Descrierea soluției de multi-output regression pentru prezicerea poziției geografice a unui tweet

George Moldovan

19 decembrie 2020

1 Prezentare Generală

Soluția propusă este una de învățare supervizată, pe baza tweeturilor si a pozițiilor geografice prezente in setul de antrenare. Abordarea prezentată este una în 2 pași, mai precis :

- Extragerea de hand-crafted features din setul de date și construcția unui feature map pentru fiecare tweet.
- Antrenarea unui model de regresie multi-output (deoarece rezultatul constă in 2 variabile, coordonatele geografice ale utilizatorului).

2 Preprocesarea textului : Diferitele abordari

Din punct de vedere al preprocesării, fiecare tweet a suferit următoarele operații :

- Modificarea textului pentru a conține doar litere mici. (Deoarece nu există o legătura directă intre capitalizarea unui anume cuvant si poziția geografică a utilizatorului)
- Eliminarea cuvintelor uzuale din limba germană (stop-words), deoarece acestea sunt folosite indiferent de locație deci nu aduc nici o valoare dacă sunt luate în calcul și scad calitatea antrenării.
- Tokenizarea tweet-ului (împărțirea acestuia în cuvinte separate).

Prima abordarea a constat in folosirea unui **CountVectorizer** din librăria **sklearn**, care, cu parametrii potriviți, oferă capacitățile de prepocesare descrise mai sus.

```
self.count_vectorizer = CountVectorizer(encoding='utf-8',lowercase=True,
stop_words=stopwords.words('german'),ngram_range=(1, 1),max_df=0.5,
min_df=0, binary = True, max_features = 20000)
```

Modul in care este definit primul modul de preprocesare, face ca doar cele mai importante 20000 de cuvinte din setul de date se fie considerate importante (default-ul este ca numar-ul de features sa fie egal cu numarul de cuvinte distincte din setul de antrenare).

Deasemenea, am ales sa iau in considerare doar cuvinte individuale, nu si perechi, de aceea ngram-range are limita maxima 1.Max-df si min-df se refera la frecvența minimă si maximă a unui cuvânt pentru ca acesta să fie pastrat în setul de cuvinte relevante. Maxim-ul e setat la 0.5, deoarece in acest mod se extinde lista de cuvinte ignorate din cauza frecvenței cu care apar, deoarece nu toate cuvintele irelevante din limba germană sunt prezente în lista de stop-words pasată ca parametru. Binary este setat pe true deoarece doresc construcția unui BOW, nu și sa număr aparițiile fiecărui cuvânt.

În urma antrenării modelului pe setul de date procesat astfel, rezultatul a fost unul nesatisfăcator, de doar **1.0133827971956562**, mai slab chiar decât baseline-ul. Astfel, am decis aplicarea unui TfidfTransformer peste datele obținute. Am decis să folosesc o astfel de abordare deoarece frecvența unui cuvânt într-un tweet oferă mult mai mult context despre nucleul mesajului, putând obține rezultate mult mai bune.

```
TfidfTransformer( norm='l2',use_idf=True, smooth_idf=True, sublinear_tf=True)
```

Pe lângă calcularea frecvenței cuvântului în tweet, ponderea acestora este calibrată de frecvența acestora în cadrul setului de antrenare. Un cuvânt cu frecvență mică în setul de date este mult mai relevant pentru locație tweet-ului decât un cuvânt care apare de multe ori in tweet, dar deasemenea este folosit si in foarte multe alte tweet-uri.

Pentru a obține aceasta pondere variabilă parametrului use-idf este True, iar pentru a evita împărțirea la 0, smooth-idf este setat pe True pentru a presupune ca orice cuvânt apare cel puțin o dată în setul de antrenare. Sublinear-tf este setat pe True pentru a logaritma frecvența termenilor, evitând un set de antrenare cu discrepanțe foarte mari intre valori.

Trecerea de la valori absolute la valorea calculată folosind frecvența termenilor în tweet ponderată de inversul frecvenței termenului în dataset duce la îmbunătățiri semnificative pe setul de validare, obținând un MSE de **0.5930440608413687**.

3 Alegerea modelului și a parametrilor folosiți

Generarea locației pe baza textului din tweet are la baza o regresie multi-label ce foloseste un regressor cu vectori suport peste care este aplicat un regressor chain, pentru a obține un output multidimensional. Prin aplicarea unei inlănțuiri pentru a obține output-ul bi-dimensional practic garantez ca coordonata y este prezisă in funcție de feature-vectorul asignat tweet-ului plus rezultatele generate de primul regressor (adica coordonata x prezisa). Faptul ca prezicera lui x este folosită în prezicerea lui

y face ca cele 2 variabile să fie dependente una de cealalta îmbunătațind rezultatele față de o abordare în care se antrenează 2 modele cu output uni-dimensional în mod separat.

```
LinearSvrModel = LinearSVR(C=0.5,random_state=1242,loss='squared_epsilon_insensitive
LinearSvrWrapper = RegressorChain(LinearSvrModel)
LinearSvrWrapper.fit(loader.train_processed_tweets,loader.train_labels)
```

Modelul de baza este un LinearSVR deoarece din experiența personala mașinile cu vectori suport oferă cele mai bune rezultate atunci cand datele au fost procesate corespunzător si prezintă un grad mare de separare. Atât valoarea C (parametrul de regularizare) cât și funcția loss au fost selectate prin încercari manuale deoarece aplicarea unui grid search pe un model ce face parte dintr-un RegressorChain nu a imbunatatit rezultatele.

4 Rezultate pe setul de date de validare

Pe setul de date de validare s-au obținut următoarele rezultate :

- MAE : 0.5863074767024781
- MSE : 0.5930440608413687