Offensive Language Classification

Boboc Ștefan, Dobre Adriana-Lia,
Guțu Stefania-Alexandra, Pirlogea Luciana-Elena

Abstract

Această proiect prezintă un model de Natural Language Processing proiectat pentru a clasifica tweet-urile în două categorii: limbaj ofensiv și limbaj normal. Modelul utilizează o combinație de tehnici de învățare automată și caracteristici lingvistice pentru a identifica cu precizie instanțele de limbaj ofensiv din tweet-uri. Un set de date cuprinzător, format din tweet-uri etichetate, a fost preluat pentru antrenarea și evaluarea performanței modelului. Diferite etape de preprocesare, inclusiv tokenizare, lematizare și eliminarea cuvintelor de legătură au fost aplicate pentru a îmbunătăți reprezentarea caracteristicilor. Modelul a fost antrenat folosind RNN(Recurrent neural network). Mai mult, o analiză extinsă a performanței modelului, inclusiv precizia, loss-ul și scorul F1 evidențiază capacitatea sa de a distinge eficient limbajul ofensiv de limbajul normal.

1 Introducere

În ultimii ani, evoluția platformelor de socializare online a dus la răspândirea largă a limbajului ofensiv, generând impacte sociale negative și provocări în menținerea unui mediu online sigur. Din această cauză, se încearcă detectarea cât mai rapidă a unui astfel de comportament pentru a combate cât mai eficient acest fenomen pe social media. Prin identificarea și clasificarea precisă a unor astfel de instanțe, ne propunem să contribuim la eforturile în curs de desfășurare pentru a promova un mediu online sigur și incluziv. Acest proiect se concentrează pe dezvoltarea unui model de NLP (Procesare a Limbajului Natural) special conceput pentru a clasifica tweet-urile în două categorii: limbaj ofensiv si limbaj normal.

Restul acestui articol este organizat în următoarele secțiuni: Secțiunea 2 oferă o prezentare generală a cercetărilor relevante în detectarea limbajului ofensiv și abordărilor bazate pe NLP. Secțiunea 3 descrie alegerea setului de date. Secțiunea 4 prezintă implementarea modelului. Secțiunea 5 discută rezultatele și analiza performanței modelului propus. În final, Secțiunea 6 încheie articolul cu un rezumat al concluziilor.

2 Articole asociate

Înainte de a intra în profunzime cu detaliile implementării modelului, am ales câteva articole pentru a înțelege mai bine contextul identificării limbajului ofensiv. Astfel, "Whose Opinios Matter? Perspective-Aware Models To Identify Opinios Of Hate Speech Victims In Abusive Language Detection" [1] prezintă impactul pe care îl are background-ul social asupra opiniilor unui individ despre comunitățile vulnerabile cum ar fi imigranții, LGBT, femeile, etc. Acești factori trebuie luați în calcul în antrenarea unui model ce oferă predicții mult mai bune decât modelele clasice.

Articolul "Detecting Hate Speech in Social Media" [2] urmărește găsirea unor metode de distingere între limbajul vulgar și limbajul ofensiv adresat unor categorii de indivizi. Folosind caracteristicile n-grams, word n-grams, word skip-grams și prin impărțirea în trei clase, modelul a obținut o acuratețe de 78%.

Articolul "Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter" [3] se concetrează asupra limbajului ofensiv adresat imigranților și femeilor. Clasificarea este împărțită în două module: primul se bazează pe detectarea limbajului ofensiv, iar al doilea se concentrează pe descoperirea caracteristicilor mai detaliate pentru a distinge daca limbajul ofensiv este adresat către o persoană sau generalizat către un grup de persoane.

3 Setul de date

În urma efectuării unei analize a mai multor seturi de date, am optat pentru setul de date Hate Speech and Offensive Language Dataset [4] descărcat de pe Kaggle. Acesta conține 24783 de tweet-uri care sunt clasificate în hate-speech, limbaj ofensiv și limbaj normal. Pentru o antrenare mai bună a modelului, am decis să echilibrăm setul de date împărțindu-l în două clase: 0 - limbaj ofensiv și 1 - limbaj normal. După extragerea datelor din fișierul CSV, am obținut un set de date care are următoarea structură:

- clasa (0 - limbaj ofensiv, 1 - limbaj normal)
- textul tweet-ului

4 Implementarea modelului

4.1 Preprocesarea datelor

Preprocesarea datelor este un pas esențial în analiza datelor și în construirea modelelor de învățare automată. Acesta implică aplicarea unor transformări și tehnici de curățare și pregătire a datelor brute pentru a le face mai ușor de înțeles și de utilizat de către algoritmi de învățare automată.

Pașii pe care i-am urmat pentru sanitizarea datelor sunt:

- eliminarea numelui de utilizator: Fiind un nume ales de fiecare utilizator, acesta nu are nicio relevanță în clasificarea tweetului. Numele de utilizator este unic și începe întotdeauna cu "@".
- eliminarea link-urilor: În multe tweet-uri, acestea sunt adăugate de utilizator pentru a referenția o altă informație care nu este utilă pentru detectarea limbajului ofensiv. (exemplu: http://t.co/7KPWAdLF0R).
- eliminarea hashtag-urilor: Acestea sunt folosite pentru a marca cuvintele cheie relevante unui trend și care nu oferă niciun detaliu important. (exemplu: #EarlyChristmas).
- eliminarea ampersandului: Multe din tweet-urile din dataset conțin secvențe de cifre/litere ce încep cu & (exemplu: &#128524).
- eliminarea caracterelor speciale și a cifrelor: Acestea nu au caracter concludent în clasificarea prezentă.
- transformarea caracterelor mari în caractere mici: Vectorizarea cuvântului "Casă" și a cuvântului "casă" va fi diferită dacă nu aplicăm peste acest pas.
- înlocuirea secvențelor repetitive cu aceeași literă cu două repetiții a acesteia: Repetiția inutilă a literelor poate afecta procesul de preprocesare, deoarece sunt cuvinte inexistente (exemplu: "funnnnnnn").
- eliminarea spațiilor multiple: Duplicarea spațiilor multiple poate afecta împărțirea tweet-urilor în cuvinte.
- lematizare: Acest proces implică reducerea cuvintelor la forma lor de dicționar, fapt care ne ajută la detectarea cuvintelor ofensive (exemplu: lematizarea cuvântului "Caring" ar întoarce "Care").
- eliminarea stopword-urilor și eliminarea cuvintelor cu mai puțin de două caractere: Termenii de acest fel nu contribuie la îmbunătățirea extragerii de caracteristici (exemple: "you", "ok", "is", "the").

4.2 Extragerea caracteristicilor

Dupa preprocesarea datelor, acestea trebuie sa fie transformate în caracteristici, iar noi am ales să lucrăm în modelul spațiului vectorial, utilizând Word2Vec. Folosind acestă tehnică de procesare, nu am luat în considerare cuvintele cu o frecvență mai mică decât trei. Am ales să utilizăm skip-gram, fiind o tehnică care găsește cele mai relevante cuvinte pentru un cuvânt dat și ne ajută la prezicerea contextului. De asemenea, am folosit un window de șapte cuvinte.

Pentru fiecare cuvânt, am făcut media vectorului generat de Word2Vec și am adaugat-o în lista corespunzătoare tweet-ului din care face parte, ignorând valorile NaN. Am adăgat padding pentru fiecare tweet, astfel că fiecare listă are aceeași lungime maximă. Apoi, am convertit datele în tensori PyTorch și am încărcat datele folosind Data Loader.

4.3 Model

Am împărțit setul de date în 3 dataframe-uri, și anume: 70% train, 15% test 15% validare, fiecare dataframe fiind stocat în batch-uri de câte 50, randomizate.

Am folosit modelul RNN cu următorii parametrii:

- input_size (lungimea maximă a unui tweet) care este numărul de caracteristici din input
- hidden_size = 32 care este numărul de hidden units din stratul ascuns
- output_size = 2 care este numărul de clase pe care le prezice modelul

Ca loss function, am folosit CrossEntropyLoss, aceasta fiind des utilizată pentru problemele de clasificare la calcularea diferenței dintre rezultatul prezis și true labels.

Pentru optimizarea parametrilor modelului în timpul antrenării, am folosit Adam Optimizer.

Am antrenat modelul pentru 100 de epoci, iar pentru fiecare epocă am afișat loss-ul pentru a le putea compara.

5 Rezultate

În urma antrenării modelului, am obținut o acuratețe de 83 % pe datele de test.

Mai multe detalii despre rezultate, precum precizia, recall sau f1-score, pot fi vizualizate în imaginea de mai jos:

```
Test Accuracy: 0.8305084745762712
              precision    recall  f1-score   support

           0       0.83      1.00      0.91      3087
           1       0.50      0.00      0.00       630

    accuracy                           0.83      3717
   macro avg       0.67      0.50      0.46      3717
weighted avg       0.77      0.83      0.75      3717
```

6 Concluzii

Prin urmare, rezultatele obținute în cadrul acestui studiu arată beneficiul utilizării modelelor NLP în detectarea și clasificarea limbajului ofensiv pe social media. Aceste modele pot contribui la crearea unui mediu online mai sigur, mai inclusiv și mai respectuos, facilitând identificarea și eliminarea limbajului ofensiv în timp real.

References

[1] Sohail Akhtar, Valerio Basile, Viviana Patti (2021) *Whose Opinios Matter? Perspective-Aware Models To Identify Opinios Of Hate Speech Victims In Abusive Language Detection*, arXiv:2106.15896v1 [cs.CL].

[2] Shervin Malmasi, Marcos Zampieri (2017) *Detecting Hate Speech in Social Media*, arXiv:1712.06427v2 [cs.CL].

[3] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Rangel, Paolo Rosso, Manuela Sanguinetti (2019) *Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter*

[4] Kaggle *Hate Speech and Offensive Language Dataset*, https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset