,

Sergio Di Donfrancesco
Non-confidential

GRADUATION INTERNSHIP REPORT

Host Organization:	Istituto Nazionale di Statistica (ISTAT)
Internship Subject:	Emotion Detection on hate speech and gender-based violence texts on social media
Internship Location:	ISTAT, Rome (Italy)
Academic Year:	2024–2025
Internship Supervisor:	Elena Catanese
ENSAI Internship Advisor:	François Portier

Contents

List of Acronyms

GBV	Gender-Based Violence
EIGE	European Institute for Gender Equality
UNESCO	United Nations Educational, Scientific and Cultural Organization
ICFJ	International Center for Journalists
AMI	Automatic Misogyny Identification
EVALITA	Evaluation of NLP and Speech Tools for Italian
EMit	Emotions in Italian
BERT	Bidirectional Encoder Representations from Transformers
LLM	Large Language Model
incel	Involuntary Celibate
SVM	Support Vector Machine
ELMo	Embeddings from Language Models
RAI	Radiotelevisione Italiana (Italian public broadcaster)
ISTAT	Istituto Nazionale di Statistica (Italian National Institute of Statistics)
NLP	Natural Language Processing
CE	Cross-Entropy (loss function)
LoRA	Low-Rank Adaptation
Adam	Adaptive Moment Estimation
LR	Learning Rate
OOM	Out of Memory

1. Introduction

1.1 Context and motivation

Gender-based violence (GBV) and online hate speech are now widespread on social media and other digital platforms. They influence who participates in discussions, who is excluded, and how public debate evolves. A report from the EIGE (2022) shows that, besides offline violence, cyber violence already affects many women and girls. Digital spaces have therefore become a place of harm and exclusion.

The risks are even greater for women in public roles. A study from UNESCO and ICFJ (2022) shows that women journalists are often targeted by online attacks, which limit their freedom of expression and reduce participation in public life. General surveys, as the one in Stevens et al. (2024) (Figure 1.1), confirm this trend: women, especially young women, report more cases of sexual harassment online than men. This confirms the gendered nature of online abuse.

Research on misogyny and hate speech has grown quickly in recent years. Fontanella et al. (2024) show that studies on online misogyny and automatic detection have increased drastically since 2018 (Figure 1.2). However, these studies are still weakly connected to broader research on violence and the "manosphere". Tontodimamma et al. (2020) highlight that, concerning the study of hate speech, in the last thirty years the focus has shifted from theoretical debates to machine learning methods, with special attention to gendered hate and cyberbullying.

In Italy, evaluation campaigns have supported the development of new datasets and methods. The AMI task at EVALITA by Fersini et al. (2018) provided Twitter data in Italian and English for misogyny detection. The EMit task at EVALITA (Araque et al. 2023) addressed emotion classification in Italian texts, showing the maturity of transformer-based approaches for social media analysis.

At the methodological level, transformers and large-scale pretraining have changed the way text classification is done. The Transformer model introduced self-attention, which made sequence processing faster and more effective. BERT later showed that bidirectional pretraining can be fine-tuned efficiently for many tasks, including social media classification, as stated in Devlin et al. (2019) and Vaswani et al. (2017).

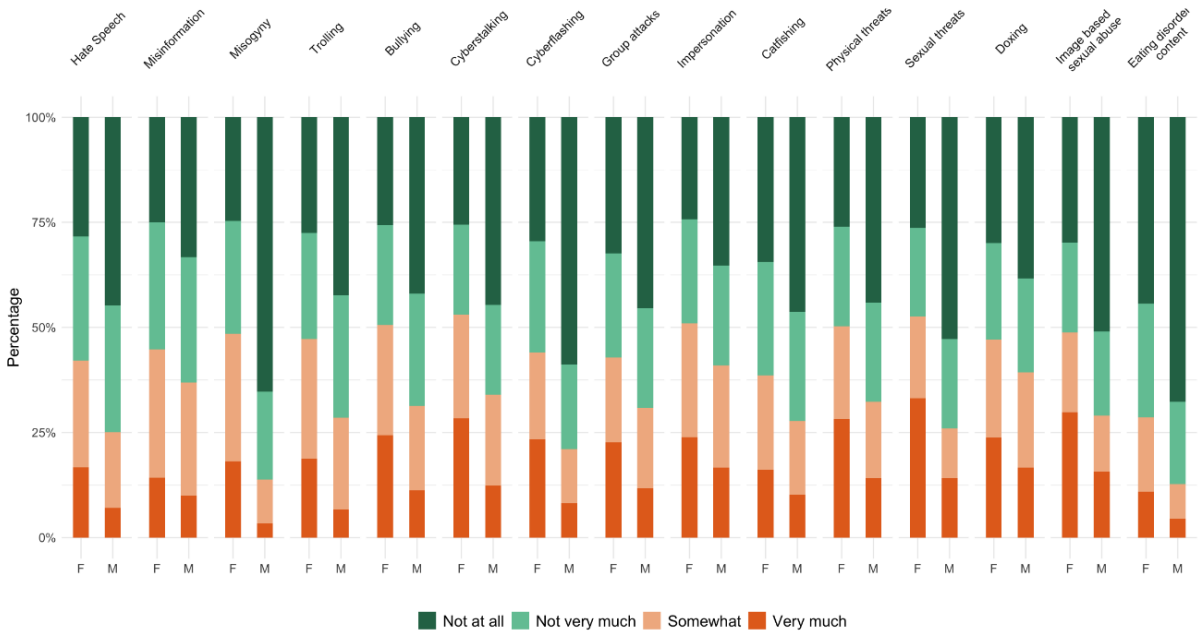

Figure 1.1: Self-reported fear of receiving harmful content online by gender.
Source: Stevens et al. (2024)

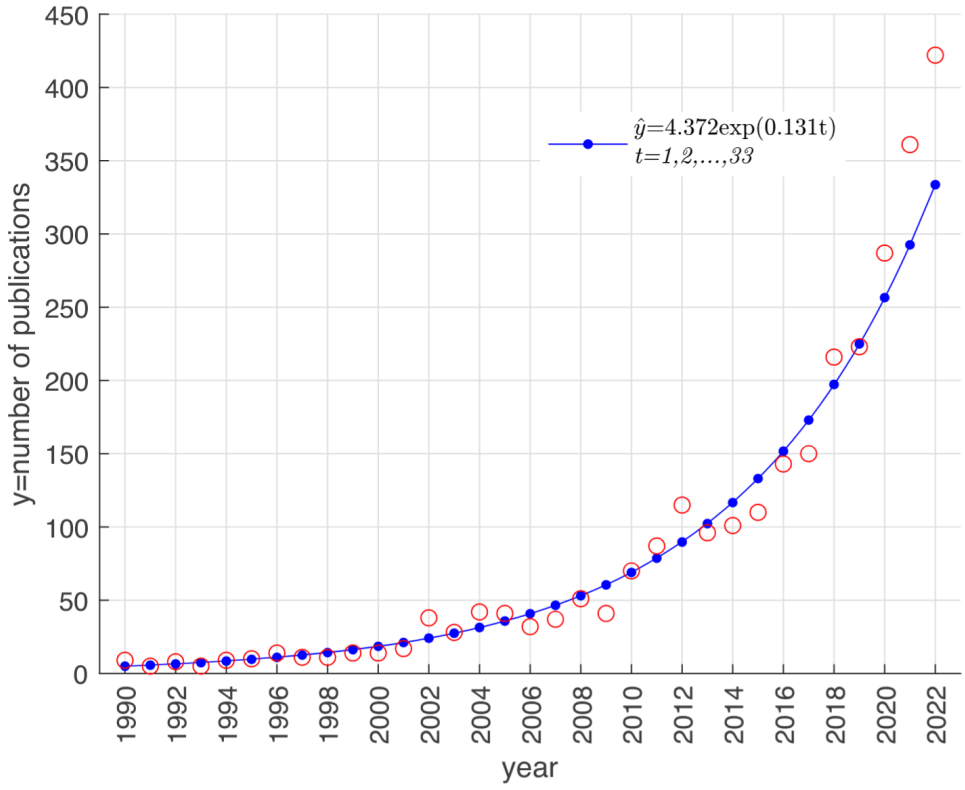

Figure 1.2: Number of publications on misogyny per year: observed and expected temporal evolution according to exponential growth.
Source: Fontanella et al. (2024)

1.2 Objectives

The project aims to lay the foundation to build and assess a robust Italian emotion classifier for social media texts, with a specific focus on content related to gender-based violence. The core objective is to fine-tune a Transformer encoder and to evaluate its ability to recognize emotions reliably both on general social posts and on GBV-related material.

Within this goal, the main objectives are to study learning under data constraints: quantify how choices aimed at coping with class imbalance and small samples affect metrics behaviour, and articulate the trade-offs between stability and capacity. Another aim is to provide a reproducible, extensible pipeline that can be updated when new GBV annotations become available, and that supports future monitoring use-cases through transparent reporting of metrics, errors, and limitations. An important aim of the project is to justify design choices against the objectives above, to measure their impact rigorously, and to outline how the resulting system can be extended and trusted in practical analysis.

1.3 Document structure

The thesis is organized to move from context and theory to implementation, evaluation, and external validation. Chapter 2 surveys prior research and conceptual groundings: it clarifies the main definitions and phenomena, summarises the methodological background that links NLP practice with social science perspectives, and positions the EMit shared task as a benchmark. Chapter 3 sets up the problem operationally: it states the task and goals, outlines the end-to-end workflow adopted in the project, and describes the computational environment used for experiments. Chapter 4 documents the data: it presents the initial datasets employed in the study and details the pre-processing and integration pipeline that yields the final training, validation, and test splits.

Chapter 5 introduces the modelling foundations used throughout: the principles behind LLMs with a focus on Transformer encoders and BERT, and the Italian backbones selected for the experiments. Chapter 6 reports the full training and evaluation process: it describes the preliminary operations prior to fine-tuning, the training on general data, the subsequent adaptation on GBV data, and the final assessment of the models on their test sets, discussing metrics, strengths and weaknesses, and the implications of design choices. Chapter 7 extends the analysis to real-world, non-annotated social-media posts collected in the last quarter of 2023: it characterizes this external dataset and presents the model's predictions, including a comparison with labels available in the corpus from another model to interpret agreements and mismatches. Chapter 8 closes the thesis by synthesizing the main findings, reflecting on limitations, and outlining directions for future work.

2. Related Work and Theoretical Background

2.1 Definitions and key concepts

Misogyny is a system of beliefs, emotions, and practices that devalue or express hostility toward women as a group.

Sexism refers to attitudes and structures that produce and justify gender-based inequality, including both overt hostility and more subtle forms.

Cyberviolence denotes intentional harm mediated by digital technologies (e.g. harassment, threats, doxxing, non-consensual image sharing) and includes behaviours that extend offline violence into online spaces (EIGE 2022).

Hate speech is not universally defined, but it is commonly referred to as any communication that disparages a person or a group on the basis of some characteristics such as race, colour, ethnicity, gender, sexual orientation, nationality, or religion (Tontodimamma et al. 2020). A comprehensive overview of different definitions can be found in Sellars (2016).

These phenomena are interconnected: misogyny and sexism often motivate hate speech and cyberviolence. The manosphere, a loose network of online communities that promote anti-feminist and hyper-masculinist ideas, together with *incel* communities (groups of "involuntary celibates" who often blame women and social norms for their situation) serve as incubators and amplifiers. "*Networked misogyny*" refers to the cross-platform spread of hostile narratives through coordinated actions, memes, and engagement-driven recommendation systems, which amplify reach and persistence (Fontanella et al. 2024).

2.2 Methodological background

From feature-based models to pretrained LMs and LLMs

Early hate-speech detection relied on feature-based pipelines with lexical resources, word/character n-grams, and surface/syntactic features trained with linear classifiers (e.g. SVM, logistic regression; Davidson et al. 2017). Contextual pretraining then reshaped the field: deep contextualized word representations (ELMo) improved handling of polysemy and context, as shown in Peters et al. (2018), then bidirectional masked-language models (BERT) made end-to-end fine-tuning standard Devlin et al. (2019). Figure 2.1 shows the

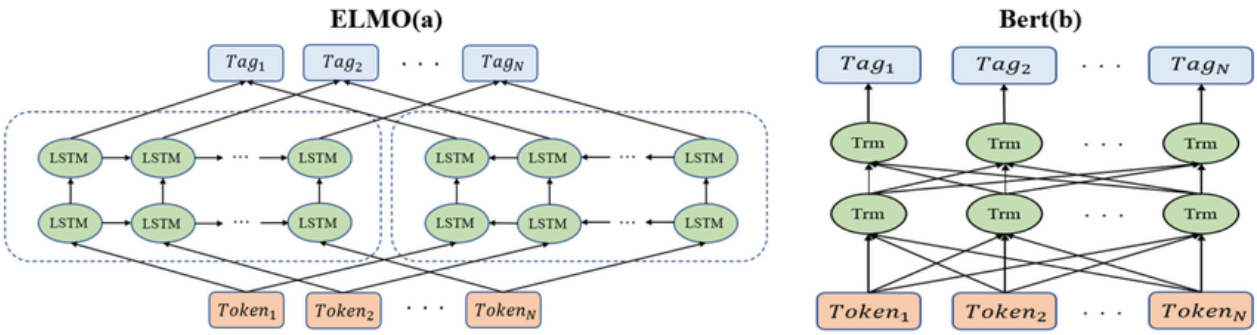

Figure 2.1: ELMO (a) and BERT (b) architecture.
Source: Song et al. (2021)

different architectures of the two models.

Quantitative NLP and social-science perspectives

Detection systems work best when they are linked to theory and supported by mixed methods. Reviews of the field show that research on automatic detection, gendered hate, and the wider manosphere often develops in separate ways, with only limited connections between them. Poletto et al. (2021) highlights the need for clearer definitions and annotation schemes that are consistent with social science concepts. Bringing together quantitative NLP and social science perspectives involves several key practices:
(i) ensuring that labels and categories have strong validity;
(ii) considering how platforms and time periods shape the data;
(iii) paying attention to fairness and uneven error rates across groups;
(iv) using qualitative analysis to better understand model errors.
This combined "two-lens" approach, common in computational social science, strengthens the reliability of results and makes models more useful for real-world monitoring and policy, as stated in Lazer et al. (2009).

2.3 Benchmarks and Baselines: the EMit Task

EMit (Emotions in Italian, Araque et al. 2023) is the first shared task on categorical emotion detection in Italian social media. It comprises two subtasks: Subtask A (emotion detection) and Subtask B (target detection). Subtask A frames emotion detection as a multi-label classification problem over ten labels: the eight basic emotions in Plutchik's model (*anger, anticipation, disgust, fear, joy, sadness, surprise, trust*; Plutchik 1980), plus *love* and *neutral*. This project is based on EMit Subtask A in both task definition and label set, but the emotion detection is performed as a multi-class classification problem.

Datasets used in EMit

EMit uses a dataset consisting of 6966 Italian tweets discussing RAI TV programmes; 5966 are used for training and 1000 for test. Annotations are multi-label at the emotion layer (Table 2.1); about 78% of tweets

Label	Train	Test
Anger	367	56
Anticipation	547	85
Disgust	874	165
Fear	91	13
Joy	650	100
Love	633	103
Neutral	1 322	210
Sadness	545	95
Surprise	591	102
Trust	1 665	272

Table 2.1: Labels distribution in EMit datasets

express at least one emotion and 19% express two or more.

Reference metrics

Official ranking uses macro-averaged F1 over classes. The *F1 score* is the harmonic mean of *precision* and *recall*. For a given class c:

- Precision measures the proportion of correctly predicted instances of c among all instances predicted as c:

$$\text{Precision}(c) = \frac{TP_c}{TP_c + FP_c},$$

  where TP_c are true positives for class c, and FP_c are false positives.

- Recall measures the proportion of correctly predicted instances of c among all actual instances of c:

$$\text{Recall}(c) = \frac{TP_c}{TP_c + FN_c},$$

  where FN_c are false negatives for class c.

- The F1 score for class c is then defined as:

$$F1(c) = \frac{2 \cdot \text{Precision}(c) \cdot \text{Recall}(c)}{\text{Precision}(c) + \text{Recall}(c)}.$$

The macro F1 score averages the F1 scores computed for each class, giving equal weight to all classes regardless of their frequency:

$$\text{Macro-F1} = \frac{1}{|C|} \sum_{c \in C} F1(c), \tag{2.1}$$

where C is the set of classes.
This metric is particularly useful in imbalanced classification settings, as it prevents majority classes from dominating the evaluation.

Organizers also provide simple baselines based on logistic regression fed by word uni/bi-grams with TF-IDF or one-hot encodings, plus a random baseline.

Best results and winning approaches

In Subtask A, the best macro-F1 is 0.6028. Competitive systems rely on transformer/LLM backbones:
(i) instruction/fine-tuned encoder–decoder or decoder-only LLMs (IT5, LLaMA);
(ii) fine-tuned Italian BERT variants (e.g. AlBERTo);
(iii) ensembles combining multiple LLMs, including mT5 and few-shot GPT-3.5 prompts.
Baselines achieve worse results (e.g. TF-IDF macro-F1 is ≈ 0.41).

3. Problem Setup and Workflow

3.1 Task definition and objectives

The project aims at emotion detection through single-label, multi-class classification of Italian social media texts related to gender-based violence. Each instance is assigned exactly one emotion, learned via fine-tuning of a Transformer encoder. The objective is to build a robust classifier that performs well with GVB data and maintains acceptable performance on data coming from different sources.

3.2 Pipeline overview

The work is structured in the following phases:

1. **Data collection and preparation**: acquire labelled Italian social-media posts from EMit and ISTAT datasets; consistently merge the data, apply cleaning, de-duplication, language filtering, and split into train/validation/test.
2. **Exploratory analysis**: inspect labels, identify class imbalance and potential sources of bias.
3. **Backbone selection**: choose a suitable Italian Transformer considering domain, sequence length, and computational budget.
4. **Generic pre-finetuning**: tune from a pretrained model on an Italian social networks corpus to improve domain coverage and stability.
5. **Finetuning on GBV data**: train again on GBV data.

6. **Models optimization**: analyze critic steps and explore solutions to improve the performance of the model and solve bottlenecks.

7. **Evaluation and error analysis**: report Accuracy and Macro-F1 on the test set, analyse rare-class behaviour.

8. **Testing on new data**: run inference on fresh samples to assert the performance of the model; collect insights for future annotation or model updates.

3.3 Computational environment

All experiments are run on Google Colab, using T4 GPU acceleration. Experiment tracking, metric logging, hyperparameter comparisons, and artifact versioning are managed with Weights & Biases (W&B), a machine learning platform that helps tracking experiments, managing datasets, and monitoring models during training and deployment. Notebooks, data, models and logs are uploaded on this GitHub repository (Di Donfrancesco 2025).

4. Data

4.1 Initial datasets

EMit (as described in the previous section)

We adopt the EMit Subtask A dataset (Italian social media). In the original release, a tweet can carry multiple emotion labels (multi-label). All user-identifying information was removed and IDs were regenerated before release, which is restricted to academic use under GDPR compliance.
The EMit dataset was annotated with a multi-layer scheme by multiple annotators. Tweets were organized in sets; each set was annotated by three independent annotators (five sets in total, i.e. 15 annotators overall).
For comparability with our single-label, multi-class setting, we later filter EMit to keep only mono-label instances (see Section 4.2).

ISTAT datasets

The ISTAT data follow a simple schema `(label, text)` and are provided in predefined splits (train/valid/test) and in three category folders:

Text [EN]	Joy	Ang.	Trust	Neu.	Sad.	Love	Dis.	Sur.	Ant.	Fear
Caspita che meraviglia [Gosh what a wonder] #LAmicaGeniale	1	0	1	0	0	0	0	0	0	0
Queste persone mi spezzano il cuore [These people break my heart] #amorecriminale	0	0	0	0	1	0	0	0	0	0
il sabato sera con [Saturday evening with] #albertoangela #viaggiosenzaritono #leggirazziali – Watching Ulisse	0	0	0	0	0	1	0	0	0	0
Ma i genitori di questi idioti non li hanno mai mandati a scuola? [But didn't the parents of these idiots ever send them to school?] #pechinoexpress	0	1	0	0	0	0	1	0	0	0

Table 4.1: EMit dataset head

Label [EN]	Text [EN]
NEUTRA [Neutral]	È la difficoltà di trovare il verbo [It's the difficulty in finding the verb]
GIOIA [Joy]	Questa tua proposta mi intriga molto e sono contento di aderire [Your proposal intrigue me much and I'm happy to join you].
RABBIA [Anger]	Alberto era fuori dai gangheri ! [Alberto was out of his mind!]
PAURA [Fear]	sono tanto preoccupato per voi [I'm really worried for you].

Table 4.2: ISTAT (train-all) dataset head

- **all**: general Italian social-media posts annotated with emotions;
- **mix**: a curated subset or alternative sampling of the general data;
- **gbv**: posts explicitly related to gender-based violence.

During inspection we observed that the intended distinction between *all* and *mix* is not fully clear from metadata, and that there are *overlapping records* (identical or near-duplicate texts) across the three categories. We therefore handle overlaps explicitly during integration (Section 4.2).
Examples of both EMit and ISTAT datasets can be found in Table 4.1 and Table 4.2.

4.2 Pre-processing and integration pipeline

We apply a unified pipeline to align EMit and ISTAT corpora to our setting, remove label inconsistencies, and produce comparable splits for modelling. Below we summarize the main steps (dimensions are reported in Tables 4.3–4.4).

- **Filter EMit train and test splits to single-label instances.** We retain only tweets annotated with exactly one emotion. This converts EMit Subtask A from multi-label to single-label and reduces label noise (Table 4.3a).

(a) Outcome of EMit single-label filtering.

EMit (single-label filter)	Train	Test
Original	6 966	1 000
After filtering	4 791	135

(b) ISTAT `all`+`mix` integration per split after de-duplication and conflict removal.

Split	all	mix	joined
Train	1 854	677	1 749
Valid	327	120	396
Test	546	200	647

Table 4.3: Dataset sizes during preprocessing.

(a) EMit stratified split (from single-label train) and unchanged test.

EMit (post split)	Train	Valid	Test
Stratified split on train	3 832	959	135

(b) Final dataset sizes per split for General and VDG families.

Split	General (EMit + ISTAT all/mix)	VDG (ISTAT only)
Train	5 461	366
Valid	1 347	64
Test	781	108

Table 4.4: Dataset sizes after preprocessing.

- **Join ISTAT-all and ISTAT-mix sets; de-duplicate and drop conflicts.** For each split (train/valid/test) we concatenate the two sources, remove exact duplicates (after normalization), and discard conflicting texts (same normalized text, different labels). Resulting sizes per split are in Table 4.3b.
- **Label harmonization.** We collapse *Disgust* into *Anger* on EMit to mitigate sparsity; on ISTAT we map Italian labels to the EMit English inventory (e.g. GIOIA → *Joy*, NEUTRA → *Neutral*) so that merged datasets share the same label space.
- **Stratified split on EMit train.** From the single-label EMit train we derive a stratified validation set that preserves class proportions (Table 4.4a).
- **Merging of EMit and ISTAT data.** For each split (train/valid/test) we merge EMit (single-label) with ISTAT general data (joined `all+mix`, after de-duplication). GBV-specific data are kept separate. Since the GBV splits come only from ISTAT, they don't contain two of the nine EMit labels available in the general corpus: *Anticipation* and *Trust*. As a consequence, the GBV fine-tuning (see Section 6.3) is conducted with a 9 to 7 label reduction, adapting the classification head accordingly, while the general-stage training mantains the full 9-label inventory for comparability, future extensions and richer supervision. Final sizes for General and GBV datasets appear in Table 4.4b.

5. LLMs, Transformers and BERT models

This chapter quickly presents the theoretical background of the models used in the project. We first introduce large language models (LLMs), with a focus on Transformer encoders and BERT-style models. We then describe the Italian backbones adopted in our experiments, highlighting differences and trade-offs.

LLMs. Large Language Models are neural networks trained on large text corpora to predict tokens. They learn general-purpose linguistic representations that transfer to many tasks.

Transformer encoders. The Transformer architecture replaces recurrence with self-attention. Each encoder layer computes multi-head self-attention to weight token-token interactions in parallel, and a position-wise feed-forward network. Residual connections and layer normalization stabilize training. Encoder-only Transformers read full context, which is ideal for classification of short texts.

BERT for classification. BERT (Bidirectional Encoder Representations from Transformers) is a trained Transformer Encoder stack. Its design involves pre-training deep bidirectional representations from the unlabeled text, conditioning on both the left and right contexts.
For classification, we append a linear head on the pooled representation of a special token (commonly `[CLS]`, see Figure 5.1). Training minimizes the cross-entropy loss over K classes:

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{k=1}^{K} y_k \log p_\theta(y = k \mid \text{input}),$$

where y_k is the one-hot label and p_θ is the softmax output.

Why BERT. Our task is single-label, multi-class emotion classification on short, noisy social posts. Encoder-only models like BERT are a natural fit because they encode bidirectional context efficiently, they fine-tune stably with modest compute. Moreover, strong domain-relevant Italian checkpoints (AlBERTo, UmBERTo) exist, and generation capabilities of decoder LLMs are not necessary for the task.

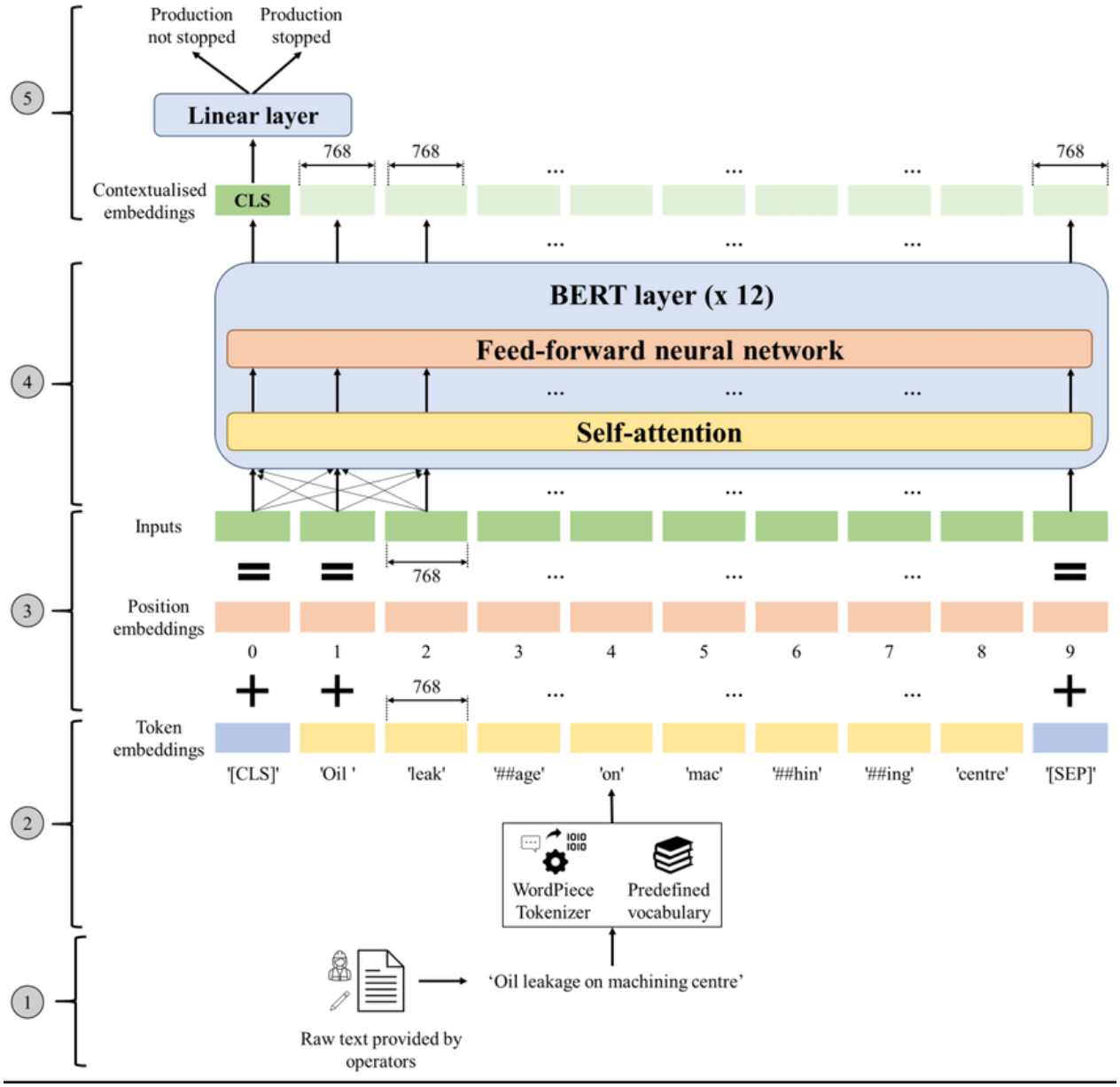

Figure 5.1: Example of a BERT architecture for classification.
Source: Usuga Cadavid et al. (2021)

5.1 Italian backbones: AlBERTo and UmBERTo

Both backbones are encoder-only Transformers pre-trained with masked language modelling. They differ mainly in pre-training data, tokenizer, and some design choices.

Key differences

- **Pre-training domain.** AlBERTo is trained primarily on Italian social media text, capturing slang, hashtags, and informal orthography; UmBERTo is trained on larger general-domain Italian corpora (news, web, Wikipedia), favoring broader coverage and cleaner syntax.
- **Tokenizer.** Both use subword tokenization; the actual vocabulary and merge rules differ because they are learnt from different corpora. This impacts OOV (Out Of Vocabulary) handling for emojis, hashtags, and user mentions.
- **Training choices.** Pre-training choices differ, yielding distinct biases (e.g. robustness to noisy punctuation vs. better long-word morphology).

AlBERTo is strong on short noisy posts, yielding to better handling of hashtags, mentions, and informal contractions. On the other hand, the specific domain of training data may limit generalization on formal text: vocabulary may fragment rare formal terms.
UmBERTo has broader coverage of standard Italian. It has stronger syntax and semantics on long words and formal registers, but may underfit highly informal slang if they aren't fine-tuned sufficiently.

6. Training and evaluation of the model

In this chapter we describe how we process and transform data from `.csv` files to datasets suitable for model operations, how we select the backbone model, how we train, regularize and evaluate models on general data, and how we adapt training to scarce GBV data in a second fine-tuning stage.

6.1 Pre-Training Operations

Early considerations

The used datasets are imbalanced (some emotions are underrepresented) and the GBV-specific split is small. Training a single model on the union of general and GBV data would tend to bias the model decision boundaries to majority classes and domain patterns from the general corpus. To avoid this problem, we decide to adopt a two-stage fine-tuning:

1. **Stage 1 (general)**: fine-tune on the larger, general corpus to learn robust Italian social-media features and a stable classifier head.
2. **Stage 2 (GBV adaptation)**: continue fine-tuning on the GBV subset with conservative hyperparameters.

This approach improves data efficiency (the backbone starts from a task-adapted minimum) and limits catastrophic forgetting compared with training from scratch on GBV or mixing everything with aggressive optimization.

From CSV to Hugging Face datasets: normalization, tokenization, batching

Starting from cleaned `.csv` files with fields `text` and `label`, we build a `datasets.DatasetDict` with `train`/`validation`/`test` splits:

- **Social normalization.** We optionally apply a normalizer tailored to social text that replaces URLs, user mentions, hashtags, and social typical puctuation with special tokens (e.g. `<url>`, `<user>`, `<hashtag>`, `<interrobang>`) and splits hashtags into subwords when clearly compositional. This approach has shown to reduce sparsity and make tokenization more consistent.

Label	Joy
CSV row (raw)	"Wow!!! Che serata a #Sanremo2024 https://t.co/abc"
After social normalisation	"wow <exclamation> che serata a <hashtag> sanremo 2024 <url>"
Tokens	[CLS], wow, <exclamation>, che, serata, a, <hashtag>, sanremo, 2024, <url>, [SEP]
Token IDs (illustrative)	[101, 2450, 999, 3021, 632, 5872, 120, 7781, 2099, 2090, 102, 0, 0, 0, 0, 0]
Attention mask (max_len = 16)	[1,1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0]

Table 6.1: Minimal example from CSV row to token IDs and attention mask after normalization. Token IDs are illustrative.

- **Tokenization.** We use the backbone's subword tokenizer (SentencePiece) with a fixed `max_length`; sequences are padded as needed. Special tokens (e.g. `[CLS]`, `[SEP]`) are handled by the tokenizer.
- **Batching.** A DataCollator applies dynamic padding to the longest sequence in the batch and assembles `input_ids`, `attention_mask`, and `labels`.

Backbone decision: UmBERTo vs. AlBERTo

UmBERTo was considered initially because it represents a strong, general-domain Italian BERT model trained on large and diverse corpora. This broad coverage made it a natural first candidate: it offers stable syntax–semantic representations, good generalization across topics, and had already proven competitive in several Italian NLP tasks. The fact that our data consist mainly of social-media posts motivated the choiche to try another italian backbone: AlBERTo, trained on social-media domain data.
AlBERTo delivered higher performance and faster, stabler convergence. This is due to domain alignment: its subword inventory and priors better capture slang, hashtags, user mentions, elongated spellings, and emoji patterns. As a consequence, there are fewer awkward splits on colloquialisms and hashtag compounds, which reduce fragmentation and information loss at the embedding layer. Having relatively small and imbalanced datasets, the social-media pretraining of AlBERTo yields stronger features early in fine-tuning, improving minority-class recall, and we can apply lighter "social normalization" (URLs/mentions placeholders) without heavy domain cleaning. Furthermore, the better initial fit reduces the step size needed during the second GBV-focused fine-tuning, reducing overfitting risks. We therefore adopted AlBERTo as the primary backbone.

6.2 Training on general data

Class imbalance: from weighted loss to dynamic oversampling

To address the problem of class imbalance (Table 6.2), we tried with a weighted cross-entropy (CE) loss, assigning a weight w_k to each class k (inversely proportional to its frequency). Given a predicted distribution

(a) General training data.

Label	Count	Percent (%)
Anger	833	15.25
Anticipation	287	5.26
Fear	324	5.93
Joy	494	9.05
Love	173	3.17
Neutral	1525	27.93
Sadness	537	9.83
Surprise	440	8.06
Trust	848	15.53

(b) GBV training data.

Label	Count	Percent (%)
Anger	93	25.62
Fear	8	2.20
Joy	70	19.28
Love	88	24.24
Neutral	33	9.09
Sadness	55	15.15
Surprise	16	4.41

Table 6.2: General and GBV Train splits: label distribution.

$p_\theta(y \mid x)$ over K classes and a one-hot encoded target vector y, weighted CE is defined as

$$\mathcal{L}_{\text{wCE}} \;=\; -\sum_{k=1}^{K} w_k \, y_k \log p_\theta(y{=}k \mid x).$$

Using this loss increases the penalty on minority-class errors without altering the data distribution, but large w_k can destabilize optimization, degrade probability calibration, and interact poorly with other regularizers. We then tried oversampling, that is duplicating minority examples to balance batches. It strengthens the gradient signal for rare classes, but the training distribution diverges from validation and test, which remain imbalanced; this can inflate evaluation metrics if not interpreted cautiously. In addition, repeated texts increase memorization risk, leading to faster overfitting. The worsening in generalization can be also due to reduced batch diversity.

The final choice is light, dynamic oversampling: we apply a small oversampling factor for minority classes during the first epochs and then anneal it back towards uniform sampling. This provides early gradient support where it is most needed and limits duplication later mitigating overfitting. The class imbalance is mitigated but not eliminated; rare-class metrics remain more volatile than majority ones.

Regularization: label smoothing, early stopping, and gradient clipping

For a K-class problem with gold class y, the one-hot target is

$$y_k = \begin{cases} 1 & \text{if } k = y, \\ 0 & \text{otherwise.} \end{cases}$$

With label smoothing we use a softened target

$$\tilde{y}_k = (1 - \varepsilon)\, \mathbb{1}[k = y] + \frac{\varepsilon}{K},$$

which discourages overconfident predictions and improves calibration. We chose $\varepsilon = 0.03$ balancing the trade-off between calibration (larger ε reduces variance and overconfidence) and class separability (too large ε dilutes the signal for minority classes and lowers recall). At $\varepsilon = 0.03$ we observed steadier validation curves and no measurable drop in performance compared with $\varepsilon = 0$.

For early stopping we monitor macro-F1 (with patience = 2 evaluations) rather than the loss, because the loss can keep decreasing while macro-F1 stays steady or even worsens due to shifts in the precision/recall balance on rare classes; monitoring macro-F1 stops training when the metric plateaus.

Gradient clipping limits the update size by capping the global gradient norm. Let g be the concatenated gradient vector over all trainable parameters for a batch; with threshold $\tau = 1$ we apply

$$g \;\leftarrow\; g \cdot \min\!\left(1,\; \frac{\tau}{\|g\|_2}\right)$$

before the optimizer step. This prevents unstable jumps in early epochs while leaving small-norm updates unchanged.

Too many trainable parameters

Fine-tuning a pretrained BERT with $P = 110M$ parameters by updating all weights can be computationally expensive and can lead to overfitting, especially when the training dataset is modest. Two remedies were explored to reduce the number of trainable parameters P_{train}:

- **freezing** a subset of layers (no updates on those weights);

- **LoRA** (Low-Rank Adaptation, Hu et al. 2021), which freezes the backbone but adds small trainable low-rank adapters on selected projection matrices.

The practical goal is to minimize P_{train} and memory/compute, while preserving enough capacity to adapt to the domain and to avoid overfitting.

Layer freezing sets $\nabla W_\ell = \mathbf{0}$ for frozen layers $\ell \in \mathcal{F}$ and only updates unfrozen layers $\ell \in \mathcal{U}$ and the classifier head:

$$P_{\text{train}}^{\text{freeze}} \;=\; \sum_{\ell \in \mathcal{U}} |W_\ell| + |W_{\text{cls}}| \;\ll\; P.$$

Instead of updating the full weight matrix W of a linear/attention projection, LoRA keeps W frozen and learns only a small, low-rank update ΔW. The update is factorized as

$$\Delta W \;=\; \tfrac{\gamma}{r}\, BA, \quad A \in \mathbb{R}^{d \times r},\;\; B \in \mathbb{R}^{r \times d},$$

with rank $r \ll d$ (see Figure 6.1). For an input vector x the forward pass becomes

$$(W + \Delta W)\, x \;=\; Wx \;+\; \tfrac{\gamma}{r}\, BAx,$$

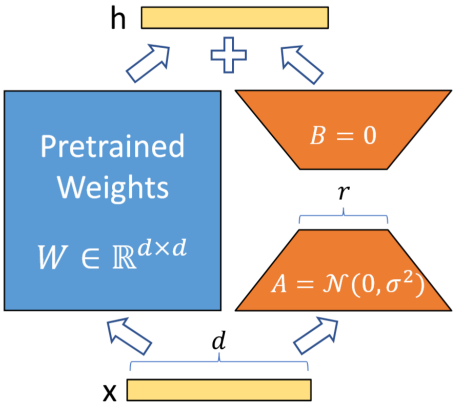

Figure 6.1: Schematization of the LoRA decomposition.
Source: Hu et al. (2021)

so W provides the pretrained mapping while BA adds a small, learned correction (often with a scaling γ). Only A and B receive gradients; W stays fixed.

Given target modules $m \in \mathcal{M}$ (e.g. W_q, W_v per layer), the number of trainable parameters using LoRA is:

$$\Delta P_{\text{LoRA}} \approx \sum_{m \in \mathcal{M}} \left(|A_m| + |B_m| \right) = 2rd \cdot |\mathcal{M}|.$$

For example, with hidden size d=768, rank r=8, and $|\mathcal{M}| \approx 24$ (e.g., W_q and W_v across 12 layers), the extra trainable parameters are

$$\Delta P_{\text{LoRA}} \approx 2 \cdot 8 \cdot 768 \cdot 24 = 294{,}912 \ (\approx 0.27\% \text{ of } 110\text{M}),$$

dramatically smaller than full fine-tuning.

During early training, solutions like oversampling and label smoothing can induce large, unstable gradients on the backbone if it is fully trainable. We therefore adopt a partial freezing warm-up: for the first two epochs we freeze the lower Transformer blocks (preserving robust general linguistic features) and update only the upper blocks plus the classifier. This has three benefits:

1. **Stability at the start:** by limiting P_{train} we reduce optimization noise and protect the pretrained manifold from early drift; empirically, the validation F1 stabilizes faster.

2. **Regularization:** freezing acts as an implicit prior, lowering variance while the classifier adapts to domain-specific cues.

3. **Capacity recovery later:** after two epochs we unfreeze deeper layers to regain representation power and let the model learn task-specific features once the head is reasonably calibrated.

Compared to LoRA, this strategy avoids additional design choices (rank r, target modules $\mathcal{M}$, scaling γ) and engineering overhead. Hence, it provided a simpler and robust trade-off between compute, generalization, and

final performance.

Optimizer: AdamW

Adam (Adaptive Moment Estimator) is an adaptive stochastic optimizer that maintains first and second moment estimates of the gradient: $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$, $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$, with bias-corrected moments $\hat{m}_t = m_t/(1-\beta_1^t)$ and $\hat{v}_t = v_t/(1-\beta_2^t)$. Parameters are updated as

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon},$$

where, α is the base learning rate; β_1 and β_2 are the exponential decay rates for the first- and second-moment estimates; and ε is a small constant for numerical stability.
It adapts the learning rate per-parameter, is robust to different feature scales, and works well with warmup schedules. These reasons make it a strong default for Transformer fine-tuning. AdamW modifies Adam by decoupling weight decay from the adaptive update. Instead of emulating ℓ_2 regularisation inside the gradient, AdamW applies decay directly to the weights:

$$\theta_{t+1} \;=\; \theta_t\big(1 - \alpha\lambda\big) \;-\; \alpha\, \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon},$$

where λ is the weight decay factor. This yields better generalization and more predictable tuning on BERT-like models.

Learning rate, LR scheduler, and warmup ratio

The learning rate (LR) controls the step size of each parameter update: too high can cause divergence, too low leads to underfitting. A LR scheduler makes LR a function of training progress to stabilize and improve generalization. We adopt a linear warmup followed by a cosine decay schedule (Figure 6.2).
For total steps T and warmup steps $w = \lfloor r_w T \rfloor$, with base LR α_0, we use

$$\alpha(t) = \begin{cases} \alpha_0 \cdot \dfrac{t}{w}, & 0 \le t < w, \\ \alpha_0 \cdot \frac{1}{2}\left[1 + \cos\Big(\pi\, \dfrac{t-w}{T-w}\Big)\right], & w \le t \le T. \end{cases}$$

For our final model we selected the LR $\alpha_0 = 2.5 \cdot 10^{-5}$ and the cosine schedule with warmup ratio $r_w = 0.06$. We opted for a cosine schedule instead of a linear decay because cosine annealing provides a smoother reduction of the learning rate. This smoother curve ensures that the model continues to make small but meaningful updates in the later epochs, which is useful for capturing minority-class details without overshooting. In practice, this resulted in more stable convergence and slightly better macro-F1 scores compared to linear decay. With about 6% of steps dedicated to warmup, the classifier head and the upper encoder layers had enough time to stabilize before larger updates were applied. Smaller ratios (≤ 0.02) led to noisier early gradients,

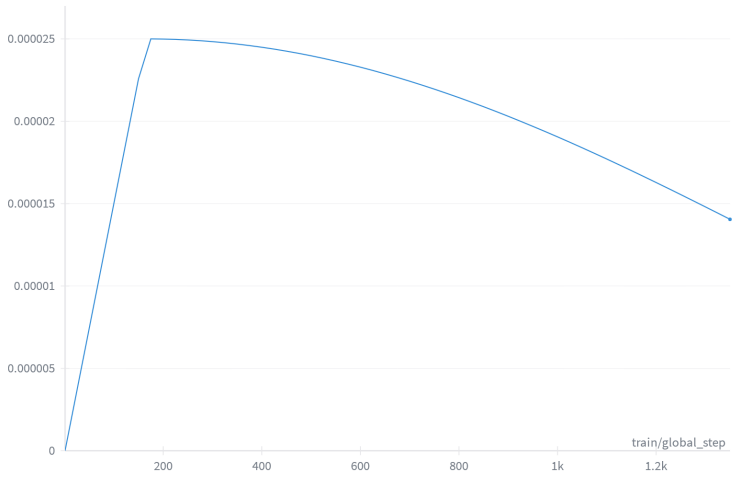

Figure 6.2: Learning Rate evolution during training.

while larger ratios (≥ 0.1) slowed learning unnecessarily. Thus, 0.06 represented a balanced trade-off, ensuring both stability in the first epochs and efficient use of the training budget.

Mixed precision and batch configuration for memory usage optimization

We use mixed precision to train faster and fit larger batches in GPU memory. FP16 (Half Precision) stores each number in 16 bits instead of 32 (FP32): 1 sign bit, 5 bits for the exponent, and 10 bits for the fraction (mantissa). It offers good detail for fractions but covers a narrower range of magnitudes. BF16 (BFloat16) also uses 16 bits, but splits them as 1 sign, 8 exponent, and 7 fraction bits. It sacrifices some fractional detail to cover a much wider range of values, which usually makes training more stable. In practice, we prefer BF16 when the GPU supports it, otherwise we use FP16. Evaluation is kept in (FP32) to avoid rounding effects. The per-device batch size is how many examples a single GPU sees at once. We choose the largest value that fits in memory, helped by dynamic padding (we pad only up to the longest sequence in each batch). When this size is not enough to reach a good optimization behaviour, we use gradient accumulation steps A: we process several small mini-batches and update the model once, as if we had used one larger batch. This lets us keep memory usage low while enjoying the benefits of a larger effective batch B_{eff}:

$$B_{\text{eff}} = B_{\text{per-device}} \times N_{\text{device}} \times A.$$

We choose a combination of $B_{\text{per-device}} = 8$ and $A = 2$ empirically to hit a stable B_{eff} that didn't produce OOM (out-of-memory) instances. This configuration doesn't affect the LR scheduler, which is defined over optimizer steps (after accumulation) and thus uses the correct step count.

6.3 Training on GBV data

This section describes how we adapted the training procedure to the small GBV dataset (see Section 4.2) while keeping the same architecture as in the general training. We resume from the best checkpoint selected on general data (both model weights and the same tokenizer) and continue fine-tuning on GBV posts. The expectation is to specialize the decision boundaries to GBV-related expressions without overfitting or forgetting the general patterns learned in the first stage.

Before GBV fine-tuning we reduced the label set from 9 to 7 classes. Keeping outputs for classes with no training examples would misalign the objective and waste capacity on logits that can only learn from noise. We therefore adapt the classification head by replacing the final linear layer with a 7-way output that matches the GBV label space.

We deliberately kept all the 9 classes in the general training stage to future-proof the pipeline: in the future we may obtain GBV datasets annotated also with *Anticipation* and *Trust*, and mantaining these classes in general training makes easier to implement subsequent integration without restarting the architecture of the model.

Regularization and stopping

We reduced the smoothing from $\varepsilon = 0.03$ to $\varepsilon = 0.02$. On a small GBV set, too much smoothing can blur minority signals. The lower value preserved calibration benefits while retaining class separability. Regarding early stopping, we kept a patience of 2 but, given the higher variance of macro-F1 on small validation sets (only 64 examples), we monitor the validation loss to trigger checkpointing. Loss is smoother epoch-to-epoch in this regime.

Capacity, optimizer, and learning-rate policy

To limit overfitting while still allowing adaptation, we apply a selective freeze at the start of GBV training: embeddings and the first encoder block are frozen for the initial epochs, then unfrozen once the classifier head stabilizes. As in the general stage, freezing was preferred over LoRA to keep the setup simple and avoid extra hyperparameters on a tiny dataset.

We retain AdamW with the standard decay, and keep the cosine LR scheduler with warmup ratio = 0.06 as in general training, but use a smaller base LR ($\alpha_0 = 2.0 \cdot 10^{-5}$). With few GBV samples, smaller steps reduce drift from the well-initialized checkpoint while cosine decay preserves smooth late updates.

6.4 Models evaluation

We evaluate the models on their respective test splits. For the GBV model, the test set is small and heavily imbalanced, so per-class metrics are volatile and confidence in comparisons is limited. We therefore report all scores but interpret GBV results with caution.

Class	Precision	Recall	F1	Support
Anger	0.7547	0.8451	0.7973	142
Anticipation	0.5714	0.4000	0.4706	10
Fear	0.9634	0.8587	0.9080	92
Joy	0.8312	0.6737	0.7442	95
Love	0.5000	0.3103	0.3830	29
Neutral	0.8419	0.8458	0.8438	214
Sadness	0.7222	0.8053	0.7615	113
Surprise	0.8462	0.7213	0.7788	61
Trust	0.3556	0.6400	0.4571	25
Macro avg	0.7096	0.6778	0.6827	781
Weighted avg	0.7903	0.7785	0.7796	781

Table 6.3: General model - classification report on the test set.

General training

The model reaches accuracy ≈ 0.779 and macro-F1 ≈ 0.683 on the test set of general data. Performance is strong on high-support or well-marked classes (e.g. Neutral, Anger, Fear), while classes with fewer examples or fuzzier boundaries (Love, Anticipation, Trust) get worse results (see Table 6.3 for details).

Fear (F1 ≈ 0.91) and Anger (F1 ≈ 0.80) show high precision and recall, suggesting that strongly emotive lexical cues are well captured. Neutral is robust (F1 ≈ 0.84) despite lexical variety, likely helped by its big support.

Anticipation (F1 ≈ 0.47) and Trust (F1 ≈ 0.46) underperform, driven by small support and semantic ambiguity. This is probably due to the fact that future-oriented or confidence-related signals are more context-dependent than the other emotions. Love (F1 ≈ 0.38) also suffers. That could be because positive valence can be confused with Joy.

GBV training

With the 7-class GBV label space, the model attains accuracy ≈ 0.778 and macro-F1 ≈ 0.735. Given the very small test set, these figures are should be read as indicative; small changes in a handful of items can shift per-class F1 substantially (see Table 6.4 for details).

Anger (F1 ≈ 0.87) and Sadness (F1 ≈ 0.90) are strong and consistent, as expected in GBV context. Love (F1 ≈ 0.76) and Joy (F1 ≈ 0.71) behave reasonably.

Fear has only 2 examples in the test set, and Surprise has only 5: this is too small for stable estimation. Neutral (F1 ≈ 0.63) is also noisy; neutral posts in GBV contexts often contain mixed signals (mostly factual reporting), complicating decisions.

Class	Precision	Recall	F1	Support
Anger	0.9200	0.8214	0.8679	28
Fear	1.0000	0.5000	0.6667	2
Joy	0.6818	0.7500	0.7143	20
Love	0.7097	0.8148	0.7586	27
Neutral	0.6667	0.6000	0.6316	10
Sadness	0.9333	0.8750	0.9032	16
Surprise	0.6000	0.6000	0.6000	5
Macro avg	0.7874	0.7087	0.7346	108
Weighted avg	0.7885	0.7778	0.7794	108

Table 6.4: GBV model - classification report on the test set.

7. Inference on not-annotated data

In this chapter we test our emotion classifier on real-world, unlabeled Italian posts collected from major social platforms during Q4 2023. The period was chosen because online conversation in Italy was strongly marked by the public reaction to the Giulia Cecchettin femicide in November 2023[1], which triggered a lot of gender-based violence related content.

7.1 Dataset

The evaluation corpus contains 70 103 Italian posts gathered from Facebook, Instagram, and Twitter/X. The dataset includes raw text, metadata, and machine-predicted emotion tags produced by an IRIDE®(Almawave) classifier. That external model classifies only 6 emotions (Table 7.0a) and has been trained on Italian tweets sampled in 2020 Q3. Its label set differs from our setup, the training data and period differ substantially, and the architecture and annotation protocol are not the same, so we do not treat those tags as gold labels and the model as a benchmark. We use them only for side-by-side comparison to help interpret patterns and disagreements.

(a) IRIDE model's label distribution.

Emotion	Count	Share %
Anger	13 684	19.5
Fear	559	0.8
Joy	14 773	21.1
Neutral	27 619	39.4
Sadness	12 110	17.3
Surprise	1358	1.9

(b) Our model's label distribution.

Emotion	Count	Share %
Anger	6274	8.9
Fear	182	0.3
Joy	31 612	45.1
Neutral	23 045	32.9
Sadness	8260	11.8
Surprise	730	1.0

Table 7.1: Predicted labels.

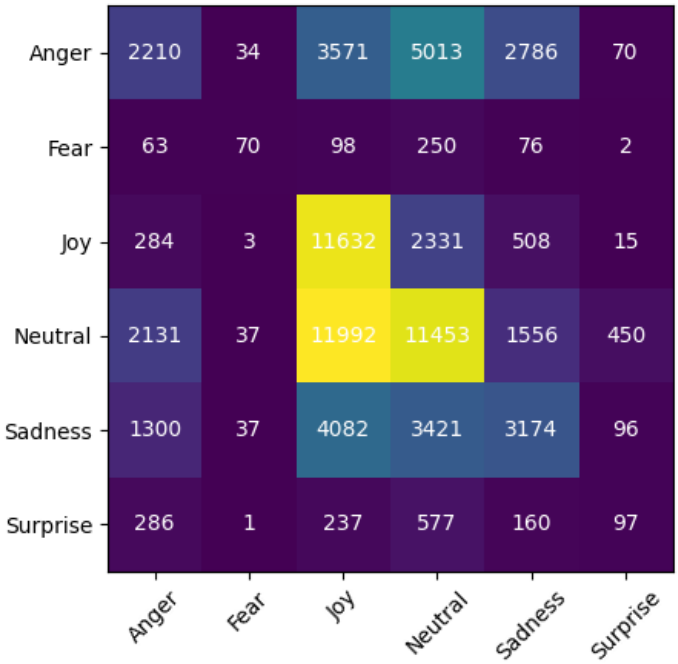

Figure 7.1: Confusion matrix between IRIDE (rows) and our model (columns) predictions.

7.2 Prediction results

Agreement with IRIDE

To give a better interpretation of the agreement and disagreement between the two models, we should refer to the training datasets label distributions (Table 6.2). In the general train split, Neutral was the most frequent class and Anger and Trust were also well represented. In the GBV train split, Anger and Love dominated and Fear was very scarce. The large column totals for Joy and Neutral in the confusion matrix are expected for two reasons: first, data comes from mixed platforms and includes many non-GBV posts that our model tends to resolve into broad categories; and second, when aligning to IRIDE's six-label setting the love predictions from our model are absorbed under joy, inflating that column. The cross-flows between IRIDE Joy and our Neutral, and vice versa, are consistent with differences in label definitions and in domain: our model was trained on multiple sources and then adapted on GBV, which encourages it to separate neutral reporting about GBV from positive affect, whereas IRIDE's older, single-platform model may encode different boundaries. The row for IRIDE Anger often maps to our Neutral or Joy. Part of this is likely topical framing (activism or news items carrying negative lexis but weak target-directed hostility), and part is class-prior shift, since our GBV adaptation emphasizes anger when explicit cues are present, but defaults to Neutral otherwise. Fear and Surprise remain small and noisy in both systems, which matches their low prevalence in training.

8. Conclusions and perspectives

This thesis set out to build and assess a reliable Italian emotion classifier for social-media texts, with a focus on gender-based violence. We framed the task around a BERT-style encoder, compared Italian backbones, designed a two-stage fine-tuning pipeline, and evaluated the resulting model on its test set and on a large, unlabeled dataset from 2023.

The main objective was met. We established a reproducible data pipeline, aligned heterogeneous sources into a consistent label space, and motivated the backbone choice in favor of ALBERTo for social-media alignment. Methodologically, we investigated imbalance-conscious training, regularization, and optimization choices, documenting their impact on stability and performance.

Three constraints shaped the outcomes. First, data scarcity and imbalance, particularly in GBV, limited adaptation on rare emotions and increased variance in validation/test metrics. Second, label-set and domain shifts across sources introduced boundary ambiguities, especially between neutral and positive classes. Third,

[1] https://en.wikipedia.org/wiki/Murder_of_Giulia_Cecchettin.

small per-class supports in GBV testing led to unreliable generalization estimates, because a few misclassifications can move metrics substantially. From a modelling standpoint, while light dynamic oversampling improved minority gradients without excessive overfitting, it only mitigated the imbalance.

The most impactful lever is data. Curating additional GBV annotations, targeting currently under-represented emotions and introducing anticipation and trust, would stabilize metrics and enable better analysis. Active learning and uncertainty sampling could prioritize ambiguous posts (e.g. in news/activism contexts) and reduce annotation cost. Domain-adaptive pretraining on recent Italian social streams is likely to improve robustness to platform slang and evolving hashtags.

One perspective for the future is to extend the task formulation: from single-label to multi-label emotions when text clearly carries mixed affect, and from pure emotion to joint modelling with GBV-specific markers (e.g. harassment types), improving practical utility. Another is to extend to more modalities and time: incorporate image or link context when available, and implement periodic re-training with drift detection to keep pace with linguistic change. In the longer run, a human-in-the-loop pipeline, combining continuous weakly supervised inference, targeted manual checks, and scheduled domain adaptation, would support sustained monitoring with transparent updates. The infrastructure developed here provides a solid base to pursue these extensions as richer GBV datasets become available.

In sum, the thesis delivers a principled, end-to-end approach to Italian emotion detection with GBV focus, demonstrates competitive results under realistic constraints, and identifies concrete, data-centric ways for improving coverage, stability, and interpretability in future iterations.

Acknowledgments

Araque, O., S. Frenda, R. Sprugnoli, D. Nozza, and V. Patti (2023). "EMit at EVALITA 2023: Overview of the Categorical Emotion Detection in Italian Social Media Task". *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2023) – CEUR Workshop Proceedings*. Vol. 3473. CEUR Workshop Proceedings, 1–8. URL: https://ceur-ws.org/Vol-3473/paper1.pdf.

Davidson, T., D. Warmsley, M. Macy, and I. Weber (2017). "Automated Hate Speech Detection and the Problem of Offensive Language". *arXiv preprint arXiv:1703.04009*. URL: https://arxiv.org/abs/1703.04009.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 4171–4186. URL: https://arxiv.org/abs/1810.04805.

Di Donfrancesco, S. (2025). *Emotion Detection on hate speech and gender-based violence texts on social media*. https://github.com/Sergio-ddf/internship-istat-ensai.

EIGE (2022). *Combating Cyber Violence Against Women and Girls*. Tech. rep. European Institute for Gender Equality. URL: https://eige.europa.eu/sites/default/files/documents/combating_cyber_violence_against_women_and_girls.pdf.

Fersini, E., D. Nozza, and P. Rosso (2018). "AMI @ EVALITA2018: Automatic Misogyny Identification". *Proceedings of EVALITA 2018*.

Fontanella, L., B. Chulvi, E. Ignazzi, A. Sarra, and A. Tontodimamma (2024). "How do we study misogyny in the digital age? A systematic literature review using a computational linguistic approach". *Humanities and Social Sciences Communications* 11. Article 478, 1–17. URL: https://www.nature.com/articles/s41599-024-02978-7.

Hu, E. J. et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. URL: https://arxiv.org/abs/2106.09685.

Lazer, D. et al. (2009). "Computational Social Science". *Science* 323.5915, 721–723. URL: https://www.science.org/doi/10.1126/science.1167742.

Peters, M. E. et al. (2018). "Deep Contextualized Word Representations". *Proceedings of NAACL-HLT 2018*, 2227–2237. URL: https://aclanthology.org/N18-1202/.

Plutchik, R. (1980). "Chapter 1 - A GENERAL PSYCHOEVOLUTIONARY THEORY OF EMOTION". *Theories of Emotion*. Ed. by R. Plutchik and H. Kellerman. Academic Press, 3–33. DOI: https://doi.org/

10.1016/B978-0-12-558701-3.50007-7. URL: https://www.sciencedirect.com/science/article/pii/B9780125587013500077.

Poletto, F., V. Basile, M. Sanguinetti, C. Bosco, and V. Patti (2021). "Resources and Benchmark Corpora for Hate Speech Detection: A Systematic Review". *Language Resources and Evaluation* 55, 477–523. URL: https://link.springer.com/article/10.1007/s10579-020-09502-8.

Sellars, A. (Dec. 2016). *Defining Hate Speech*. Research Publication 2016-20. Also published as Boston University School of Law, Public Law Research Paper No. 16-48. Berkman Klein Center for Internet & Society. DOI: 10.2139/ssrn.2882244. URL: https://ssrn.com/abstract=2882244.

Song, B., Z. Li, X. Lin, J. Wang, T. Wang, and X. Fu (2021). "Pretraining model for biological sequence data". *Briefings in Functional Genomics* 20.3, 181–195. DOI: 10.1093/bfgp/elab025. URL: https://doi.org/10.1093/bfgp/elab025.

Stevens, F. et al. (2024). "Women are less comfortable expressing opinions online than men and report heightened fears for safety: Surveying gender differences in experiences of online harms". *arXiv:2403.19037*.

Tontodimamma, A., E. Nissi, A. Sarra, and L. Fontanella (2020). "Thirty years of research into hate speech: topics of interest and their evolution". *Scientometrics* 126.1, 157–179. URL: https://link.springer.com/article/10.1007/s11192-020-03737-6.

UNESCO and ICFJ (2022). *The Chilling: A Global Study of Online Violence Against Women Journalists*. Tech. rep. UNESCO Publishing. URL: https://www.icfj.org/sites/default/files/2022-11/ICFJ_UNESCO_The%20Chilling_2022_1.pdf.

Usuga Cadavid, J. P., S. Lamouri, B. Grabot, and A. Fortin (July 2021). "Using deep learning to value free-form text data for predictive maintenance". *International Journal of Production Research* 60, 1–28. DOI: 10.1080/00207543.2021.1951868.

Vaswani, A. et al. (2017). "Attention Is All You Need". *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 5998–6008. URL: https://papers.neurips.cc/paper/7181-attention-is-all-you-need.pdf.