"""

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: [0.8, 0.1, 0.1]  ### a gestione di più sottoinsiemi può richiedere risorse aggiuntive.

  global\_max\_sequence\_length: 1210

trainer:

  type: finetune

  epochs: 2

  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 schermata, linea, testo, Diagramma

Descrizione generata automaticamente

Analisi Approfondita del Grafico

**1. Loss per Epoch**

Nel grafico a sinistra, analizziamo la metrica di perdita (Loss) per l'addestramento (Train), la validazione (Validation) e il test (Test) nel corso di due epoche.

Train Loss: La perdita di addestramento inizia a circa 1.35 e scende leggermente nella seconda epoca, indicando che il modello sta apprendendo e riducendo la sua perdita su questo set di dati.

Validation Loss: La perdita di validazione è inizialmente più bassa rispetto alla perdita di addestramento, partendo da circa 1.10. Questo suggerisce che il modello generalizza abbastanza bene sui dati di validazione fin dall'inizio. Tuttavia, non vediamo un miglioramento significativo nella seconda epoca.

Test Loss: La perdita di test è la più bassa tra le tre, partendo da circa 0.95. Questo potrebbe indicare che il set di test è più facile rispetto ai set di addestramento e validazione, oppure che il modello ha visto dati simili durante l'addestramento.

**2. Perplexity per Epoch**

Il grafico centrale mostra la perplessità (Perplexity) per l'addestramento, la validazione e il test attraverso le epoche.

Train Perplexity: La perplessità di addestramento è inizialmente molto alta (circa 31990) e diminuisce leggermente nella seconda epoca. Questo è coerente con la riduzione della perdita di addestramento, indicando che il modello sta migliorando la sua capacità di predire la sequenza dei token.

Validation Perplexity: Anche la perplessità di validazione inizia alta (circa 31990) ma è leggermente inferiore rispetto a quella di addestramento. Non mostra un grande miglioramento nella seconda epoca, suggerendo che il modello non sta migliorando significativamente la sua generalizzazione.

Test Perplexity: La perplessità di test è la più bassa e mostra un significativo miglioramento nella seconda epoca, scendendo a circa 31987. Questo potrebbe indicare che il modello è particolarmente efficace nel predire i token per il set di test, o che il set di test ha sequenze più semplici.

**3. Token Accuracy per Epoch**

Nel grafico a destra, vediamo l'accuratezza dei token (Token Accuracy) per l'addestramento, la validazione e il test.

Train Token Accuracy: L'accuratezza dei token durante l'addestramento è estremamente bassa, iniziando a circa 0.00006 e migliorando leggermente nella seconda epoca. Questo riflette la difficoltà del modello nel predire accuratamente ogni token nella sequenza di addestramento.

Validation Token Accuracy: L'accuratezza dei token per la validazione è inizialmente la più alta tra le tre, a circa 0.00013, ma non migliora significativamente. Questo potrebbe indicare che il modello ha una certa capacità di generalizzare ma non sta migliorando con ulteriori addestramenti.

Test Token Accuracy: L'accuratezza dei token per il test è intermedia tra l'addestramento e la validazione e mostra un leggero miglioramento nella seconda epoca, arrivando a circa 0.00010. Questo suggerisce una discreta capacità del modello nel predire i token correttamente per il set di test.

Considerazioni e Raccomandazioni

**Overfitting e Underfitting:**

Non sembrano esserci segni chiari di overfitting, dato che la perdita e la perplessità di validazione non aumentano significativamente rispetto a quelle di addestramento.

Il miglioramento marginale nelle metriche di validazione suggerisce che il modello potrebbe aver bisogno di più epoche di addestramento per migliorare ulteriormente.

**Parametri di Addestramento:**

Potrebbe essere utile sperimentare con diversi tassi di apprendimento, batch size, o aggiungere tecniche di regolarizzazione per migliorare la performance del modello.

**Architettura del Modello:**

Considerare la possibilità di utilizzare un'architettura di modello diversa o più complessa, se le risorse lo permettono, per migliorare la capacità del modello di catturare la complessità dei dati.

**Ulteriori Epoche:**

Aumentare il numero di epoche potrebbe portare a un miglioramento continuo, specialmente se le metriche di addestramento mostrano una tendenza a migliorare.

Questo grafico fornisce una chiara visualizzazione delle performance del modello attraverso le epoche, aiutando a identificare aree di miglioramento e a guidare ulteriori esperimenti.

Immagine che contiene testo, schermata, linea, Diagramma

Descrizione generata automaticamente

**Training and Validation Loss per Epoch**

Il grafico mostra l'andamento della perdita (Loss) sia per l'addestramento (Training) che per la validazione (Validation) attraverso due epoche.

**Osservazioni Principali:**

**Andamento della Perdita di Addestramento (Training Loss)**

Epoca 1: La perdita di addestramento inizia a circa 1.35.

Epoca 2: Scende costantemente fino a circa 1.10.

Questo andamento suggerisce che il modello sta apprendendo dai dati di addestramento e sta migliorando la sua capacità di minimizzare l'errore.

**Andamento della Perdita di Validazione (Validation Loss)**

Epoca 1: La perdita di validazione inizia a circa 1.15, più bassa rispetto alla perdita di addestramento.

Epoca 2: Scende costantemente fino a circa 1.00.

Questo indica che il modello sta migliorando anche nella sua capacità di generalizzare sui dati non visti (di validazione), mantenendo una buona prestazione su di essi.

**Considerazioni e Raccomandazioni:**

Miglioramento Continuo:

Il grafico mostra un miglioramento costante sia nella perdita di addestramento che in quella di validazione. Questo è un buon segno che il modello sta imparando efficacemente senza mostrare segni di overfitting (dove la perdita di validazione aumenterebbe mentre la perdita di addestramento continua a diminuire).

Equilibrio tra Addestramento e Validazione:

La perdita di validazione è sempre inferiore alla perdita di addestramento, suggerendo che il modello non sta semplicemente memorizzando i dati di addestramento ma sta anche generalizzando bene.

Questo grafico suggerisce che il modello sta apprendendo bene sia dai dati di addestramento che da quelli di validazione, con una riduzione costante della perdita in entrambe le categorie. Continuare l'addestramento e monitorare le prestazioni potrebbe portare a ulteriori miglioramenti.

Immagine che contiene testo, linea, schermata, Diagramma

Descrizione generata automaticamente

Next Token Perplexity per Epoch

Il grafico mostra l'andamento della perplessità del token successivo (Next Token Perplexity) attraverso due epoche.

Osservazioni Principali:

**Andamento della Perplessità del Token Successivo**

Epoca 1: La perplessità inizia a circa 17.000.

Epoca 2: Scende costantemente fino a circa 16.200.

Questo andamento indica che il modello sta migliorando la sua capacità di prevedere il prossimo token nel contesto dato, con una riduzione significativa della perplessità.

**Interpretazione della Perplessità**

La perplessità è una misura dell'incertezza del modello nel prevedere il prossimo token. Un valore più basso indica che il modello è più sicuro nelle sue previsioni.

La riduzione da 17.000 a 16.200 suggerisce un miglioramento nella fiducia del modello sulle sue previsioni di token successivi.

**Considerazioni e Raccomandazioni:**

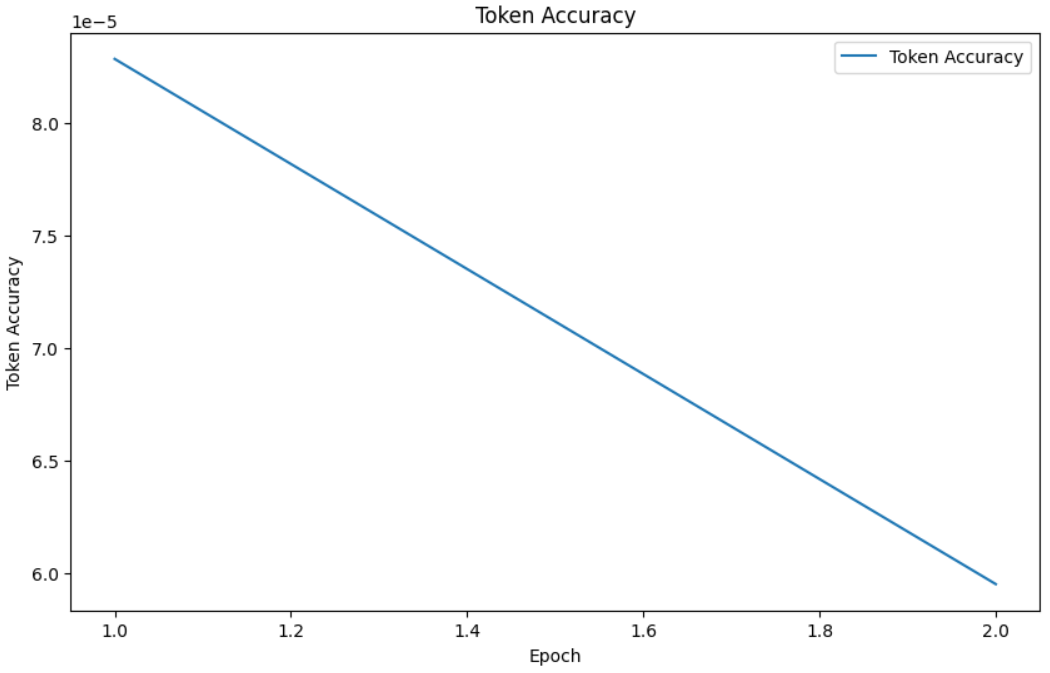
Miglioramento Continuo:

Il grafico mostra un miglioramento costante nella perplessità del token successivo. Questo è un buon segno che il modello sta diventando più efficace nel prevedere il prossimo token nel contesto dato.

Monitoraggio Continuo:

Continuare a monitorare la perplessità durante l'addestramento può fornire indicazioni utili su come il modello sta apprendendo e dove potrebbero essere necessari aggiustamenti.

Questo grafico suggerisce che il modello sta migliorando nella sua capacità di prevedere i token successivi, con una riduzione costante della perplessità nel corso delle epoche. Continuare l'addestramento e monitorare attentamente le prestazioni potrebbe portare a ulteriori miglioramenti.



Token Accuracy per Epoch

Il grafico mostra l'andamento dell'accuratezza dei token durante due epoche di addestramento.

Osservazioni Principali:

**Andamento dell'Accuratezza dei Token**

Epoca 1: L'accuratezza dei token inizia a circa 8.5e-5.

Epoca 2: Scende costantemente fino a circa 6e-5.

Questo andamento indica che l'accuratezza del modello nel predire i token corretti è in diminuzione nel corso delle epoche.

**Interpretazione dell'Accuratezza dei Token**

L'accuratezza dei token misura la proporzione di token correttamente previsti rispetto al totale dei token previsti.

Una diminuzione dell'accuratezza dei token potrebbe suggerire che il modello sta avendo difficoltà a migliorare la sua capacità di predizione, o che potrebbe esserci un problema con l'addestramento.

Considerazioni e Raccomandazioni:

Miglioramento Continuo:

A differenza della perplessità, l'accuratezza dei token mostra un trend negativo. Questo potrebbe indicare che il modello sta lottando per migliorare o che è in una fase di decadenza delle prestazioni.

Questo grafico suggerisce che l'accuratezza dei token del modello sta diminuendo durante l'addestramento, il che indica che potrebbero essere necessari aggiustamenti nei parametri di addestramento o nell'architettura del modello per migliorare le prestazioni.

**Conclusione:**

Il modello mostra segni di miglioramento in termini di perdita e perplessità, ma l'accuratezza dei token in calo è una preoccupazione significativa. Pertanto, sarebbe consigliabile rivedere i parametri di addestramento, considerare modifiche all'architettura del modello, migliorare la qualità dei dati e continuare a monitorare le prestazioni per ottenere un miglioramento complessivo delle prestazioni del modello.