

# EmotivITA-BERT- VAD

Predizione delle dimensioni emotive

---

Valence

---

Arousal

---

Dominance

# Obiettivo del Progetto

## Modello VAD

Valence (piacevolezza), Arousal (attivazione), Dominance (controllo) su scala 1-5

## Dataset

EmoITA: corpus di frasi italiane annotate con valori VAD

## Approccio BERT

Utilizzo di dbmdz/bert-base-italian-cased per regressione continua

Il progetto si concentra sulla predizione delle tre dimensioni emotive attraverso tecniche di deep learning, confrontando due strategie architetturali differenti per la gestione della regressione multi-target.

# Due Approcci: Task A vs Task B

## Task A: Regressione Single-Target

- Tre modelli BERT separati (uno per dimensione)
- Ogni modello predice una sola dimensione
- Specializzazione per ciascuna emozione
- Training indipendenti

## Task B: Regressione Multi-Target

- Un singolo modello BERT condiviso
- Predizione simultanea di V, A e D
- Efficienza computazionale superiore
- Apprendimento congiunto

📄 **Differenza critica nello split:** Task A crea file separati (train\_valence.csv, train\_arousal.csv, train\_dominance.csv) con split 80/20 ripetuto per dimensione. Task B produce un unico train.csv e val.csv contenenti tutte e tre le dimensioni.

# Architettura Tecnica del Modello



## Input Testuale

Frasi italiane con lunghezza massima 64-128 token



## Tokenizzazione BERT

Generazione input\_ids e attention\_mask



## BERT Encoder

12 layer transformer, primi 10 congelati, ultimi 2 addestrati



## Token [CLS]

Aggregazione in vettore 768 dimensioni



## Regressore Lineare

Linear(768→1) per Task A o Linear(768→3) per Task B



## Output VAD

Valori predetti in scala 1-5 (denormalizzazione ×5)

Durante il training, le etichette sono normalizzate nell'intervallo [0,1] dividendo per 5, per facilitare la convergenza del modello.

# Scelte di Training

## Funzione di Loss

**SmoothL1Loss:** meno sensibile agli outlier rispetto a MSE, combina robustezza di L1 con smoothness di L2.

## Ottimizzatore

**Adam** con learning rate adattivo:  $2e-5$  per Task A,  $3e-5$  per Task B. Ideale per addestramento di BERT.

## Regolarizzazione

**Dropout 0.3** sul token [CLS] per ridurre overfitting. Seed 42 per garantire riproducibilità completa.

## Iperparametri

- Batch size: 16
- Max epoche: 8
- Early stopping: patience 3
- Salvataggio modello migliore

# Task A: Regressione Single-Target

1

## Preparazione Dati

Script `prepare_taskA_datasets.py` crea split 80/20 sul Development set, generando file separati per ogni dimensione:  
`train_valence/val_valence`, `train_arousal/val_arousal`,  
`train_dominance/val_dominance`

2

## Architettura

Tre modelli BERT identici, ognuno con regressore lineare  $768 \rightarrow 1$ .  
Specializzazione per singola dimensione emotiva

3

## Training

Tre sessioni di training separate, salvataggio modelli `valence.pth`, `arousal.pth`, `dominance.pth`

4

## Valutazione

Output `model_*_results.txt` con MAE e Pearson r per ciascuna dimensione su test set con gold labels

# Task B: Regressione Multi-Target

01

## Split Dataset Unico

Script `prepare_dataset.py` produce `train.csv` e `val.csv` (80/20) con colonne `text`, `Valence`, `Arousal`, `Dominance`

03

## Training Singolo

Una sessione di training, salvataggio modello unico (`model.pth`)

02

## Modello Condiviso

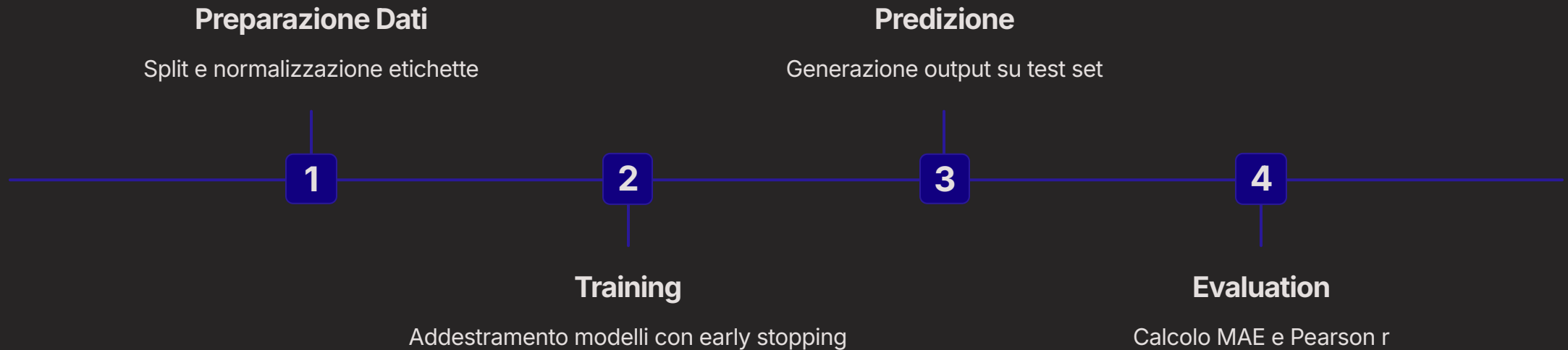
Un solo BERT con regressore  $768 \rightarrow 3$  che predice simultaneamente `V`, `A`, `D`

04

## Predizione e Valutazione

Test su `Test set.csv`, confronto con `Test set - Gold labels`. Output `risultati.txt` con metriche complete

# Pipeline di Valutazione



## Metriche di Performance

- **MAE** (Mean Absolute Error): errore medio assoluto su scala 1-5
- **Pearson r**: correlazione lineare con valori gold

## Script di Analisi

- `comparativa_taskA.py`: grafico V vs A vs D
- `plot_loss.py`: curve loss training/validation
- `comparativa_AB.py`: confronto Task A vs Task B

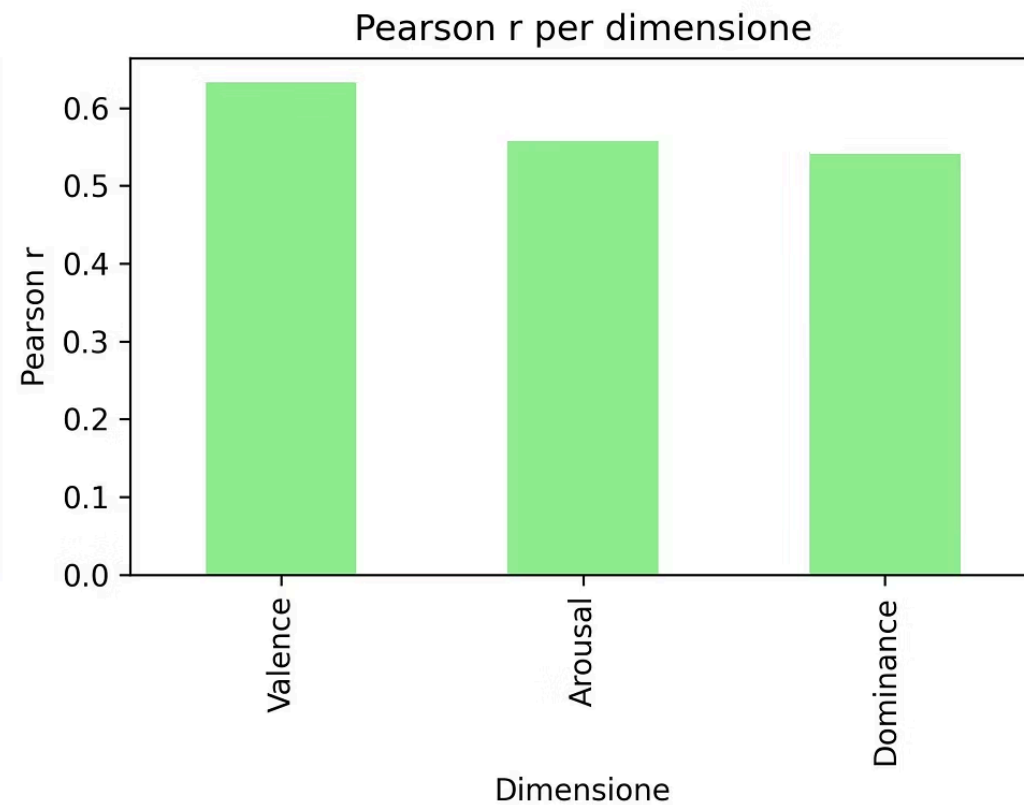
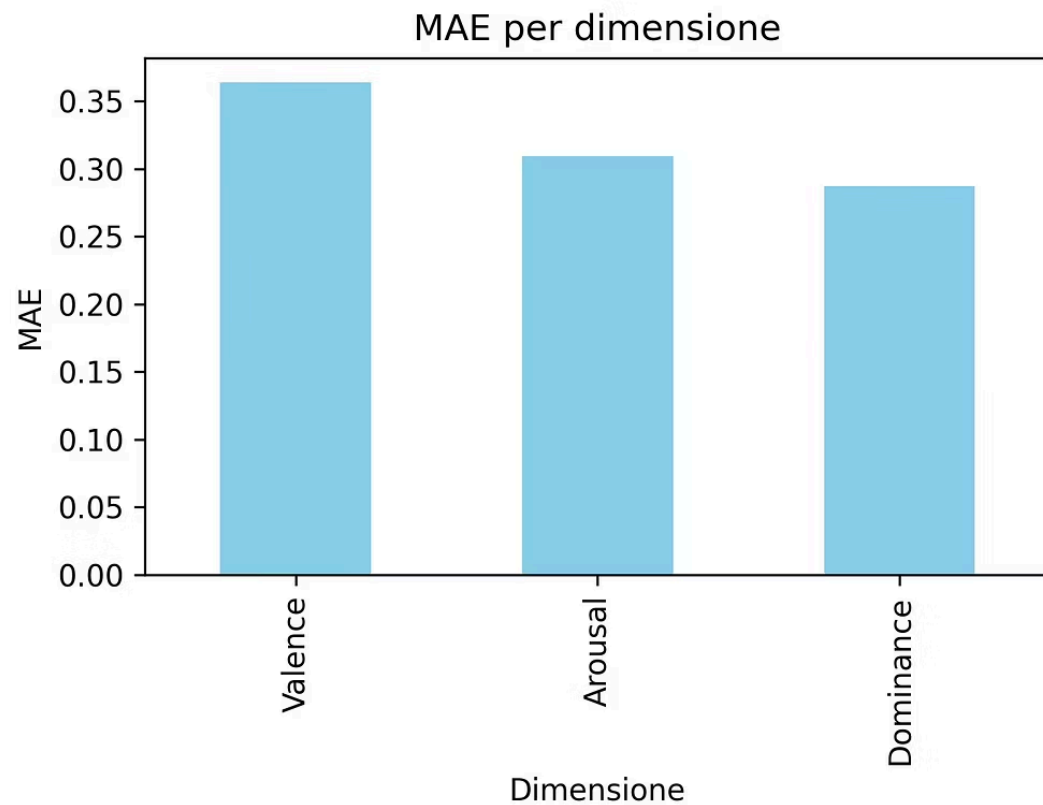


# Risultati Sperimentali

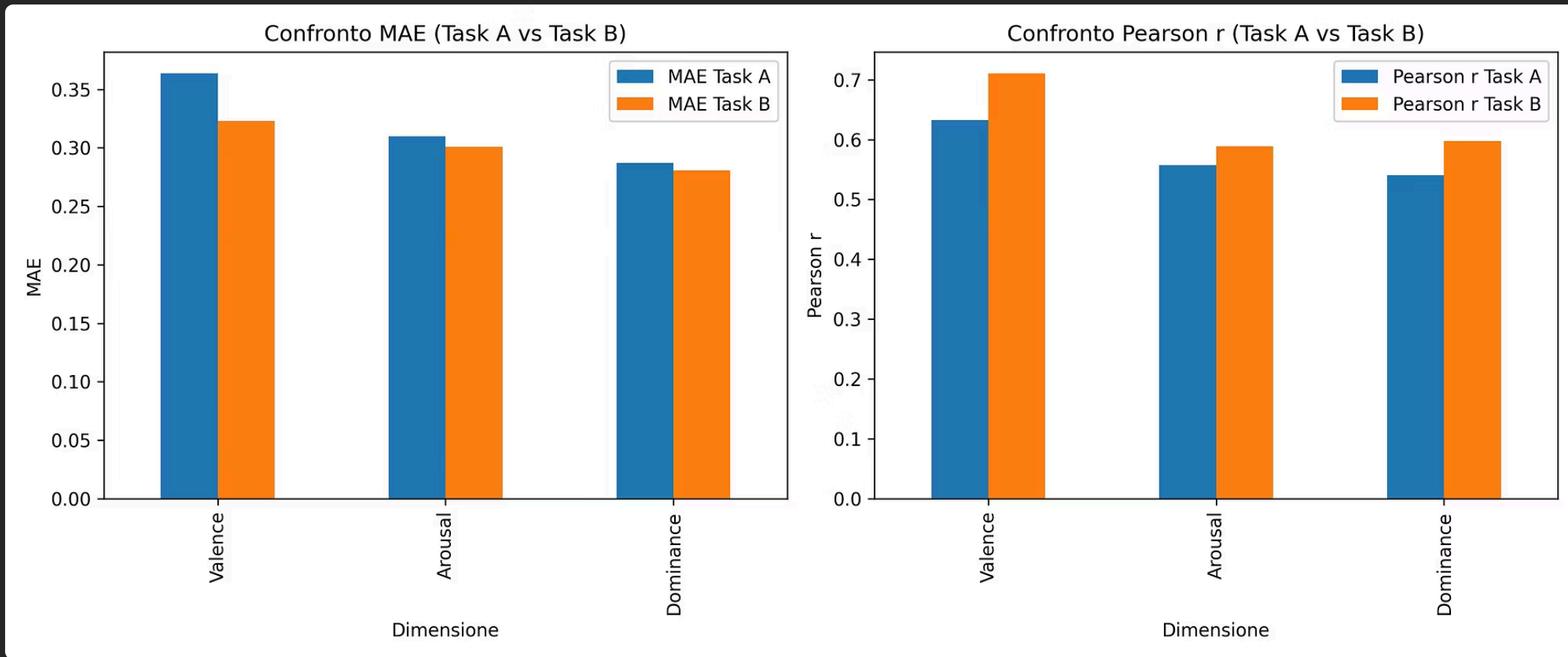
Dimensione	Approccio	MAE	Pearson r
Valence	Task A	0.36	0.63
Arousal	Task A	0.31	0.56
Dominance	Task A	0.29	0.54
Valence	Task B	0.32	0.71
Arousal	Task B	0.30	0.59
Dominance	Task B	0.28	0.60

I risultati mostrano correlazioni Pearson r nella fascia 0.54–0.71 e MAE tra 0.28 e 0.36 su scala 1-5. **Valence** in Task B raggiunge la correlazione più elevata ( $r=0.71$ ), mentre **Dominance** presenta sempre il MAE più basso.

# Grafico comparativo Task A



# Grafico comparativo Task A vs Task B



# Conclusioni e Sviluppi Futuri

## Risultati Ottenuti

Implementazione completa di due approcci per regressione VAD su testo italiano. Performance consistenti con letteratura NLP: Pearson  $r$  0.54-0.71, MAE 0.28-0.36.

## Trade-off Analizzati

Task A: modelli specializzati ma training triplo. Task B: efficienza computazionale ma condivisione parametri. Nessun approccio dominante assoluto.

## Riproducibilità

Scelte tecniche documentate (Adam, SmoothL1Loss, early stopping, dropout). Seed 42. README completo con setup e pipeline.

## Estensioni Possibili

- Modelli transformer più avanzati (RoBERTa, DeBERTa)
- Dataset più ampi e diversificati
- Ottimizzazione iperparametri con ricerca sistematica
- Analisi errori e casi limite

**Grazie per l'attenzione!**