Podstawy sztucznej inteligencji

Metody klasyfikacji tekstu
Laboratorium 1 - raport

Piotr Świderski śr 14:40 - 16:10 A

1. Zbiór opisów filmów

W trakcie realizacji ćwiczenia korzystałem z kodów umieszczonych w notebooku. Zbiór treningowy wynosił 90% wszystkich filmów, a zbiór testowy pozostałe 10%. Trening obejmował 5 epok.

```
train_df, test_df = train_test_split(all_data, train_size=0.9)
```

Kod 1. Tworzy zbiory treningowe i testowe

```
ClassificationModel.tokenizer = tokenizer
cls_model_2 = ClassificationModel('roberta', './')
cls_model_2.train_model(train_df, args={"num_train_epochs": 5})
```

Kod 2. Uruchamia trening z 5 epokami

Przy ewaluacji wyników otrzymano następujące wyniki:

TP	TN	FP	FN	acc	precision	recall	F1
119	116	13	8	0.91796875	0,9015151515	0,937007874	0,9189189189

Komentarz: Korzystając z 5 epok uzyskano dobry klasyfikator. Precyzja wynosi ponad 90%.

Korzystając z prostszych metod otrzymano:

ELAPSED: 10.9 S	precision	recall	f1-score	support
0	0.90	0.90	0.90	129
1	0.90	0.90	0.90	127
accuracy			0.90	256
macro avg	0.90	0.90	0.90	256
weighted avg	0.90	0.90	0.90	256

Komentarz: Nawet prostsza metoda Bayesowska pozwoliła uzyskać wyniki o zbliżonej precyzji.

2. Dostosowanie zbioru PolEval

```
# Tworzy z pliku text.txt plik csv

tmp_text = pd.read_csv("training_set_clean_only_text.txt", sep='\n',
engine="python")
tmp_text.to_csv('training_set_clean_only_text.csv', index = None)
tmp_text['Index'] = tmp_text.index
tmp_text.to_csv("training_set_clean_only_text.csv", header=["text", "Index"],
index=False)

# Tworzy z pliku tags.txt plik csv

tmp_tag = pd.read_csv("training_set_clean_only_tags.txt", sep='\n',
engine="python")
tmp_tag['Index'] = tmp_tag.index
tmp_tag.to_csv('training_set_clean_only_tags.csv', index = None)
tmp_tag.to_csv("training_set_clean_only_tags.csv", header=["labels", "Index"],
index=False)

# Łączy dwa piki w jeden

labels = pd.read_csv("training_set_clean_only_tags.csv")
text = pd.read_csv("training_set_clean_only_text.csv")
merged = labels.merge(text, on="Index")
merged.to_csv('twitter.csv', index=False)
```

Kod 3. Za pomocą powyższego kodu dostosowałem plik do wymogów biblioteki

3. Trening i testy dla zbioru PolEval

Zbiór treningowy wynosił 90% wszystkich filmów, a zbiór testowy pozostałe 10%. Trening obejmował 5 epok. Po uruchomieniu testów uzyskano następujące wyniki:

TP	TN	FP	FN	acc	precision	recall	F1
0	930	0	74	0.9262948207171	dzielenie przez 0	0	0

Komentarz: Przy braku modyfikacji parametrów otrzymano wynik o wysokiej dokładności, ale całkowicie bezużyteczny. System w ogóle się nie nauczył rozpoznawać cyberbullyingu i traktował wszystkie wpisy jako neutralne (a było ich 90%). Widać to na tym wyniku:

```
Index(['labels', 'Index', 'text'], dtype='object')
0    8272
1     764
Name: labels, dtype: int64
0    917
1     87
Name: labels, dtype: int64
```

Korzystając z prostszych metod otrzymano poniższy wynik:

ELAPSED: 13.5	precisio	recall	f1-score	support
0	0.95	0.99	0.97	930
1	0.69	0.30	0.42	74
accuracy			0.94	1004
macro avg	0.82	0.64	0.69	1004
weighted avg	0.93	0.94	0.93	1004

Komentarz: Jak widać precyzja wyznaczania neutralnego wpisu wynosi dużo, bo aż 95%, a wpisu nienawistnego 69%, czyli nieco mniej. Dla źle dobranych parametrów prostsze metody naiwne mogą okazać się skuteczniejsze.

4. Modyfikacja hiper-parametry treningu

Warunki	TP	TN	FP	FN	acc	precision	recall	F1
weight=[0.1, 100], train batch size:60, eval_batch_size:50, epoch:7	87	0	917	0	0,08665338645	0,08665338645	1	0,1594867094
weight=[1, 9], train batch size:60, eval_batch_size:50, epoch:5	59	860	57	28	0,9153386454	0,5086206897	0,6781609195	0,5812807882
weight=[100, 0.1], train batch size:60, eval_batch_size:50, epoch:7	0	917	0	87	0,9133466135	dzielenie przez 0	0	0
weight=[0.1, 100], train batch size:60, eval_batch_size:50, epoch:5	87	0	917	0	0,08665338645	0,08665338645	1	0,1594867094

Komentarz: Jak widać najlesze wyniki uzyskano dobierając wagi odpowiednio do stosunku tweetów nienawistnych, do neutralnych. Zwiększenie rozmiaru batcha (domyślnie ustawionego na 8, też polepszyło wynik). W wierszu drugim widać, że udało się uzyskać precyzję na poziomie 50% (w pozostałych przypadkach wynosiła ona mniej niż 1%). Zmniejszenie liczby epok z 7 na 5 (pierwszy i ostatni wiersz) nie miało wpływu na jakość wyników (być może zależało to od błędnie dobranych wag).

ELAPSED: 8.8 S	precision	recall	f1-score	support
0	0.95	0.98	0.97	917
1	0.71	0.46	0.56	87
accuracy			0.94	1004
macro avg	0.83	0.72	0.76	1004
weighted avg	0.93	0.94	0.93	1004

Komentarz: Powyższe wyniki są w zasadzie równym wynikom uzyskanym w punkcie nr 4.

5. Podsumowanie

Na podstawie wyników można dojść do wniosku, że klasyfikator bayesowski daje podobne wyniki (a nawet czasem lepsze), jak transformator. Zważywszy na czas treningu transformatora można uznać, że klasyfikator naiwny jest nawet lepszą metodą. Aczkolwiek wpływ na wynik mógł mieć mój brak intuicji przy doborze parametrów treningowych.