OFFENSIVE LANGUAGE CLASSIFICATION

Boboc Ștefan
Dobre Adriana-Lia
Guțu Ștefania-Alexandra
Pirlogea Luciana-Elena

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 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

```
87,3,0,3,0,1,"""@BrokenPiecesmsc: @ItsNotAdam faggot read my tweets after dat k"" it wasn't even funny lol"
88,3,0,3,0,1,"""@BrosConfessions: This bitch was so ungrateful http://t.co/06e77bGwbx"" fr ..... LULWHORE"
89,3,0,3,0,1,"""@CASHandBOOBIES: I been kidnapped yo bitch"""
90,3,3,0,0,0,"""@CB_Baby24: @white_thunduh alsarabsss"" hes a beaner smh you can tell hes a mexican"
91,3,1,2,0,1,"""@CCobey: @AydanMcCoy happy birthday nigs"" Thanks yo"
92,6,1,5,0,1,"""@CHlLDHOODRUINER: when ur teacher tells u that u have homework https://t.co/RKk5vawIj1"" this bitch need to go!!!"
```

PREPROCESAREA DATELOR

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

- eliminarea numelui de utilizator
- eliminarea link-urilor
- eliminarea hashtag-urilor
- eliminarea ampersandului
- eliminarea caracterelor speciale și a cifrelor
- transformarea caracterelor mari în caractere mici
- înlocuirea secvențelor repetitive cu aceeași literă cu două repetiții a acesteia
- eliminarea spațiilor multiple
- lematizare
- eliminarea stopword-urilor
- eliminarea cuvintelor cu mai puțin de două caractere

```
[['woman', 'complain', 'clean', 'house', 'man', 'trash'],
 ['boy', 'dat', 'coldtyga', 'dwn', 'bad', 'cuffin', 'dat', 'hoe', 'place'],
 ['dawg', 'fuck', 'bitch', 'start', 'cry', 'confuse', 'shit'],
 ['look', 'like', 'tranny'],
 ['shit', 'hear', 'true', 'faker', 'bitch', 'tell'],
 ['shit', 'blow', 'meclaim', 'faithful', 'somebody', 'fuck', 'hoe'],
 ['sit', 'hate', 'bitch', 'shit'],
 ['cause', 'tired', 'big', 'bitch', 'come', 'skinny', 'girl'],
 ['bitch'],
 ['hobby', 'include', 'fight', 'mariam', 'bitch'],
 ['keek', 'bitch', 'curve', 'lol', 'walk', 'conversation', 'like', 'smh'],
 ['murda', 'gang', 'bitch', 'gang', 'land'],
 ['hoe', 'smoke', 'loser', 'yea'],
 ['bad', 'bitch', 'thing', 'like'],
 ['bitch']]
```

EXTRAGEREA CARACTERISTICILOR

01 WORD2VEC

```
w = w2v(
    dataset_tweets,
    min_count=3,
    sg=1,  # folosire skip-gram
    window=7
)
```

02 PADDING

```
# padding every tweet to the longest sentence
padded_tweets = pad_sequence([torch.tensor(tweet) for tweet
  in vectorized_sentences], batch_first=True, padding_value=0)
```

03 TENSORII DIN PYTORCH

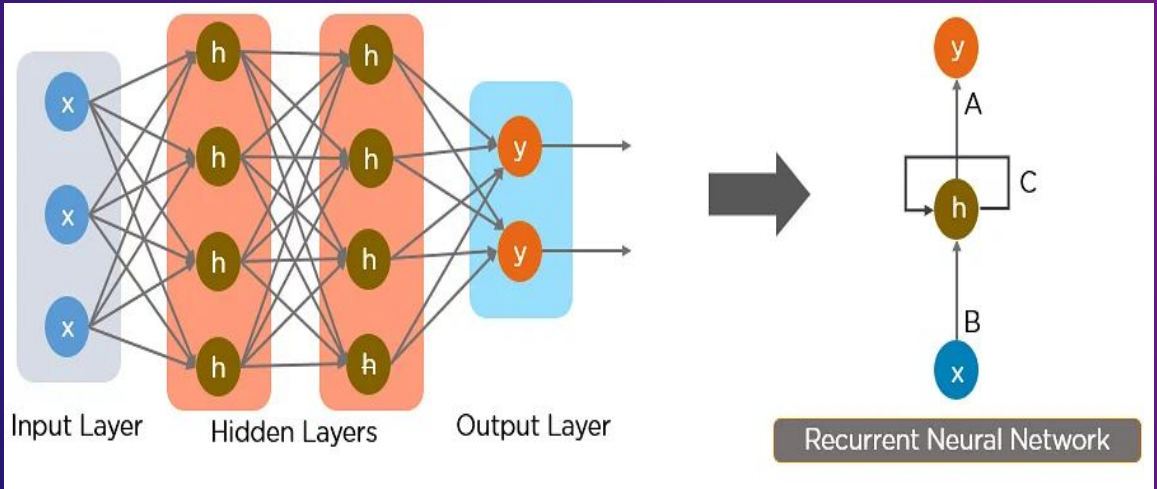
```
# Convertire date in tensori PyTorch

X_train_tensor = torch.tensor(X_train)
X_val_tensor = torch.tensor(X_validation)
X_test_tensor = torch.tensor(X_test)
y_train_tensor = torch.tensor(y_train)
y_val_tensor = torch.tensor(y_validation)
y_test_tensor = torch.tensor(y_test)

train_data = TensorDataset(X_train_tensor, y_train_tensor)
val_data = TensorDataset(X_val_tensor, y_val_tensor)
test_data = TensorDataset(X_test_tensor, y_test_tensor)
```

04 DATA LOADER

```
# dataloaders
batch_size = 50
train_loader = DataLoader(train_data, shuffle=True, batch_size=batch_size)
val_loader = DataLoader(val_data, shuffle=True, batch_size=batch_size)
test_loader = DataLoader(test_data, shuffle=True, batch_size=batch_size)
```

MODEL

- Am împărțit setul de date în trei dataframe-uri, după cum urmează: **70% train**, **15% test**, **15% validare**; fiecare dataframe fiind stocat în batch-uri de câte 50, randomizate
- Am folosit modelul RNN, CrossEntropyLoss și Adam Optimizer:



REZULTATE

- Am antrenat modelul pentru 100 de epoci, iar pentru fiecare epocă am afișat loss-ul pentru a le putea compara
- În urma antrenării modelului, am obținut o **acuratețe de 83%** pe datele de test

```
Epoch 9, Loss: 0.26368284225463867, Val Loss: 0.452763177951177
Epoch 10, Loss: 0.4475571811199188, Val Loss: 0.44804271121819816
Epoch 11, Loss: 0.38624855875968933, Val Loss: 0.448045175075531
Epoch 12, Loss: 0.3779705762863159, Val Loss: 0.44833411594231926
Epoch 13, Loss: 0.3736109435558319, Val Loss: 0.44657398025194806
Epoch 14, Loss: 0.431583970785141, Val Loss: 0.4461916430791219
Epoch 15, Loss: 0.4304144084453583, Val Loss: 0.4504768451054891
```

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

ARTICOLE ASOCIATE

- Whose Opinios Matter? Perspective-Aware Models To Identify Opinios Of Hate Speech Victims In Abusive Language Detection
- Detecting Hate Speech in Social Media
- Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter