Racism Analysis

NATURAL LANGUAGE PROCESSING

1. ALEGEREA DATASETULUI

Pentru alegerea datasetului am optat pentru datasetul tweets_hate_speech_detection oferit de Hugging Face. Acest dataset contine 31,962 de tweet-uri colectate folosind API-ul de la Tweeter si clasificate in tweet-uri rasiste si non-rasiste.

Fiecare intrare din dataset este formata din 3 campuri:

1. id-ul tweet-ului
2. label-ul tweet-ului (0 = neutru, 1 = rasist)
3. textul tweet-ului

2. ARTICOLE ASOCIATE

Pentru a înțelege mai bine contextul studiului discursului rasist, am analizat un set de articole științifice care au la bază aceeași tematică.

De aceea, am ales paper-ul „Whose opinions matter? Perspective-aware models to identify opinions of hate speech victims in abusive language detection" [1] scris de Sohail Akhtar, Valerio Basile și Viviana Patti. Acesta prezintă un studiu bazat pe faptul că unele caracteristici ale diferitelor comunități influențează opiniile acestora, iar, de aceea, în analiza tipurilor de hate speech, trebuie luate în calcul mai multe criterii. Se folosesc de un data set adnotat de persoane care fac parte din comunitățile cele mai afectate de hate speech, arătând astfel cum predicțiile făcute cu aceste informații sunt mai bune decât cele produse prin alte metode.

De asemenea, un alt articol care a contribuit la formarea unei imagini despre tema aleasă a fost „A survey of Race, Racism, and Anti-Racism in NLP" [2] scris de Anjalie Field, Su Lin Blodgett, Zeerak Waseem și Yulia Tsvetkov. Acesta prezintă ideea că, studiul limbajelor naturale de procesare este influențat de caracteristica rasei, dar acest fapt este ignorat în multe lucrări de specialitate. De asemenea, articolul subliniază ideea că research-ul pentru NLP ar trebui să fie mai inclusiv în practicile sale.

Un al treilea articol ales de noi a fost „Racism Detection by Analyzing Differential Opinions Through Sentiment Analysis of Tweets Using Stacked Ensemble GCR-NN Model" [3] scris de Ernesto Lee, Furqan Rustam, Patrick Bernard Washington, Fatima El Barakaz, Wajdi Aljedaani și Imran Ashraf. Acest paper își propune prezentarea unui model de deep learning care detectează tweet-uri rasiste prin intermediul sentiment analysis. Modelul ales combina gated recurrent unit (GRU), convolutional neural networks (CNN) și recurrent neural networks RNN, cu o detecție de 97%.

3. PREPROCESAREA DATELOR

De cele mai multe ori conținutul regăsit pe internet, din partea utilizatorilor, nu este într-o formă optimă pentru a fi procesat și a se aplica metode de învățare. Devine importantă normalizarea textului prin aplicarea unei serii de pași de preprocesare. Am aplicat un set de pași de preprocesare pentru a-l face potrivit pentru algoritmii de învățare și pentru a reduce dimensiunea setului de caracteristici.

Din punctul de vedere al clasificării unui text, cele mai importante aspecte ale preprocesării necesare setului de date ales implică:

- **Eliminarea userului:** Fiecare utilizator Twitter are un nume de utilizator unic. Orice lucru îndreptat către acel utilizator poate fi indicat scriind numele de utilizator precedat de „@”, care nu furnizează nicio informație utilă, fiind un nume propriu.
- **Eliminarea link-urilor:** Utilizatorii partajează adesea hyperlinkuri în tweet-urile lor. Twitter le scurtează (folosind serviciul său intern de scurtare a adreselor URL), cum ar fi http://t.co/FCWXoUd8.
- **Eliminarea punctuațiilor, numerelor și a caracterelor speciale:** Utilizatorii folosesc punctuațiile, numerele și caractere speciale într-un mod abuziv, intenționat sau accidental, iar acestea nu reflectă nicio stare sau sentiment și elimiarea acestora este necesară pentru a avea un text cât mai curat.
- **Eliminarea cuvintelor scurte:** Termenii precum „hmm”, „oh” sunt foarte puțin folositori și nu au niciun sens sau rol specific în text.
- **Modificarea literelor mari în litere mici:** Ajută la menținerea fluxului de consistență și extragerii de text.
- **Eliminarea caracterelor repetitive:** În limbajul de zi cu zi, oamenii de multe ori nu sunt strict gramaticali. Vor scrie lucruri precum „I looooooove it", pentru a sublinia cuvântul dragoste. Cu toate acestea, computerele nu știu că „looooooove” este o variație a „iubirii” decât dacă li se spune.
- **Modificarea literelor mari în litere mici:** Cuvinte precum „Carte" și „carte" înseamnă același lucru, dar atunci când nu sunt convertite în litere mici, cele două sunt reprezentate ca două cuvinte diferite în modelul spațiului vectorial
- **Eliminarea simbolului # din hashtag-uri:** Simbolul nu oferă niciun sens util textului în analiza sentimentelor, fiind folosit pe rețelele de socializare, ca o indicație că o bucată de conținut se referă la un anumit subiect sau aparține unei categorii.
- **Eliminarea spațiilor multiple:** De cele mai multe ori, textele conțin spații suplimentare sau în timpul efectuării tehnicilor de preprocesare de mai sus, rămâne mai mult de un spațiu intre cuvinte.
- **Stemming:** Există multe variante de cuvinte care nu aduc informații noi și creează redundanță, aducând în cele din urmă ambiguitate atunci când antrenăm modele de învățare automată pentru predicții.

4. EXTRAGEREA FEATURE-URILOR

Pentru analizarea datelor preprocesate, este nevoie ca acestea să fie convertite în feature-uri. Depinzând de metoda în care urmează să fie folosite, feature-urile de text pot fi construite folosind diferite tehnologii. Noi am ales să transformăm datele în reprezentări numerice vectoriale.

Pentru a putea efectua această operație, avem nevoie să eliminăm toate semnele de punctuație și apoi să calculăm frecventa cuvintelor. Cu ajutorul acestora putem să creăm un vocabular cu anumite caracteristici specifice, prin intermediul funcției createVocab.

Vom folosi funcția createVectorize pentru a atribui fiecărui caracter din vocabularul creat un index. Luăm în calcul posibilitatea ca unele carcatere să nu fie cunoscute și, de aceea, lor le vom atribui indexul 0. În cadrul funcției createVectorize, după ce efectuăm vectorizarea, vom apela funcțiile vectorizeSenteces și pad.

Funcția vectorizeSentences este utilizată pentru a transforma propozițiile într-o reprezentare vectorială. Aceasta ia fiecare tweet și îl transformă în reprezentarea lui sub formă de indici specifici caracterelor conținute. Apoi, fiecărui indice îi facem reprezentarea one-hot.

Din cauza faptului că tweet-urile folosite sunt de dimensiuni diferite, este necesară aducerea acestora la aceeași lungime maximă. Pentru aceasta folosim funcția pad, care primește un set de tweet-uri și o lungime maximă. Aceasta are ca scop fie scurtarea datelor care depășesc lungimea

maximă, fie adăgarea valorii de padding (aleasă de noi ca fiind 1) la datele care sunt mai scurte dectât lungimea maximă.

La finalul tuturor acestor pași putem extrage feature-urile care urmează să fie folosite pentru antrenarea modelului.

5. CLASIFICARE

A. Model folosit

Pentru clasificarea datelor, am folosit un model CNN cu un dropout (p=0.4), 2 layere convolutionale de sizex128 si 128x128, un average pooling 1D si un layer liniar de 128xout pentru output.

Am folosit functia Adam pentru optimizarea modelului cu learning_rate = 0.001 si CrossEntropyLoss.

B. Dataset

Datasetul a fost impartit in 3 dataframe-uri (train 80%, test 10% si validation 10%), fiecare dataframe fiind stocat in batch-uri de cate 64, randomizate.

C. Evaluare modelului

Datorita setului de date neechilibrat (doar 8 % din tweet-uri sunt rasiste), am evaluat acuratetea ca media dintre acuratetile tweet-urilor rasiste si non-rasiste.

D. Antrenarea modelului

Pentru a antrena modelul, am rulat de 40 ori cate 100 de epoci, salvand mereu modelul cel mai bine evaluat. Astfel, dupa finalizarea unui set de 100 de epoci, incarcam cel mai bun model si il rulam din nou pe cate 100 de epoci, pentru a se evita overfitting-ul.

E. Rezultate

In final, in urma a 3 rulari a cate 100 de epoci (salvand mereu cel mai bun model), am putut obtine o acuratete maxima de 83.5% (True labels: 84.3%, False labels: 82.8%)

REFERENCES

1. S. Akhtar, V. Basile, and V. Patti, "Whose opinions matter? perspective-aware models to identify opinions of hate speech victims in abusive language detection," arXiv preprint arXiv:2106.15896 (2021).
2. A. Field, S. L. Blodgett, Z. Waseem, and Y. Tsvetkov, "A survey of race, racism, and anti-racism in nlp," arXiv preprint arXiv:2106.11410 (2021).
3. E. Lee, F. Rustam, P. B. Washington, F. E. Barakaz, W. Aljedaani, and I. Ashraf, "Racism detection by analyzing differential opinions through sentiment analysis of tweets using stacked ensemble gcr-nn model," IEEE Access **10**, 9717–9728 (2022).