Clasificator companii

Challenge-ul care m-a intrigat pe mine cel mai tare a fost acesta de a atribui etichete de asigurare pentru un set de business-uri. Luand in considere ca nu aveam inainte un volum prea mare de cunostinte in aceste tipuri de probleme / solutii, am ales ce am considerat eu ca mi se potriveste si mi se pare mai interesant.

Primul pas pe care l-am realizat a fost sa analizez datele furnizate, in primul rand ca sa inteleg mai bine ceea ce se cere, dar in mod special pentru a face o statistica si a vedea in ce directie as putea sa ma indrept cu rezolvarea. Am observat diferite pattern-uri, dar si probleme care ar putea sa apara in cadrul rezolvarii. Cateva exemple ar fi:

1. Doar 7 valori distincte la sector
2. 2 sectoare dominante: Manufacturing (aprox 4000) si Services (aprox 3500) => ~80% din datele pe care le aveam
3. Niche era singurul camp care nu avea date lipsa
4. Cand aveam camp lipsa la sector, aveam lipsa si la categorie
5. Din toate cele 220 de label-uri de atribuit aveam: 146 care se terminau in Services si 28 care se terminau in Manufacturing => ~80% din label urile totale (se poate observa asemanarea cu punctul 2.)
6. etc

Dupa ce am evaluat datele si m-am familiarizat cu problema, primul lucru pe care l-am facut in rezolvare a fost de a face un dictionar cu label-urile respective si cuvintele extrase din label. Motivul a fost pentru a cauta acele cuvinte in business-uri. Am ales sa evit cautarea lor in descriere pentru ca am observat ca face matching gresit pe acest caz. Acea cautare era pentru fiecare cuvant in tot textul si de cele mai multe ori nu aveau legatura cu label-ul original (putea sa gaseasca primul cuvant la inceput si urmatorul la sfarsit in contexte complet diferite).

Am folosit un sentence transformer pentru a calcula similaritatea dintre label-uri si datele businessului. In continuare nu am folosit descrierea pentru ca mi-am dorit un set de date foarte precis la inceput si nu voiam sa risc sa am prea multe atribuiri gresite. Pe aceste label-uri atribuite am facut antrenarea si mi-era teama sa nu se antreneze gresit daca aveam prea multe match-uri incorecte.

Am observat ca in urma celor aplicate, exista label-uri din lista care nu au fost atribuite, asa ca m-am gandit sa fac un set de date manual (am folosit chat-gpt, dar le-am verificat si le am introdus eu intr-un excel). Dupa crearea lor, le-am multiplicat ca sa aiba un numar mai mare de aparitii. Am separat datele cu label-uri si cele fara pentru a le pregati pentru model.

Am inceput antrenarea pe aceste date si dupa ce am prezis pe restul, label-urile nu erau deloc atribuite corespunzator, asa ca am schimbat modelul dintr-un bert-base-uncased intr-un longformer, crezand ca textul meu era prea mare pentru primul care nu poate sa aiba atat inputul, dar si ca modelul nu era suficient de bun.

Dupa ce am incercat partea de antrenare si prezicere din nou, ceva tot nu mergea bine. Stiam ca am unele label-uri care apar de foarte multe ori si altele extrem de rar in setul de train. Astfel, le-am echilibrat in functie de label-ul care apare cel mai des si le-am adus pe toate la acel

numar. Pentru ca antrenarea a durat enorm pentru ele si echilibrarea lor dupa aceasta regula a dus la mult prea multe date la fel, chiar daca era „shuffleuite”, modelul a inceput sa faca overfitting. Astfel, am incercat sa rezolv acesta problema reducandu-le semnificativ, dar pastrandu-le echilibrate ca numar de aparitii.

Chiar si dupa aceasta imbunatatire (cred eu), modelul meu prezicea si mai prost si overfitting a fost problema cu care m-am confruntat cel mai tare. In functie de ce am tot testat eu, modelul ori invata prea bine niste label-uri si doar pe acelea le atribuia, ori era plafonat pe toate si nu mai conta. In plus, threshold-ul era extrem de mic si nici cand am incercat top k, nu erau raspunsuri macar aproapiate de adevar.

In urma cautarii problemei, am ajuns la concluzia ca 220 de label-uri sunt prea mule pentru atat de putine date pentru un encoder. Asa ca am modificat iar abordarea si am folosit un decoder de data aceasta (qwen2.5-3b-instruct) si am cautat prompt ul necesar pentru acesta. Am preferat sa aiba in input toate label urile din care sa aleaga pentru a evita cat mai mult atribuirea unora generate de el.

Aceasta modificare din encoder in decoder a fost de o importanta maxima. Am ajuns sa am o parte din date prezise corespunzator (am extras label-urile din prompt-ul specific). In continuare, am facut perfect match intre label-urile prezise si cele din lista mea pentru a evita diverse probleme de scriere. Pentru label-urile prezise care nu se regaseau in lista am cautat cel mai apropiat label din lista mea de cel prezis (am folosit un prag destul de mare=0.75 pentru a fi sigura ca majoritatea vor fi corecte).

Pentru a verifica corectitudinea, am adaugat un scor de similaritate intre label-urile atribuite prin predict si intreg textul businessului. Din verificarile mele, majoritar cele peste 0.4 scor sunt corespunzatore. In continuare, au ramas business-uri cu label-uri neatribuite pentru ca nu am reusit sa le corespund pe cele prezise cu cele date (in schimb din cele prezise lipsesc doar 2%, deoarece inputul ori era inconsistent ori business ul avea mult prea multe servicii).

OBSERVATII:

- In train set am avut doar 225 din 1213 date care sa aiba multi-label, astfel nici modelul meu nu a putut prezice prea multe multi-label.
- Initial am vrut sa folosesc Galore in loc de Lora pentru ca este mai bun, doar ca fiind nou implementat nu era compatibil cu primul meu model folosit (bert-base-uncased), asa ca am ramas la varianta de Lora pe tot parcursul rezolvarii
- La Lora am folosit un alpha la jumatatea rank-ului, chiar daca din research-ul meu am vazut ca se foloseste fix invers, dar dupa mai multe cautari am aflat ca aceasta este o solutie pentru overfitting (cum am zis mai sus, problema mea esentiala pentru foarte mult timp)
- Folosirea unui decoder chiar daca initial din cautarile mele parea contraintuitiv, s-a demonstrat a fi rezolvarea unor probleme majore
- O solutie la care ma gandisem initial pentru anumite minoritati de label-uri sau sectoare dominante era sa impart in 2 problema si sa privesc distributia lor astfel: partea dominanta facea parte dintr-o clasa de date si toate celelalte faceau parte din alta, astfel toate minoritatile adunate ajungeau sa fie cat de cat la egalitate cu partea dominanta (in cazul de fata la sectoare erau 2 care alcatuiau un procent prea mare pentru ideea mea

de abordare, in schimb cred ca s-ar fi putut implementa pe label-urile care erau de tip Services si dupa toate celelalte ramase)

- Am ales sa folosesc un GPU inchiriat pentru ca pe localul pe care lucram a durat aprox 8 ore prima antrenare de 1000 si ceva de date
- O alta metode de verificare a solutiei ar fi sa ai un set de date al caror label-uri le cunoastem si sa verificam acuratetea modelului (setul de date poate fi generat de un LLM cum am facut cu cele care nu apareau deloc)
- Inca o alta idee pentru verificare ar fi sa facem rephrasing pe business-uri si sa vedem cat de corecte sunt acum label-urile fata de evaluarea initiala
- Sunt generate foarte multe fisiere csv pentru a putea analiza datele la fiecare pas
- Cele mai dificile parti au fost: ce metode aplic dupa un research extrem de lung pe machine learning, cand incepi sa inveti un volum mare de informatii (nu aveam prea multe cunostinte de ML) cel mai greu este sa faci discernamant pe ce vei folosi si ce nu; iar a doua: argumentele de training si predict (sa inteleg fiecare ce face si ce trebuie pus si ce nu si ce valori, mai ales in momentul in care dureaza ceva timp sa vezi rezultatele)