- Temă predicție -

Grigore Iulia-Andreea

1. **Modelul folosit:**

**CamemBERT** este bazat pe arhitectura transformer, antrenat pe un volum mare de text în limba franceză, ceea ce îl face potrivit pentru task-ul de față. De asemenea, modelul captează în profunzime semantica limbii, fiind extrem de eficient pentru clasificarea textului.

1. **Metoda abordată:**

În vederea etichetării textului din *test.csv*, am folosit modelul camembert/camembert-large preantrenat, disponibil pe [huggingface.co](https://huggingface.co/almanach/camembert-large), la care am făcut fine-tuning pe setul de date *train.csv*. Astfel, modelul se va specializa pe setul de date pus la dispoziție și va fi capabil să eticheteze textul drept true, fake sau biased cu o acuratețe notabilă.

1. **Procesul de Fine Tunning:**

În bază analizei asupra setului de date *train.csv*, am constatat că este dezechilibrat, clasa predominanta fiind cea a textelor cu eticheta true. Ca atare, am făcut **oversampling** pe setul de date, replicând intrări din clasele minoritare, pentru a preveni obținerea unui model cu performanțe bune numai pe clasa majoritară.

Pentru validare, am împărțit setul de date *train.csv* în doua:

* 90% este setul de training pentru model
* 10% este setul de validare, care permite validarea performantelor modelului, în urma procesului de fine tuning.

Pentru a preveni erorile în ceea ce privește clasele minoritare, folosesc **class weights**, incorporata în compute\_loss pentru a penaliza greșelile în predicție.

**Optuna** este folosită pentru a găsi combinația cea mai bună de parametri pentru o **acuratețe** cât mai mare. Aceasta caută parametri optimi pentru:

* learning\_rate
* num\_train\_epochs
* per\_device\_train\_batch\_size

pe care ii pasează la final\_training\_args,trainerul ce conține parametri optimi pentru o acuratețe cât mai bună.

Această metodă poate fi aplicată pentru mai mulți parametri, însă acuratețea, în cazul acesta, este puțin mai scăzută față de scenariul optim.

**Pentru a preveni overfitting-ul,** am adăugat o metodă de **Early Stopping**, care, în baza valorii early\_stopping\_patience, 4, în cazul de față, oprește procesul de antrenare, dacă acuratețea nu crește în 4 epoci.

1. **Rezultate:**

În urma procesului de fine tuning, modelul este validat folosind setul de validare creat în faza incipienta a procesului de antrenare, care constituie 10% din train.csv. **Scorul cel mai bun** obținut de model este de **aproximativ 84%**.

Matricea de confuzie:

A screenshot of a computer screen

AI-generated content may be incorrect.

1. **Surse:**

Codul folosit pentru antrenarea modelului CamemBERT și etichetarea datelor din *train.csv* poate fi regăsit aici:

<https://github.com/Iulia-Andreea20/CamemBert/tree/main>