for fine-tuned BERT

C. Most Toxic Sentence Detection

The most toxic sentence detection task was evaluated using two distinct approaches: (i) classification of sentences using a fine-tuned BERT model, and (ii) generation of toxic sentences using a BART-based seq2seq model. Below we present the results for each method.

1) Most Toxic Sentence Classification with BERT:

The BERT-based classifier achieved a test set accuracy of 87.40%. While the overall accuracy and weighted metrics are high (weighted F1: 0.86), a detailed analysis reveals a significant imbalance in class-level performance. The classifier reached high precision (0.91) and recall (0.96) for the Non-Toxic class, but struggled considerably with the Toxic class, yielding a precision of 0.36 and recall of 0.20. The low recall indicates that the model missed a large proportion of toxic phrases, likely due to the imbalance in the dataset where non-toxic sentences dominate. This suggests that despite fine-tuning, BERT had difficulty in detecting the minority class in a highly imbalanced setting.

Table VI: Performance metrics for BERT toxic phrase classification

Class	Precision	Recall	F1-score	Support
Non-Toxic	0.9077	0.9558	0.9311	679
Toxic	0.3617	0.2048	0.2615	83
Accuracy		0.8740		
Macro avg	0.6347	0.5803	0.5963	762
Weighted avg	0.8482	0.8740	0.8582	762

As depicted in the loss curves (Figure 9), the model initially achieved a good reduction in both training and validation loss. However, after the first few epochs, the validation loss began to increase while the training loss remained relatively stable, suggesting the onset of overfitting. Future work might explore rebalancing strategies, stronger regularization, or alternative loss functions (e.g., focal loss) to better handle the class imbalance and improve generalization. The confusion matrix (Figure 10) further illustrates this imbalance, with a large number of false negatives in the *Toxic* class.

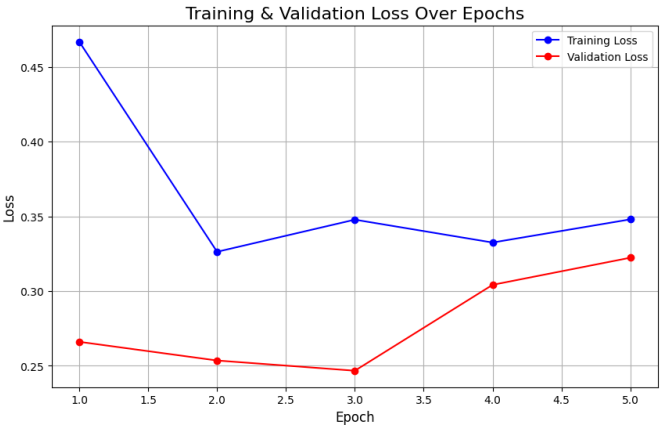

Figure 9: Training and validation loss curves for BERT toxic phrase classification

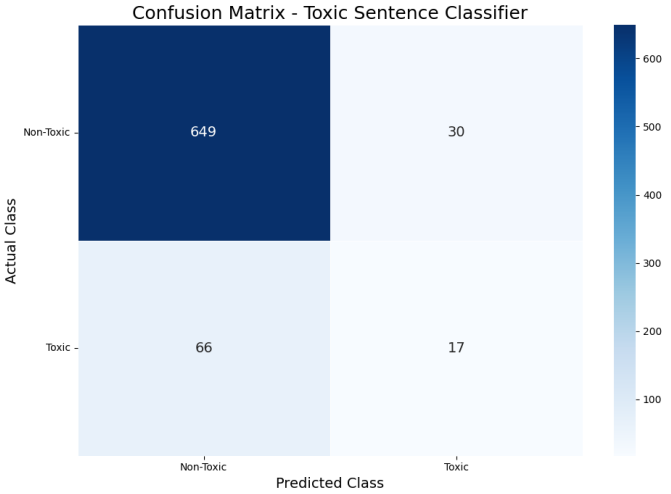

Figure 10: Confusion matrix for BERT toxic phrase classification

2) Most Toxic Sentence Generation with BART:

For generation, we evaluated the BART model using BLEU and ROUGE metrics to assess the similarity between generated toxic phrases and human-labeled toxic phrases. The model achieved an average BLEU score of 0.17, ROUGE-1 F-measure of 0.30, ROUGE-2 F-measure of 0.22, and ROUGE-L F-measure of 0.30. These results indicate that the generated phrases share some lexical overlap with reference toxic phrases, but the relatively low BLEU score reflects the challenge of generating exact matches in open-ended, linguistically varied tasks such as toxic phrase synthesis. The ROUGE F-measure scores suggest that BART captured some key n-grams and longer subsequences of reference phrases, yet there remains substantial room for improving fluency and semantic alignment.

Table VII: Performance metrics for BART toxic phrase generation

Metric	Score
BLEU	0.1706
ROUGE-1	0.3031
ROUGE-2	0.2170
ROUGE-L	0.2987

As seen in the loss curves (Figure 11), the model's training dynamics show a steady decrease in training loss across epochs, while the validation loss remains low and relatively stable. This indicates that the model fits the training data progressively better without significant overfitting. However, more sophisticated decoding strategies (e.g., nucleus sampling, top-k sampling) could potentially further enhance the generation quality.

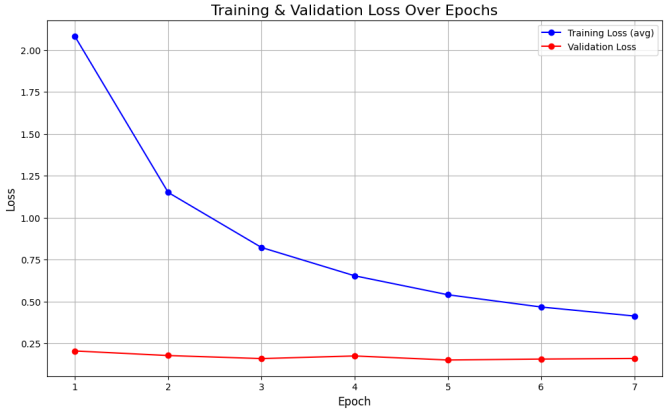

Figure 11: Training and validation loss curves for BART toxic phrase generation

VI. Conclusion

In this report, we presented a comprehensive approach to toxicity detection, couple dynamics analysis and most toxic sentence detection in Italian conversational data. Our methodology combined traditional machine learning techniques with advanced transformer-based models, specifically BERT and BART, to achieve robust performance across multiple tasks.

We successfully generated a synthetic dataset of Italian conversations, which was used to train and evaluate our models. The binary toxicity classification task demonstrated that classical machine learning models, such as Logistic Regression and Multinomial Naive Bayes, can achieve high accuracy even without extensive text preprocessing.

The couple dynamics classification task highlighted the effectiveness of fine-tuned BERT models in capturing complex relational patterns, achieving an accuracy of 80.13%.

The most toxic sentence detection task showcased the strengths and limitations of both classification and generation approaches. The BERT-based classifier achieved high accuracy but struggled with class imbalance, while the BART generator produced reasonable toxic phrases with some lexical overlap to human-labeled references.

Finally, we developed a web application using the Gradio framework to facilitate interactive experimentation and demonstration of our models. This application allows users to explore couple dynamics and toxic sentence detection in real-time, providing a user-friendly interface for understanding conversational toxicity.

VII. Limitations and Future Work

A. Current Limitations

While our approach has demonstrated promising results, several limitations remain:

- **Class Imbalance:** The binary toxicity classification and most toxic sentence detection tasks faced significant class imbalance, which affected model performance. Future work should explore techniques such

as oversampling, undersampling, or using focal loss to mitigate this issue.

- **Interpretability:** The black-box nature of transformer models makes it challenging to interpret their decisions. Future work should explore techniques for model interpretability, such as attention visualization or feature attribution methods, to better understand the factors driving toxicity predictions.
- **Real-time Performance:** The web application, while functional, may face performance issues with larger conversations or more complex models. Future work should optimize the inference pipeline and consider model distillation or quantization techniques to improve real-time responsiveness.

B. Future Directions

Future work could explore the following directions:

- **Adding explanations to predictions:** Enhancing the web application to provide explanations for toxicity predictions, such as highlighting specific phrases or words contributing to the classification.
- **Expanding the dataset:** Collecting more diverse conversational data to improve model generalization and robustness, particularly for underrepresented classes in couple dynamics.
- **Exploring other transformer architectures:** Investigating alternative transformer models, such as RoBERTa or GPT-3, to compare their performance on toxicity detection and couple dynamics analysis.
- **Use of different loss functions:** Experimenting with different loss functions, such as focal loss or label smoothing, to address class imbalance and improve model performance on minority classes.
- **Incorporating user feedback:** Implementing a feedback loop in the web application to allow users to provide corrections or additional context, which could be used to iteratively improve model performance.

VIII. Code and Data Availability

Code and dataset are available on GitHub at the following link: https://github.com/Davy592/NLP

References

[1] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," 04 2016, pp. 145–153.

[2] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, 03 2017.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 10 2018.

[4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 07 2019.

[5] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," 01 2017, pp. 1–10.

[6] A. Xenos, J. Pavlopoulos, I. Androutsopoulos, L. Dixon, J. Sorensen, and L. Laugier, "Toxicity detection can be sensitive to the conversational context," 11 2021.

[7] C. Bosco, F. Dell'Orletta, F. Poletto, M. Sanguinetti, and M. Tesconi, "Overview of the evalita 2018 hate speech detection task," pp. 67–74, 01 2018.

[8] F. Poletto, V. Basile, M. Sanguinetti, C. Bosco, and V. Patti, "Resources and benchmark corpora for hate speech detection: a systematic review," *Language Resources and Evaluation*, vol. 55, pp. 1–47, 06 2021.

[9] F. Mairesse, M. Walker, M. Mehl, and R. Moore, "Using linguistic cues for the automatic recognition of personality in conversation and text," *J. Artif. Intell. Res. (JAIR)*, vol. 30, pp. 457–500, 09 2007.

[10] J. Oberlander and S. Nowson, "Whose thumb is it anyway? classifying author personality from weblog text." 01 2006.

[11] A. Kazemeini, S. Fatehi, Y. Mehta, S. Eetemadi, and E. Cambria, "Personality trait detection using bagged svm over bert word embedding ensembles," 07 2020.

[12] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "Bleu: a method for automatic evaluation of machine translation," 10 2002.

[13] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," 01 2004, p. 10.