Számítógépes szemantika

Semeval 2021 Task 5: Toxic span detection

Feladat:

Gyorsan és egyszerűen: Mondatokból kiszűrni a toxic szavak indexeit.

Például:

input: „Youre an idiot”.

output: [9,10,11,12,13]

Nehézségek a projekttel kapcsolatban:

* Kettő data mappa volt a versenyfeladat github repojához feltöltve. Mindkettőben voltak tanító adatok, egyikben 46.000, másikban csak 7900. Az utóbbit csak később vettem észre, mert az egy almappában volt. Végül kutakodás után megtudtam, hogy a 7900-ast használták a versenyhez, így a BERT modelleket én is azon tanítottam, de nem azzal kezdtem el dolgozni elsőre (kódban benne hagytam).
* Számítási kapacitás: elég lassú volt, bárhogy is futtattam. Így a tesztelés sokkal nehézkesebb volt, sok időt elvett.
* Feladat elsőre egyszerűnek hangzik, de mégsem az, sok idő kellett a pontos megértéséhez, és végrehajtásához.

Volt megadva teszt csv fájl, melyben vannak hibák, ráadásul sok. Teljesen átkozódó mondatoknál is van olyan, hogy a teszt.csv alapján nincs benne toxic tartalom. Ez miatt erős gyanúm van, hogy sokkal jobb eredményeket érnék el egy helyes, validált adatbázison.

Például:

1.

"Chris Birch is a mean, self-centered, contrary ass. He repeatedly interrupted Robin Brena during his presentation and always, always sucks up to Big Oil. He could care less about his constituents and the rest of Alaska. He needs to get gone. What has he done for you, South Anchorage? Sarcastic is the perfect adjective for Chris Birch!"

Toxic spans: []

2.

"Hillary Clinton is a liar, thief, foul-mouthed, evil, dishonorable, sexual predator enabler and you choose to support her. What does that indicate about you?"

Toxic spans:[]

A képen szöveg, szoftver, képernyőkép, Webhely látható

Automatikusan generált leírás

Kiértékelés:

F1 score-t kellett használni.

(For each system, we computed the

F1 score per post, between the predicted

and the ground truth character offsets.

Then, we returned the macro-averaged (over test posts) score.)

A legjobb eredmények létrehozói ezeket használták:

RoBERTa modell, CRF layerrel összekötve.

A baselineokat a feladat kiírói készítették:

BENCHMARK I: Roberta+CRF – 69.89%

BENCHMARK II: lexicon-based system – 64.98%

BENCHMARK III: random baseline -12.22%

**Alapvető célnak azt tűztem ki, hogy legalább a BENCHMARK** **II-t legyőzzem.**

Első megközelítés: ToxicBERT+TXT

**Score: 1. 50% | 2. 45.9% | 3. 54.69%**

Egy ideig tartó ötletelés és egy sikertelen bert modell építés után keresgéltem a neten, és találtam egy már létező BERT modellt, amely detektálja, hogy mely mondatok tartalmaznak toxic szavakat, vagy jobban mondva melyik sugároz ’mérgező’ tartalmat. Ezt végig futtattam az összes adatra, és prediktált mellé egy értéket. A ToxicBERT outputja 0-tól 1-ig jelzi, hogy milyen toxic egy mondat. Ha toxicnak vélte, az összes szóra megnézte hogy benne van-e egy txt fájlban. Ezt az értéket lehet állítani, hogy mikortól gondolja toxicnak, ezt használtam ki. A legjobb eredmények fentebb láthatóak, a pontos küszöbérték már nincs meg, de valahol 0.2-0.4 között volt. A tesztelő dataframe-ben ezt eltároltam, hogy melyik mondat toxic, és melyik nem.

**1.**

Az első megközelítésben egy neten talált txt fájlt használtam fel, amelyben toxic szavak voltak felsorolva (1930 szó). Az abban levő szavakat eltároltam egy változóban, és a tokenizált mondatokkal összevetettem. Ha talált olyan szót, ami benne van a listában, akkor annak az indexeit belerakta a tesztelő dataframe ’pindex’ nevű oszlopába, ami egy lista.

**F1 score: 50%**

**2.**

A másodiknál szintén egy neten talált txt fájlt használtam, de ebben voltak olyan szavak is amik nem feltétlen mérgezőek, csak szimplán nem egyetértőek, nem pozitív töltetűek (pl: disagreed). Erre rosszabb eredmény született.

**F1 score: 45.9%**

**3.**

A harmadik próbálkozásnál az 1. pontban taglalt txt fájlt használtam, csak kibővítettem. Neten találtam pár kisebb toxic szó txt-ket, azokat bemásoltam egy külön fájlba az, majd arra az 1. pont txt fájlját is rámásoltam. Ezután python kóddal kivettem az olyan szavakat, amik többször szerepelnek. Ez már jobb eredményt adott, mert csak toxic szavak voltak, és több mint az elsőben.

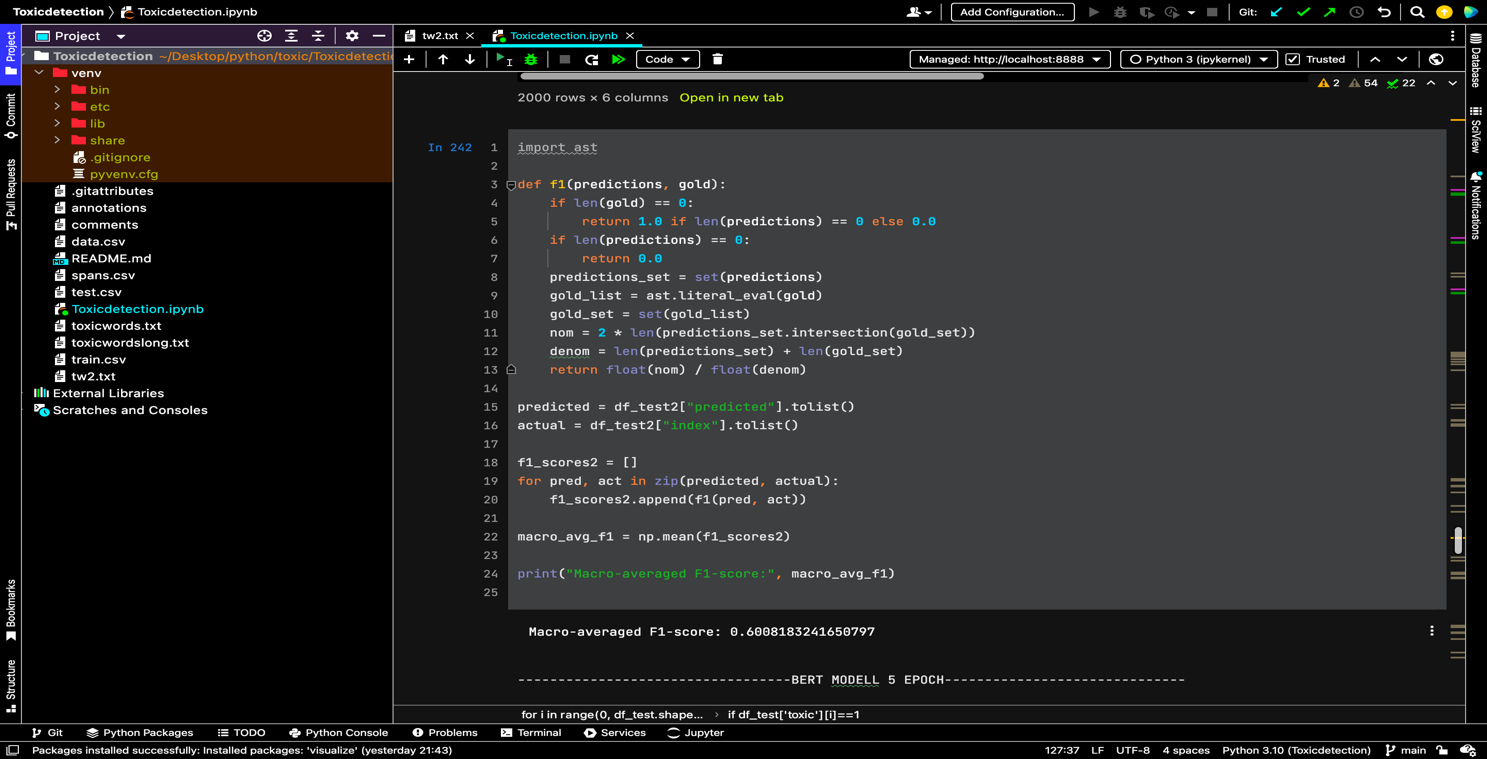
**F1 score: 54.69%**

BERT MODELL 2 EPOCH

**Score: 60%**

Ebben a megközelítésben egy BERTClassifier modellt építettem az alábbi megoldásokkal:

1. A tokeneket kiegészítettem padding tokenekkel, hogy azonos hosszúságúak legyenek.
2. Az inputként kapott tokeneket átalakítja token azonosítókká a BERT szótárában, majd alkalmazza rájuk a paddinget
3. A tanítóadat tokenjeit és maszkjait létrehozzuk.
4. Tanítóadatból kinyerjük a cimkéket, majd azt átalakítjuk tensor formátumúra, és alkalmazzuk a paddinget.
5. BertClassifier osztály: Egy BERT-alapú osztályozó modellt definiál.
6. A modellunk tehát: Létrehoz egy példányt a BertClassifier osztályból és betölti a BERT modellt a ’bert-base-uncased’ előre betanított verziójából.
7. Az optimalizáló algoritmus az Adam, a veszteségfüggvény pedig a bináris keresztentrópia.
8. A kód minden egyes batch feldolgozása során a veszteséget és a predikciókat is rögzíti.

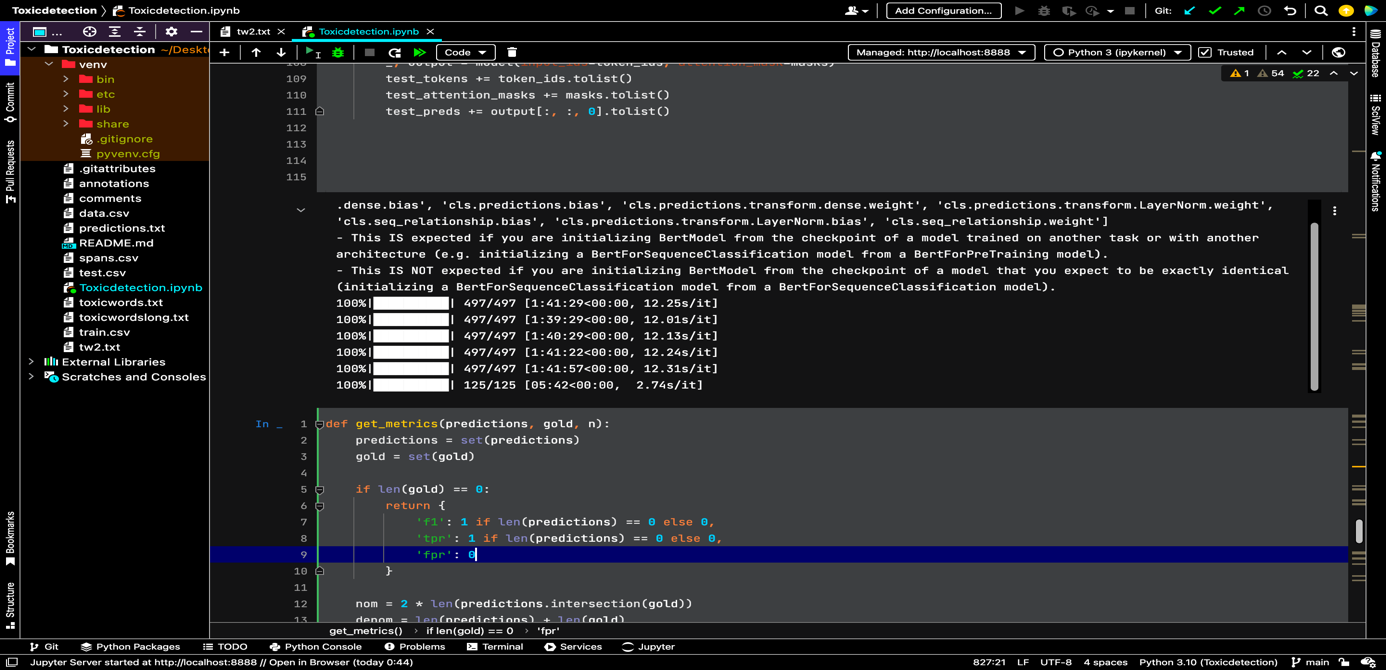
A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

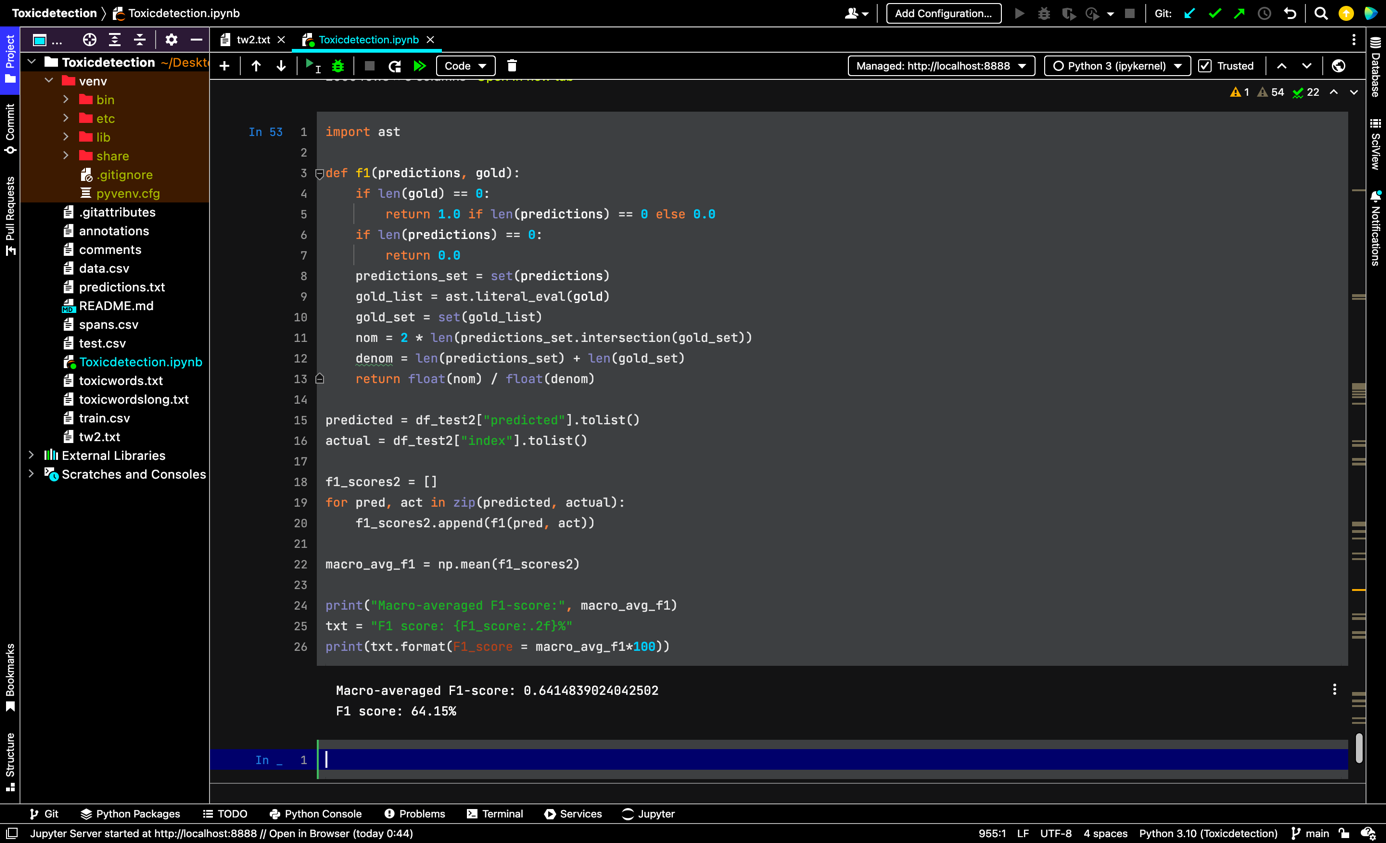
Automatikusan generált leírásAz első próbálkozás 2 epoch-al futott le. Nagyjából 4,5 óráig futott a tanítása, ami mutatja hogy elég időigényes. Ezzel elért eredmény: 60%

BERT MODELL 5 EPOCH

**Score: 64.15%**

Lényegi megközelítése teljesen ugyanaz, mint az előzőnek. Annyi eltérés van, hogy itt 5 epoch-nyi tanulás futott le. Ez elég sok idő lett, 5\* 1 óra 40 perc, nagyjából 8-8,5 óra volt. A várakozásnak megfelelően ez hozta a legjobb eredményt, bár az én személyem jobb eredményt ettől.





Nem használtam LARGE modellt, mert azt írta az egyik csapat, hogy nem alkalmas rá a kevés adat miatt. Több adattal elkerülhetőbb lenne a túltanulás. Erre alkalmas lehet a 46.000 mondattal rendelkező .csv fájl, viszont ehhez már nincs megfelelő számítási kapacitása a gépemnek. (Már a 7900 mondatos tanulási session után is fel akart robbani, és 10 órát vett igénybe).