Relazione progetto NLP

Raffaele Calì m:1000005479

Strategie di Classificazione del testo

Indice

Introduzione

Il seguente progetto approfondisce il campo della classificazione del testo, definito come "il processo di categorizzazione del testo in gruppi organizzati, i classificatori di testo analizzano il testo e assegnano un insieme di etichette o categorie predefinite basate sul suo contenuto". L'obiettivo della ricerca è la classificazione dei testi in cinque categorie: business, sport, politica, tecnologia e scienza, e 'altro'. A tale scopo, sono stati combinati tre dataset, con caratteristiche differenti ma aventi le stesse etichette. Per effettuare la classificazione , sono state utilizzate tecniche di deep learning , tecniche classiche di machine learning ed una combinazione di entrambe. Inoltre, è stata realizzata una demo live che permette di testare la capacità dei modelli sviluppati di classificare nuovi testi in tempo reale.

Dataset

Il dataset finale è il risultato della combinazione di tre dataset , BBC News , AG News, 20Newgroup. Per garantire coerenza attraverso le diverse fonti, è stato necessario un processo di mappatura per categorizzare i testi in cinque categorie predefinite. I testi dei tre dataset sono di tipologie differenti :

- **BBC News** offre articoli giornalistici, con un linguaggio formale che copre un'ampia gamma di argomenti, da eventi globali a notizie scientifiche.
- **AG News** include sintesi di notizie, caratterizzate da brevi e incisivi riassunti di eventi attuali e storie di interesse.
- **20Newsgroup** presenta una collezione di messaggi di forum su vari argomenti, riflettendo uno stile di scrittura più informale e una vasta gamma di opinioni e argomentazioni.

Il dataset contiene ~130 000 record ,con una predominanza di testi appartenenti alla categoria tecnologia e scienza. Tuttavia, la distribuzione dei testi per categoria è pressoché bilanciata, garantendo una varietà rappresentativa. Ogni record è strutturato in colonne che indicano il testo (*Text*), la categoria (*Category*) e un indice di provenienza (*df_index*), quest'ultimo identifica il dataset originale (1 per BBC News, 2 per 20Newsgroup, 3 per AG News).
Il dataset è stato poi suddiviso in due parti distinte: `dataset_k_neigh.csv` e `dataset_Longformer.csv`. Quest'ultimo è stato ulteriormente elaborato per generare un dataset secondario, `generated_pairs.csv`, composto da coppie di testi (*Text1*, *Text2*), una variabile booleana (*Same_Category*) che indica l'appartenenza alla stessa categoria e due colonne aggiuntive che specificano la categoria di ciascun testo. Questo dataset secondario include 15.000 record, per un totale di 30.000 testi, di cui circa ~25.000 unici, ed i restanti ripetuti per fornire esempi sia positivi che negativi della stesso testo. Inoltre, esso mantiene una distribuzione equilibrata del 50% tra record "true" e "false", e cercando di preservare una distribuzione uniforme di esempi positivi e negativi per ogni categoria.

Tecniche utilizzate e Risultati

La strategia iniziale, o baseline, impiega l'estrazione dei token CLS da Longformer, senza il fine-tuning, seguita dalla classificazione tramite k-nearest neighbors (KNN). Nella seconda strategia, è stato utilizzato il modello Latent Dirichlet Allocation (LDA) per analizzare il testo e fornire una distribuzione di probabilità sui topic pre configurati, con questi vettori di probabilità passati successivamente al KNN per la classificazione.
Il terzo approccio ha visto il fine-tuning di Longformer su un task di classificazione binaria, utilizzando il dataset `generated_pairs.csv`. Questo modello è stato addestrato per distinguere se due testi appartengono alla stessa categoria (true) o a categorie diverse (false), attraverso l'uso della distanza del coseno mappata con una sigmoide, per cinque epoche. Post addestramento, sono stati estratti gli embeddings CLS per il KNN, sperimentando sia con l'uso esclusivo di questi embeddings sia integrandoli con la distribuzione di probabilità di LDA.
Infine, l'ultima strategia impiegata consiste nel fine-tuning di Longformer per un task di classificazione multiclasse, integrando un layer di classificazione direttamente nel modello, che produce l'etichetta di classe.

Il **Lonformer** è un language model basato su Transformer, progettato per elaborare testi di lunghezza superiore rispetto ai modelli come BERT. Una delle principali innovazioni di Longformer è l'introduzione di meccanismi di attenzione globale e locale che permettono di gestire sequenze di testo estremamente lunghe (fino a 4096 token) in modo efficiente. Questa caratteristica rende Longformer particolarmente adatto per applicazioni che richiedono l'analisi di documenti lunghi, senza la necessità di troncare il testo.

Il **token CLS** è un token speciale inserito all'inizio della sequenza di input nei modelli Transformer. In contesti di classificazione, l'embedding del token CLS dopo il passaggio attraverso il modello è spesso utilizzato come rappresentazione aggregata dell'intero testo di input. Questo embedding può essere poi impiegato per la classificazione diretta o come input per ulteriori modelli o tecniche di classificazione, come il k-nearest neighbors (KNN).

Il **k-nearest neighbors (KNN)** è un algoritmo di classificazione basato su distanza che non richiede un modello esplicito per la classificazione. Utilizzando gli embeddings CLS estratti come caratteristiche, il KNN classifica ogni testo confrontandolo con i k esempi più vicini nel training set, basandosi su una metrica di distanza. La categoria predominante tra i vicini viene assegnata al testo in esame.

Baseline: Estrazione Token CLS da Longformer e Classificazione con KNN

La strategia baseline impiega due componenti : l'estrazione dei token CLS dal modello Longformer, e l'uso dell'algoritmo k-nearest neighbors (KNN) per la classificazione.

Procedura:

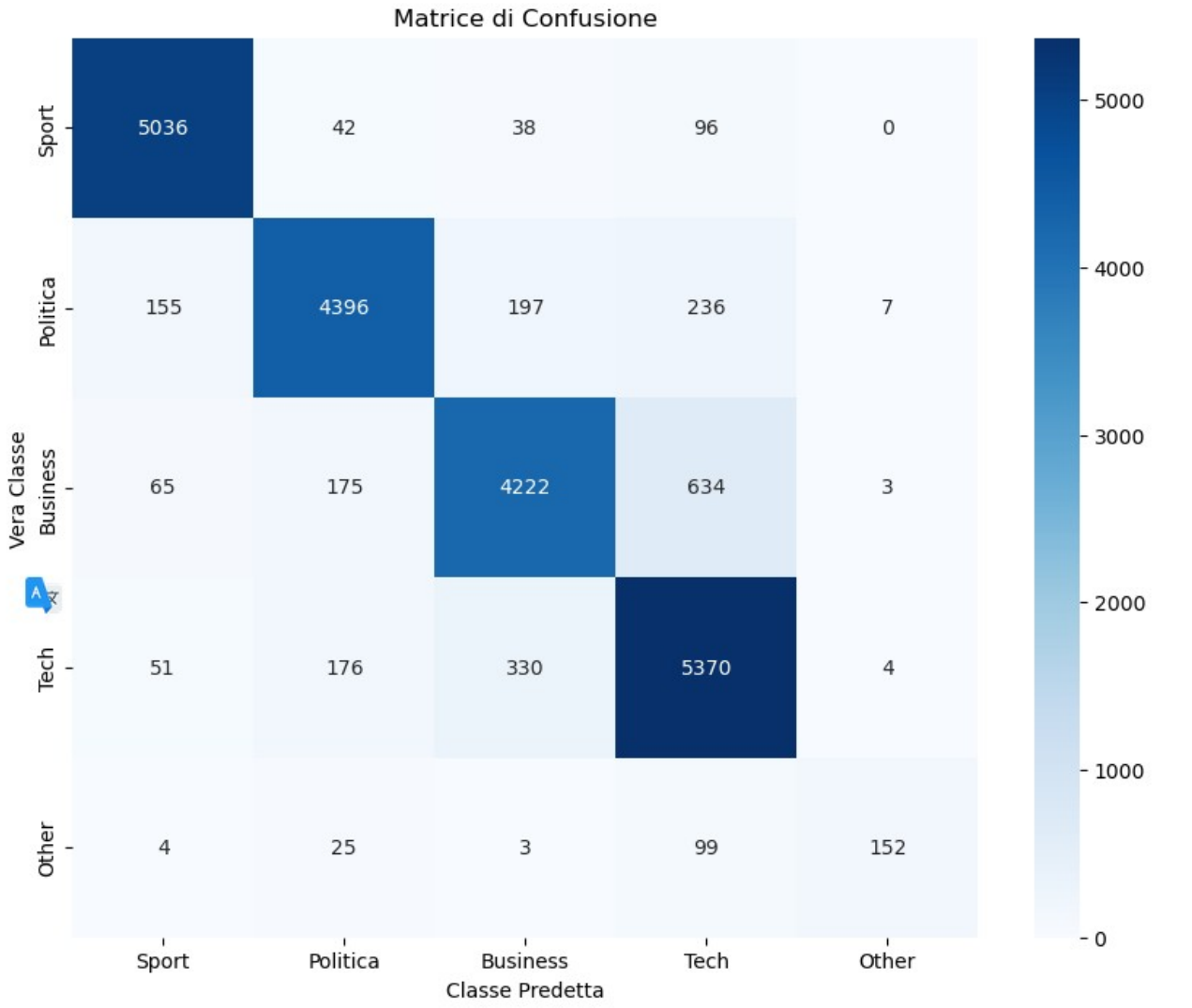
- **Preparazione dei Dati:** I testi vengono prima preprocessati (tokenizzazione, padding) per adattarli ai requisiti di input del Longformer.
- **Estrazione degli Embeddings:** Per ogni testo nel dataset, il Longformer processa il testo e ritorna il token CLS (768 dimensioni).

- **Classificazione con KNN:** Gli embeddings CLS servono come input per l'algoritmo KNN, che li utilizza per classificare i testi nelle categorie predefinite basandosi sulla similitudine con i campioni di training.

Questo approccio consente di utilizzare tanti dati per il KNN, perche non vi è la necessita di ulteriori suddivisioni. Infatti, l'unica suddivisione è tra i set di dati di addestramento e test per il KNN stesso.

Risultati

Accuracy	Precision	Recall	F1
`0.891243`	`0.898594`	`0.823507`	`0.850655`



Seconda Strategia: LDA e KNN

LDA (Latent_Dirichlet_allocation) è una rete bayesiana che consente di scoprire i topic latenti presenti in una collezione di documenti. Operando su principi di distribuzione di probabilità, LDA

analizza i pattern di co-occorrenza delle parole all'interno dei documenti per identificare temi ricorrenti, senza richiedere annotazioni o etichette predefinite.

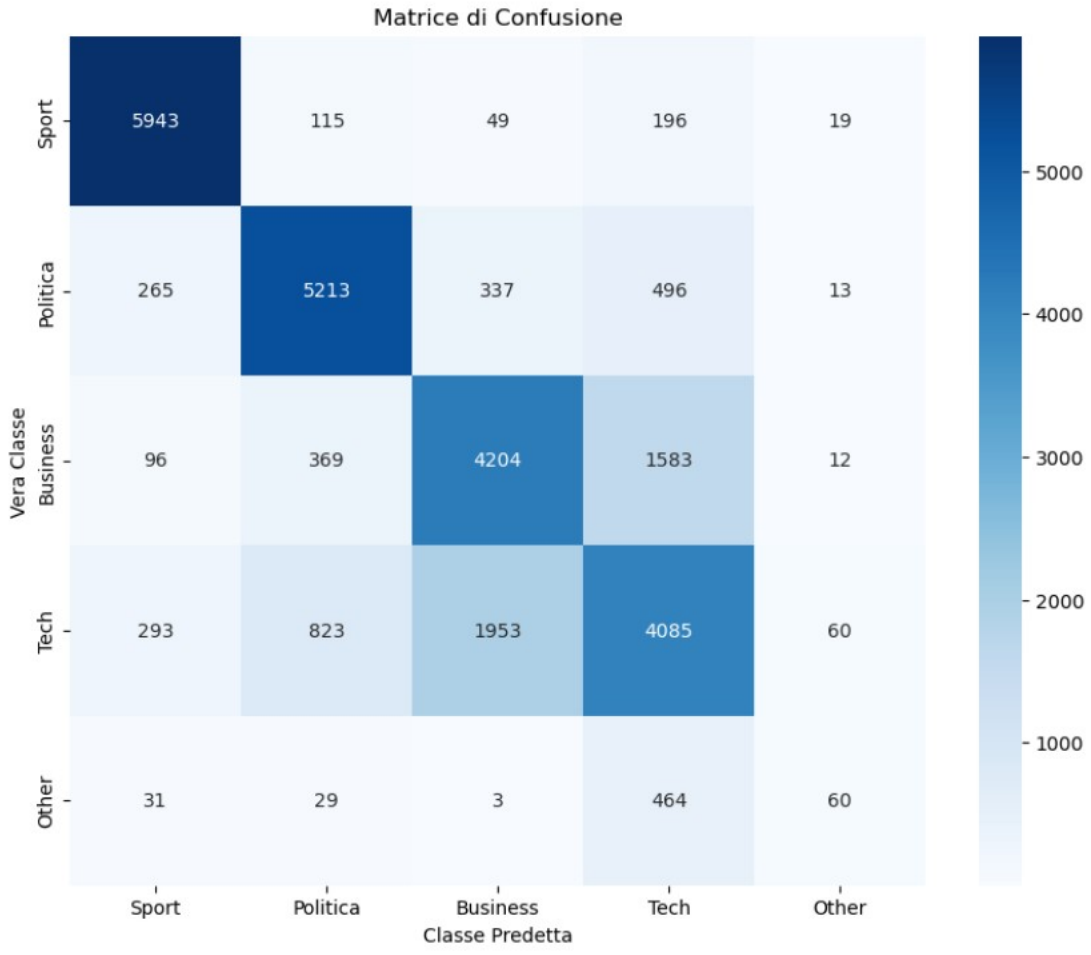
Procedura :

1. **Configurazione dei Topic:** LDA è stato configurato per identificare una distribuzione di probabilità su 16 topic predefiniti (numero topic = 16).
2. **Analisi dei Testi:** Ogni documento del dataset viene processato attraverso LDA, che assegna al testo una distribuzione di probabilità relativa alla pertinenza di ciascun topic. Questo risultato fornisce una rappresentazione vettoriale basata sui topic per ogni testo.
3. **Classificazione con KNN:** Utilizzando le distribuzioni di probabilità dei topic come caratteristiche, procediamo con la classificazione dei documenti impiegando nuovamente l'algoritmo KNN.

Questo approccio raggiunge buoni risultati in tempi brevi, tutorial nel file *lda_only.ipybn*

Risultati

Accuracy	Precision	Recall	F1
0.730223	0.659870	0.620796	0.625178



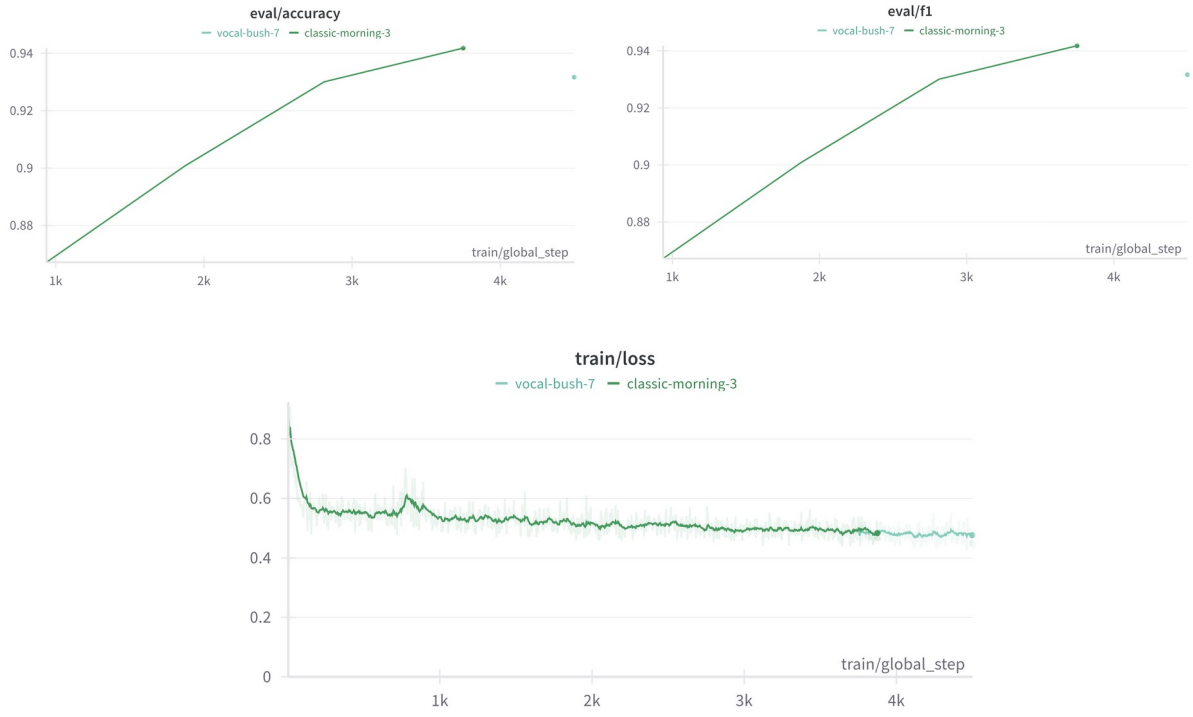
Terza strategia : finetuning longformer + knn

Questa strategia adottata introduce un livello aggiuntivo : il fine-tuning di Longformer su un task di classificazione binaria, utilizzando come dataset `generated_pairs.csv`. Questo approccio mira a sfruttare la capacità del modello di apprendere direttamente dalle relazioni tra coppie di testi, determinando se appartengono alla stessa categoria o meno.

Procedura di Fine-Tuning:

- **Addestramento:** Il modello è stato addestrato su `generated_pairs.csv` per cinque epoche.

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
0	0.547500	0.524543	0.867200	0.912270	0.867200
2	0.511300	0.487825	0.930067	0.959081	0.930067
4	0.512700	0.473986	0.941800	0.967134	0.941800
5	0.446900	0.477389	0.931667	0.965980	0.931667



- **Estrazione degli Embeddings CLS:** Post addestramento, sono estratti gli embeddings CLS dal modello fine-tunato.

Classificazione con KNN: Con gli embeddings CLS ottenuti dal modello fine-tunato, procediamo alla classificazione utilizzando l'algoritmo KNN. Sono date esplorate due varianti :

- **Uso Esclusivo degli Embeddings CLS:** In questa variante, i testi sono rappresentati esclusivamente dagli embeddings CLS estratti.
- **Combinazione con la Distribuzione di Probabilità di LDA:** Per arricchire ulteriormente il set di caratteristiche, gli embeddings CLS sono integrati con le distribuzioni di probabilità dei topic ottenute dal modello LDA (768 + 16 dimensioni).

Risultati KNN:

	Accuracy	Precision	Recall	f1
CLS	0.891243	0.898594	0.823507	0.850655
CLS+LDA	0.882041	0.879741	0.820107	0.843028
FT CLS+LDA	0.843272	0.815307	0.830376	0.822154
FT CLS	0.803820	0.761076	0.794695	0.775339

Quarta strategia : finetuning longformer con classificazione multiclasse

La quarta strategia consiste nel fine-tuning del modello Longformer per un compito di classificazione multiclasse diretta (predire direttamente la categoria di appartenenza). Diversamente dalle strategie precedenti, che si basavano sull'utilizzo di embeddings estratti dal modello e successiva classificazione tramite algoritmi esterni come il KNN, o sulla combinazione con altre tecniche di analisi del testo come LDA, questa strategia integra il processo di classificazione direttamente nel modello Longformer.

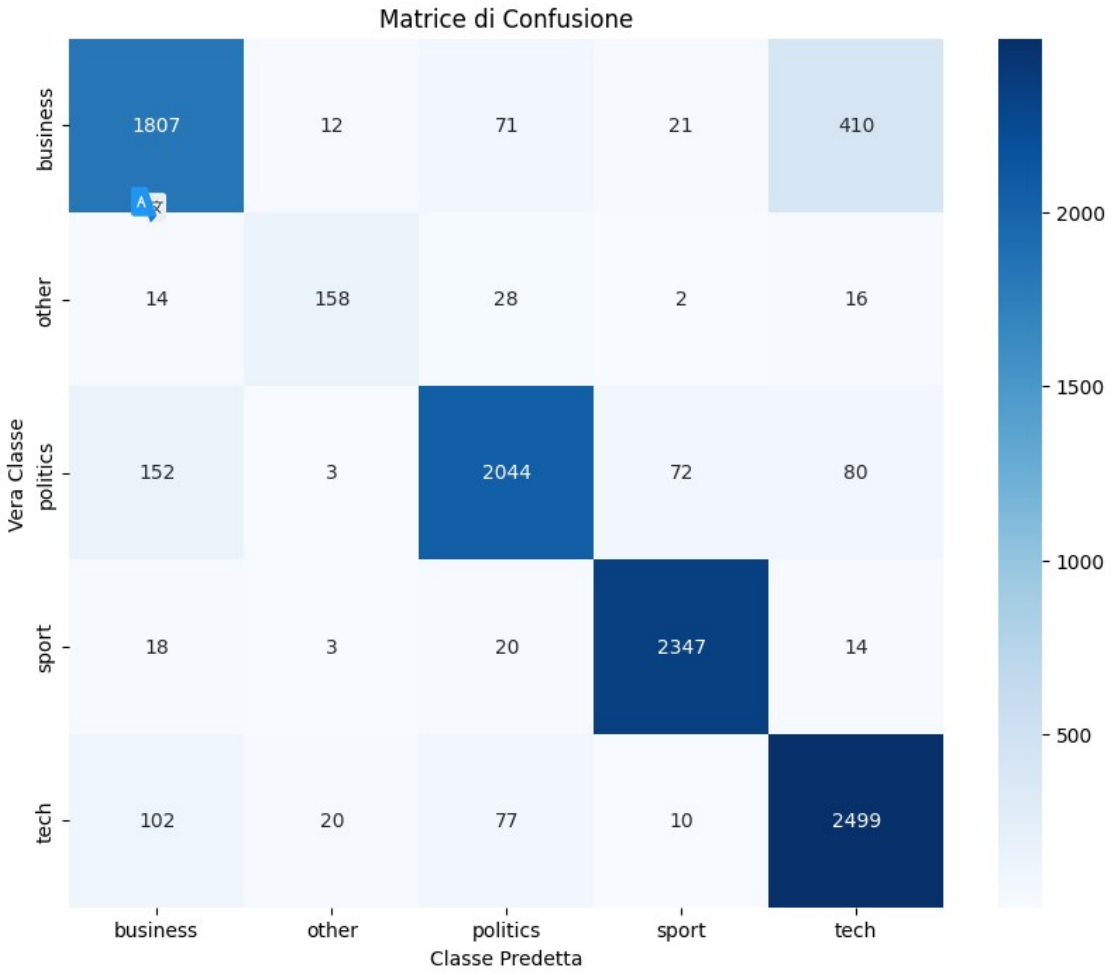
Procedura di Fine-Tuning e Classificazione Multiclasse:

1. **Preparazione dei Dati**: Le categorie sono state trasformate in vettori one hot.
2. **Addestramento**: Il modello è stato addestrato per 5 epoche con un campione (20 000 record) del dataset `dataset_Longformer.csv` ,

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
1	0.170800	0.183865	0.874375	0.921319	0.873889
2	0.183000	0.189403	0.869541	0.918056	0.868333
3	0.163900	0.167383	0.876047	0.921042	0.871667
4	0.145500	0.181163	0.879666	0.923958	0.877222
5	0.092700	0.168075	0.883380	0.922917	0.871111

Risultati:

	Accuracy	**Precision**	**Recall**	**f1**
Multiclass_Longf	0.8855	0.8733	0.8545	0.8624



Conclusioni e sviluppi futuri

	Accuracy	Precision	Recall	f1
CLS	0.891243	0.898594	0.823507	0.850655
CLS+LDA	0.882041	0.879741	0.820107	0.843028
FT CLS+LDA	0.843272	0.815307	0.830376	0.822154
FT CLS	0.803820	0.761076	0.794695	0.775339
Multiclass_Longf	0.8855	0.8733	0.8545	0.8624

Attraverso l'esplorazione di vari approcci, è emerso che il fine-tuning di Longformer per un compito di classificazione multiclasse ha dimostrato le prestazioni migliori, con pochi dati e in tempi rapidi.

Per quanto riguarda gli sviluppi futuri, un'area di particolare interesse è l'ulteriore esplorazione di metodi per gestire l'attenzione globale. Quest'ultima è un concetto chiave nei modelli trasformers come il Longformer, la quale permette di catturare le dipendenze a lungo raggio nel testo.

Demo

Per provare la demo eseguire il file demo.py, inserire il testo nella textarea e cliccare il tasto classifica testo.