FILE YAML:  
yaml\_config\_template = """

model\_type: llm

base\_model: {model\_path}

input\_features:

  - name: DOMANDE

    type: text

    preprocessing:

      truncation: true

      max\_sequence\_length: 140

      lowercase: true

output\_features:

  - name: RISPOSTE

    type: text

    preprocessing:

      max\_sequence\_length: 200

      truncation: true

    metrics:

      - type: bert\_score

        model: dbmdz/bert-base-italian-uncased

      - type: rouge

      - type: perplexity

      - type: loss

prompt:

  template: >-

    Sei un esperto cardiologo. Rispondi alla seguente domanda in modo empatico e con frasi complete e concise, terminando ogni frase con un punto.

    ### Esempio 1:

    Domanda: Quali sono i sintomi dell'ipertensione?

    Risposta: I sintomi dell'ipertensione includono mal di testa, vertigini e difficoltà respiratorie.

    ### Esempio 2:

    Domanda: Come posso controllare la mia pressione arteriosa?

    Risposta: Puoi controllare la tua pressione arteriosa con una dieta sana, esercizio fisico regolare e farmaci prescritti.

    ### Domanda: {DOMANDE}

    ### Risposta (Assicurati che la tua risposta termini con un punto e abbia senso compiuto):

generation:

  temperature: 0.1    ### per la generazione del testo deve essere molto basso

  max\_new\_tokens: 40

  top\_p: 0.9          ### aiuta a generare testo più coerente e fluido. , il modello avrà un equilibrio tra casualità e coerenza, garantendo che i token generati siano tra quelli con le probabilità più alte.

adapter:

  type: lora

quantization:

  bits: 4

### sessione di validazione

preprocessing:

  split:

    type: random

    probabilities: [0.6, 0.2, 0.2]

  global\_max\_sequence\_length: 340

### sessione di validazione

trainer:

  type: finetune

  epochs: 100

  batch\_size: 8

  eval\_batch\_size: 4                        ### indica la dimensione dei batch utilizzata durante la fase di valutazione del modello. (I dati verranno processati in batch di 4 campioni alla volta) >>> ottimizzazione delle risorse.

  gradient\_accumulation\_steps: 2

  learning\_rate: 3.0e-5                     ### definisce la velocità di apprendimento iniziale.

  enable\_gradient\_checkpointing: true

  weight\_decay: 0.01                        ### definisce il tasso di decadimento del peso, che è una tecnica che riduce progressivamente la velocità di apprendimento. (una tecnica che aiuta a prevenire l'overfitting)

  dropout\_rate: 0.1                         ### specifica la probabilità di dropout, che aiuta a prevenire l'overfitting disattivando casualmente unità (neuroni) durante l'allenamento. (10% disattivato)

  learning\_rate\_scheduler:                  ### è la sezione che configura come la velocità di apprendimento viene adattata durante l'allenamento.

    decay: cosine                              ### indica che viene utilizzata una strategia di decadimento coseno. (riduce la velocità di apprendimento seguendo una curva coseno, favorendo un adattamento più dolce verso la fine dell'addestramento)

    warmup\_fraction: 0.1                       ### specifica che la velocità di apprendimento aumenterà gradualmente durante il 10% iniziale dell'addestramento.

  checkpoint\_interval: 500

### un checkpoint del modello verrà salvato ogni 500 step di addestramento, permettendo il ripristino del modello in caso di interruzioni.

early\_stopping:                             ### monitora la perdita di validazione (val\_loss) e interrompe l'addestramento se la perdita non migliora per 5 epoche consecutive, per prevenire l'overfitting (min: il monitoraggio è configurato per minimizzare la perdita).

  monitor: loss

  patience: 5

  mode: min

logging:                                    ### i log dell'allenamento verranno inviati a TensorBoard.

  log\_to\_tensorboard: true

callbacks:

  - type: progress\_bar

  - type: model\_checkpoint

    save\_best\_only: true

    monitor: loss

    mode: min

  - type: early\_stopping

    monitor: loss

    patience: 5

    mode: min

  """

LOG RESULT:  
{

"evaluation\_frequency": {

"frequency": 1,

"period": "epoch"

},

"test": {

"RISPOSTE": {

"loss": [

4.984861850738525,

4.711007595062256

],

"next\_token\_perplexity": [

25778.255859375,

25621.5859375

],

"perplexity": [

28433.421875,

28622.587890625

],

"sequence\_accuracy": [

0.0,

0.0

],

"token\_accuracy": [

0.017873430624604225,

0.016018986701965332

]

},

"combined": {

"loss": [

4.984861850738525,

4.711007595062256

]

}

},

"training": {

"RISPOSTE": {

"loss": [

6.557398319244385,

6.231690883636475

],

"next\_token\_perplexity": [

27810.97265625,

27767.59765625

],

"perplexity": [

28340.322265625,

28400.142578125

],

"sequence\_accuracy": [

0.0,

0.0

],

"token\_accuracy": [

0.021984655410051346,

0.0217923354357481

]

},

"combined": {

"loss": [

6.557398319244385,

6.231690883636475

]

}

},

"validation": {

"RISPOSTE": {

"loss": [

5.031159400939941,

4.754090309143066

],

"next\_token\_perplexity": [

25897.35546875,

25745.716796875

],

"perplexity": [

28542.8828125,

28697.3125

],

"sequence\_accuracy": [

0.0,

0.0

],

"token\_accuracy": [

0.016218744218349457,

0.014472403563559055

]

},

"combined": {

"loss": [

5.031159400939941,

4.754090309143066

]

}

}

}

Immagine che contiene testo, linea, schermata, Diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Diagramma

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Diagramma

Descrizione generata automaticamente

Immagine che contiene testo, linea, diagramma, schermata

Descrizione generata automaticamente

Immagine che contiene testo, schermata, linea, Diagramma

Descrizione generata automaticamente