"""

model\_type: llm

base\_model: {model\_path}

input\_features:

  - name: DOMANDE

    type: text

    preprocessing:

      truncation: true

      max\_sequence\_length: 210

      lowercase: true

output\_features:

  - name: RISPOSTE

    type: text

    preprocessing:

      max\_sequence\_length: 1000

      max\_length: 1000

      truncation: true

    metrics:

      - type: bert\_score

        model: dbmdz/bert-base-italian-uncased

      - type: loss

      ### - type: rouge, - type: perplexity

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

preprocessing:

  split:

    type: random

    probabilities: [0.9, 0.05, 0.05]  ### a gestione di più sottoinsiemi può richiedere risorse aggiuntive.

  global\_max\_sequence\_length: 1210

trainer:

  type: finetune

  epochs: 1

  batch\_size: 2

  eval\_batch\_size: 2

  gradient\_accumulation\_steps: 8  # effective batch size = batch size \* gradient\_accumulation\_steps

  learning\_rate: 1.0e-4  # Prova con un learning rate leggermente più basso

  enable\_gradient\_checkpointing: true

  learning\_rate\_scheduler:

    decay: cosine

    warmup\_fraction: 0.1  # Aumenta il warmup fraction

    reduce\_on\_plateau: 0

### 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: val\_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: val\_loss

    mode: min

  - type: early\_stopping

    monitor: val\_loss

    patience: 5

    mode: min

"""

Immagine che contiene testo, linea, diagramma, Diagramma

Descrizione generata automaticamente

Il grafico che hai condiviso mostra tre diverse metriche (Loss, Perplexity e Token Accuracy) per i tre dataset (Train, Validation e Test). Ecco cosa possiamo dedurre da ogni grafico:

**1. Loss per Dataset**

* **Train Loss (punto blu)**: La perdita durante l'addestramento è di circa 1.35.
* **Validation Loss (punto arancione)**: La perdita durante la validazione è di circa 1.10.
* **Test Loss (punto verde)**: La perdita durante il test è di circa 1.09.

Questa differenza tra le perdite indica che il modello sta generalizzando abbastanza bene dai dati di addestramento ai dati di validazione e test. La perdita è inferiore nei dati di validazione e test rispetto ai dati di addestramento, il che potrebbe suggerire che il modello non è sovra-addestrato.

**2. Perplexity per Dataset**

* **Train Perplexity (punto blu)**: La perplessità durante l'addestramento è di circa 31.983.
* **Validation Perplexity (punto arancione)**: La perplessità durante la validazione è di circa 31.989.
* **Test Perplexity (punto verde)**: La perplessità durante il test è di circa 31.984.

La perplessità è una misura di quanto bene il modello prevede una sequenza. Valori di perplessità simili tra i tre dataset indicano che il modello mantiene una consistenza nelle previsioni tra i dati di addestramento, validazione e test.

**3. Token Accuracy per Dataset**

* **Train Token Accuracy (punto blu)**: L'accuratezza dei token durante l'addestramento è di circa 0.00009.
* **Validation Token Accuracy (punto arancione)**: L'accuratezza dei token durante la validazione è di circa 0.00016.
* **Test Token Accuracy (punto verde)**: L'accuratezza dei token durante il test è di circa 0.00004.

L'accuratezza dei token è estremamente bassa in tutti i dataset, il che potrebbe suggerire che il modello ha difficoltà a prevedere correttamente i token individuali. Questo potrebbe essere un indicatore di un problema con l'architettura del modello, i parametri di addestramento, o la qualità dei dati.

**Conclusioni Generali**

* **Loss**: La perdita è ragionevolmente bassa e simile tra i dataset, il che è un buon segno.
* **Perplexity**: La perplessità è costante tra i dataset, indicando una buona capacità del modello di generalizzare.
* **Token Accuracy**: L'accuratezza dei token è molto bassa, suggerendo che il modello potrebbe non essere ben ottimizzato per il compito specifico.

**Raccomandazioni**

1. **Rivedere l'architettura del modello**: Potrebbe essere necessario modificare o utilizzare un'architettura diversa.
2. **Regolare i parametri di addestramento**: Provare diversi tassi di apprendimento, dropout, e altre iperparametri.
3. **Analizzare i dati di addestramento**: Assicurarsi che i dati siano di alta qualità e ben preprocessati.
4. **Aumentare il numero di epoche**: Potrebbe essere utile aumentare il numero di epoche per vedere se l'accuratezza migliora.

Il grafico indica che il modello ha una buona perdita e perplessità, ma l'accuratezza dei token molto bassa suggerisce che ci sono problemi che devono essere risolti per migliorare la precisione delle previsioni.