Sarcasm Detection con SLM & Multitask Learning

- Domenico De Marchis
- Francesco Fontanesi

DEEP LEARNING GENNAIO 2026

OBIETTIVO

Generalità

Problema Chiave

Il sarcasmo richiede contesto e informazioni spesso non disponibili: Sarcasmo Implicito e Intended

Ipotesi

Il Fine - Tuning di modelli più piccoli può permettere di superare LLM in zero - shot

Modelli Analizzati

- **ModernBERT**: encoder-only, evoluzione di BERT
- **Phi-3**: decoder-only SLM open-source
- **Llama-3.3-70b & Kimi-K2**: LLM benchmark via API Groq

OBIETTIVO

Pre - Concetti

ModernBERT

- più contesto, RoPE, Alternating Attention, Unpadding e GeGLU
- la classificazione del testo sfrutta sempre $[CLS]$

Phi3

- più contesto, RoPE, RMSNorm, SwiGLU, Alternating BlockSparse Attention
- la classificazione del next token T sfrutta l'embedding del token $T-1$

LoRA

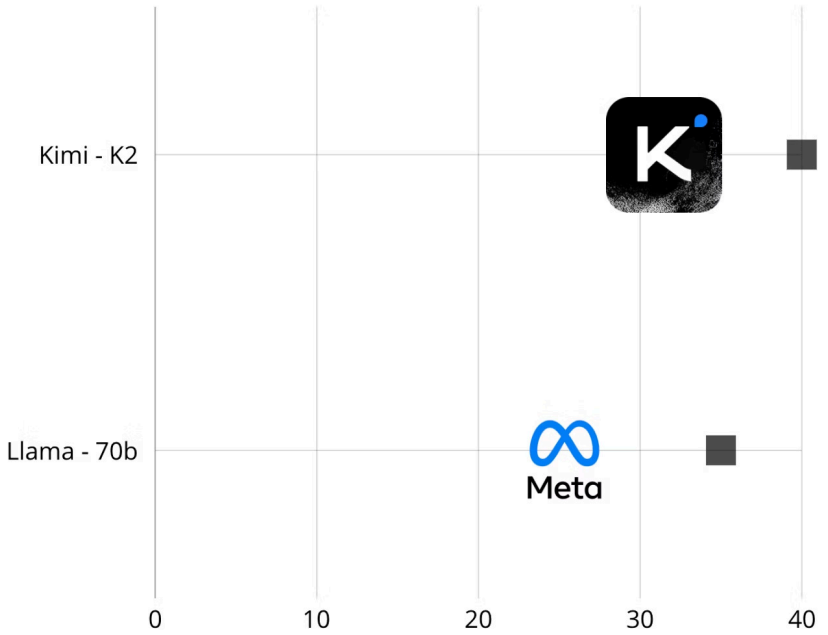
- si sfrutta una correzione di basso rango:

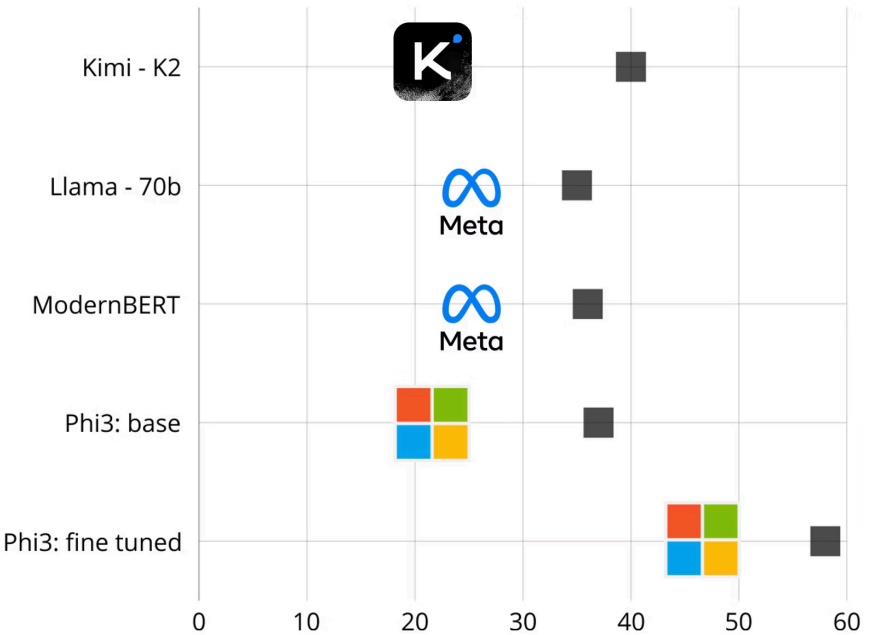
$$W' = W + \Delta W, \Delta W = BA$$

- **Idea**: pochi parametri per adattare l'espressività

OBIETTIVO

Baseline e Risultati - 1





OBIETTIVO

Baseline e Risultati - 2

98%

Recall Llama-3.3

Trova quasi tutto il sarcasmo

21%

Precision Llama

Moltissimi falsi positivi

733

Falsi Positivi

Su 1200 esempi non sarcastici

Comportamento Zero-Shot: I LLM tendono a sovrastimare massivamente la classe "Sarcastic"

DATASET

iSarcasmEval

1	**Sbilanciamento 1:3** Rapporto significativo tra classe sarcastica e non sarcastica, principalmente tweets
2	**Scelta Deliberata** Nessuna integrazione Reddit per evitare vocabulary mismatch (slang, stile diverso)
3	**Rephrasing Unico** Versioni parafrasate non sarcastiche per esempi sarcastici - base per approccio multitask

DATASET

La Scelta

1 Data Augmentation

- fatto dagli autori stessi
- permette stime più solide
- Vocabulary Mismatch Problem

2 Impostazione del Problema

- usare i dati che abbiamo
- costruire task ausiliari
- peso maggiore ai positives

Per gli obiettivi di un progetto come questo, abbiamo esplorato la seconda alternativa

METODOLOGIA

Strategie di Training

Positive Weight

Peso maggiore alla loss su record positivi (rapporto 1:3)

Multitask Learning

Diversi Task ausiliari: tutti sfruttano il rephrasing del dataset

Loss Combinata

Combinazione lineare a peso configurabile: $L = \lambda_A L_A + \lambda_B L_B$

METODOLOGIA

Auxiliary Tasks

La filosofia adottata è sempre quella di un **Hard Parameter Sharing**

1	**Main Task** Dato il Testo, prevedere se sia Sarcastic / Non Sarcastic
2	**Auxiliary Task A** Data la coppia $(Testo, Rephrase)$, identificare quale dei due sia la versione Sarcastic
3	**Auxiliary Task B** Data la coppia $(Testo, Rephrase)$, spingere gli score prodotti ad essere distanti di almeno una quantità m
4	**Auxiliary Task C** Dato il Testo, generare la sua versione Rephrased (solo per Phi-3)

METODOLOGIA

Auxiliary Losses

La base è sempre una **Cross Entropy**: $L(s, y) = -log(p(y))$ o **Binary** $L(s, y) = [-\alpha \cdot y \cdot log(\sigma(s)) + (1 - y) \cdot log(1 - \sigma(s))]$

Main Task & Task A

- Funzionamento classico
- Score s alto positivo per $y = 1$, viceversa per $y = 0$.

Task B

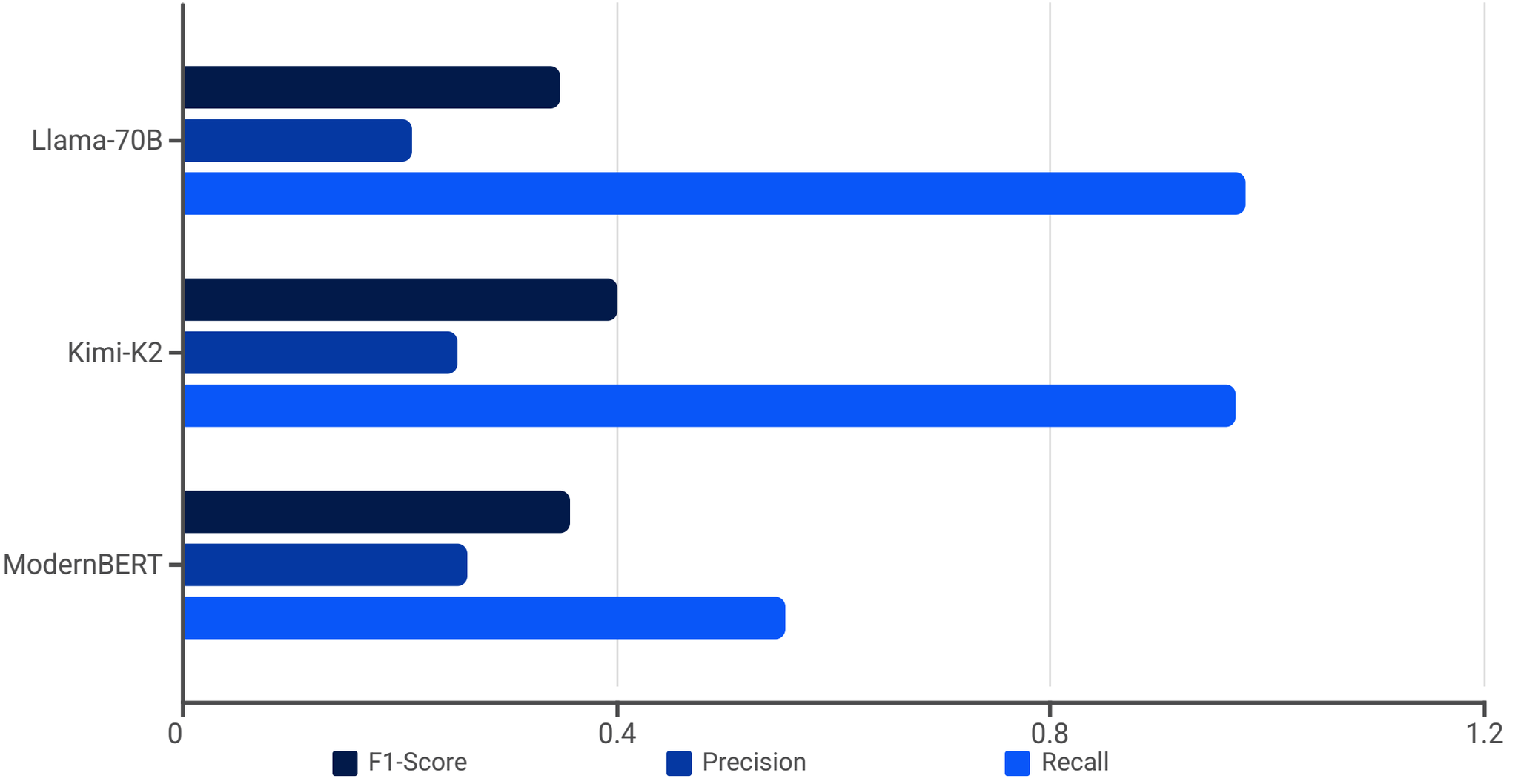
- Praticamente una **Contrastive Loss**
- Massimizza distanza tra gli score: $L(s_a, s_b) = max(0, m - (s_a - s_b))$

Task C

- **Generative Loss**
- CE rispetto la distribuzione del target: $\mathcal{L} = \frac{\sum_{i,t} w_i \cdot 1_{\text{active}}(i,t) \cdot \text{CE}(\hat{y}_{i,t}, y_{i,t})}{\sum_{i,t} w_i \cdot 1_{\text{active}}(i,t)}$

MODERNBERT

Risultati



ModernBERT supera LLama, ma tra tutte le configurazioni provate Kimi rimane un tetto

MODERNBERT

Ablazione

ALL

Collasso sulla classe maggioritaria (*Non Sarcastic*): training bloccato

WEIGHTS

Il Task B impedisce il collasso. Tetto prestazionale: poco segnale

TASK B

Ablazione simile a Weights, è il lavoro congiunto che dà il Gain

Model	F1 - Score	Precision	Recall	Threshold	Auxiliary Task
modernbert_o1	0.25	0.14	0.96	0.5	Task A
modernbert_o2_only_task_main	0.24	0.17	0.43	0.75	None
modernbert_o2_second	0.36	0.26	0.55	0.75	Task B
modernbert_o2_ablation_weight	0.27	0.28	0.25	0.75	Task B
modernbert_o2_ablation_all	0.08	0.24	0.05	0.5	None

Nota: la Tabella riporta Performance sulla classe 'Sarcastic'

METODOLOGIA

Threshold Tuning

Il Validation Set è piccolo e rumoroso: non possiamo ricavare stime significative, usiamo $t = 0.5$

1 **Dataset Sbilanciato**

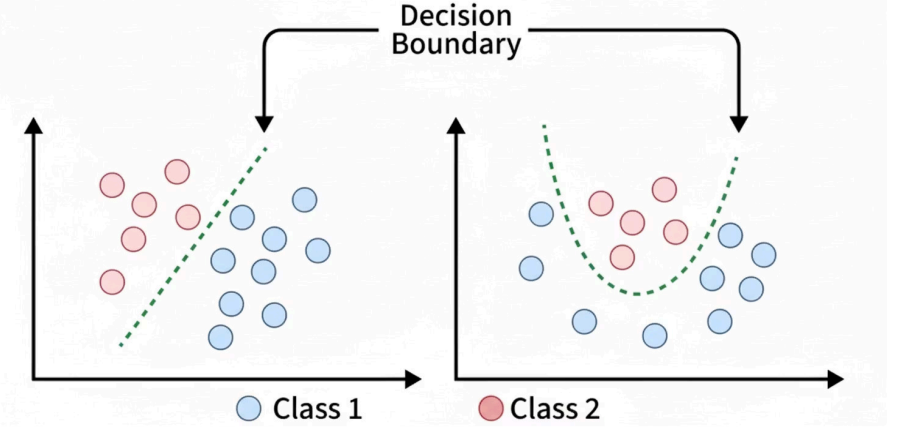
Il parametro pos_weight permette di bilanciare la distribuzione del gradiente

2 **Soglia Teorica**

Con $pos\_weight = 3.0$, soglia neutrale si sposta a 0.75

3 **Calibrazione**

$$p_{real} = \frac{p_{weighted}}{p_{weighted} + (1 - p_{weighted})W}$$



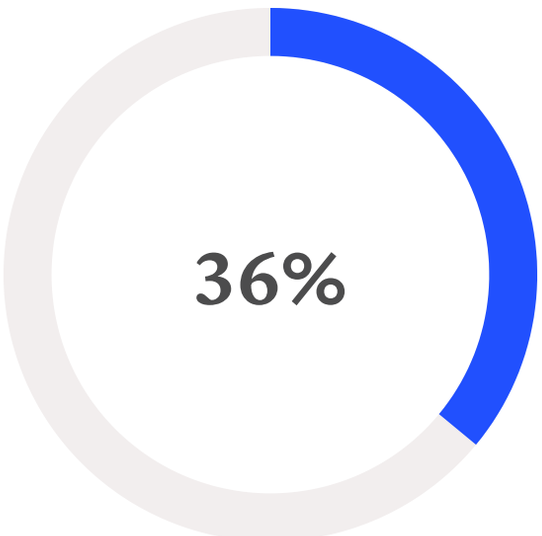
MODERNBERT

ModernBERT: Conclusioni

1

Risultato

Prestazioni discrete, performa similmente ad un 70B in zero - shot ed i parametri allenati sono irrisori



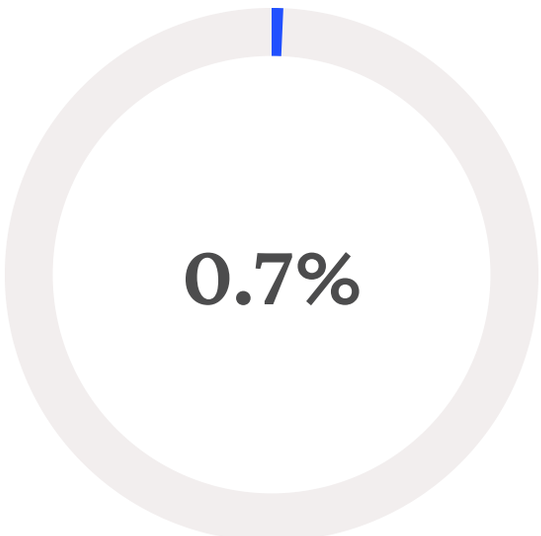
Best Score F1

Margin Loss Task B + Positive Weighting

2

Problematiche

Il Task è molto complesso, difatti gli autori raggiungono prestazioni migliori con ensemble



Parametri Trained

LoRA su ModernBERT totale

PHI-3

Decoder-Only: Approccio Generativo

01

Predizione Generativa

Modello genera etichetta esplicita (A=sarcastico, B=non sarcastico)

02

Analisi Logit

Valutazione di confidenza, soglia τ su $\Delta = log(p(A)) - log(p(B))$

03

Task C: Rewriting

Il modello impara direttamente il Rephrasing… puoi farlo qui!

⚠ Con ModernBERT alleniamo di fatto una NUOVA TESTA, non con Phi-3!

PHI-3

Ablazione

WEIGHTS

Interessante caso: la configurazione migliore è con un $pos\_weight = 1$, un ulteriore peso ai positivi causa drift

TASK C

Il Task C di Rephrasing si dimostra importante per un Gain prestazionale, ma solo con opportuno peso

Model	F1 - Score	Precision	Recall	Threshold	Setup
phi_ablation_c	0.50	0.37	0.76	1.10	No Task C
phi_o1	0.58	0.61	0.56	0.00	No Pos. Weights
phi_o2	0.52	0.42	0.69	1.10	Task C + Pos. Weights
phi_eq_weight	0.27	0.28	0.25	1.10	Task C + Pos. Weights

Nota: ci sono meno configurazioni che con ModernBERT, puramente per bottleneck hardware

PHI-3

Domain Weight Shift

Perché il peso positivo causa drift?

- In **ModernBERT** l'ultimo layer (head) è **inizializzato casualmente** e deve imparare **da zero** il decision boundary, qui la LM Head è fissa
- in BERT pos_weight serve a "trovare" la minoritaria; in Phi-3 può **creare confusione** perché interferisce con una **distribuzione prior già presente**

Non c'è una spiegazione esatta, quella fornita è (per noi) la più plausibile

PHI-3

Conclusioni: Phi-3 Vince 🏆

58%

F1-Score Phi-3

Multitask fine-tuned (best)

60%

Precision

+35 punti vs Kimi

0.23%

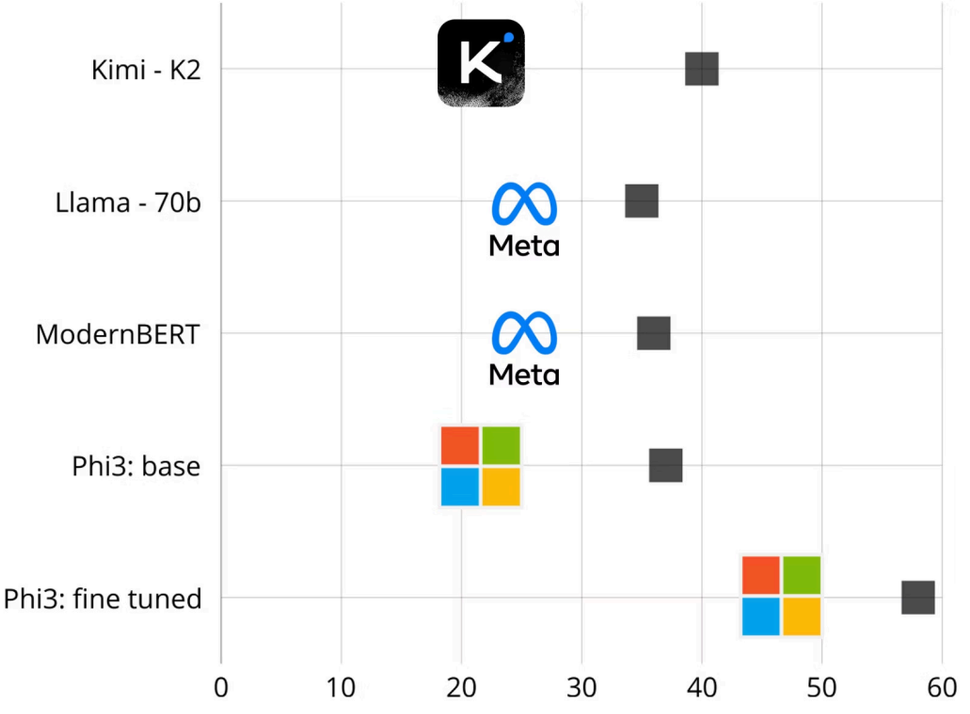
Parametri Trained

9M su 3.8B totali con LoRA

60%

Best F1 iSarcasmEval

Ensemble di BERT - based



È possibile superare le prestazioni di LLM con SLM Fine - Tuned per Sarcasm Detection?
Si e di molto!

ESTENSIONI

Reddit

Obiettivo

Far vedere perché i LLM sono spesso preferibili ed il problema del Fine Tuning: zero - shot e domain shift

Reddit

Lo stile è lo stesso dei Tweet, cambia il vocabolario e le prestazioni degradano rispetto ad iSarcasmEval

Model	F1 - Score	Precision	Recall	Setup
phi_3	0.51	**0.66**	0.42	Base
phi_3	0.55	0.62	0.49	Fine Tuned iSarcasmEval
modernbert	0.43	0.62	0.33	Fine Tuned iSarcasmEval
kimi_k2 🏆	**0.680**	0.58	0.80	Zero - Shot
llama_70b 🏆	**0.684**	0.55	0.89	Zero - Shot

Il Fine - Tuning è Domain Specific, i benifici vengono mitigati da un domain shift