POLITECHNIKA KRAKOWSKA im. T. Kościuszki
Wydział Informatyki i Telekomunikacji

Kierunek studiów: Informatyka
Specjalność: Data Science

Przetwarzanie języka naturalnego

Projekt

Quora Question Pairs

Autorzy
Piotr Dudek DS1
Mariusz Górny DS2
Michał Janiak DS2

Prowadzący
dr Radosław Kycia

Data wykonania
22.05.2022

SPIS TREŚCI

1. CEL PROJKTU I OPIS PROBLEMU

Celem projektu jest wykonanie modelu sztucznej inteligencji, który będzie w stanie wykryć czy dwa pytania pochodzące z platformy Quora[1] oznaczają to samo. Quora to platforma internetowa, która umożliwia dzielenie się wiedzą pomiędzy jej użytkownikami. Powstały w ramach projektu model mógłby być pomocny w wykrywaniu pytań (wątków, tematów), które są duplikatami.

2. OPIS DANYCH

Dane użyte w celu stworzenia i przetestowania modelu pochodzą z turnieju na Kaggle – Quora Question Pairs[2]. Zawierają pary pytań podzielone na zbiory trenujący i testowy. Zbiór trenujący, oprócz 2 pytań, posiada także oznaczenie, czy pytania są duplikatami. Zbiór testowy nie posiada tego oznaczenia, więc w ramach projektu zostanie tylko wykorzystany zbiór trenujący.

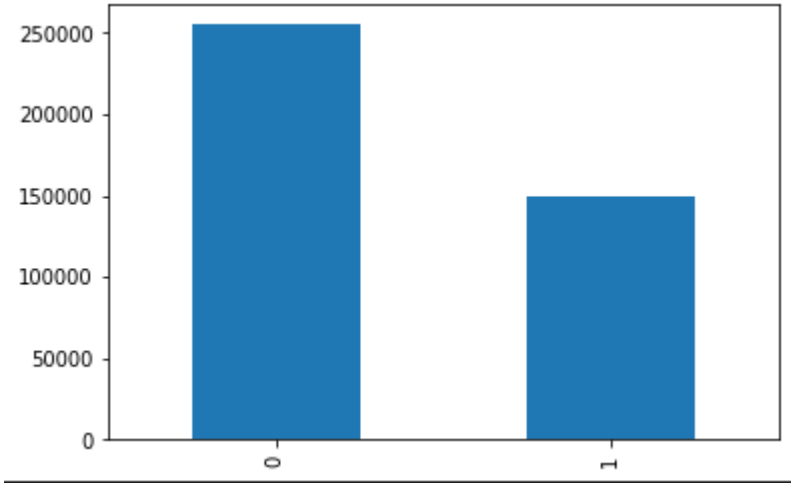
Zbiór danych zwiera 404 287 par pytań, przykładowe pary przedstawia rysunek 1. Wszystkie pytania są po angielsku i zwierają tylko jedno pytanie. Par pytań, które nie są duplikatami jest o około 2/3 więcej, niż par pytań, które nie są duplikatami (rysunek 2). Zbiór został podzielony na podzbiory trenujący (67%) i testujący (20%).

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0
2	2	5	6	How can I increase the speed of my internet co...	How can Internet speed be increased by hacking...	0
3	3	7	8	Why am I mentally very lonely? How can I solve...	Find the remainder when [math]23^{24}[/math] i...	0
4	4	9	10	Which one dissolve in water quikly sugar, salt...	Which fish would survive in salt water?	0
...	...	...	...	...	...	...
404285	404285	433578	379845	How many keywords are there in the Racket prog...	How many keywords are there in PERL Programmin...	0
404286	404286	18840	155606	Do you believe there is life after death?	Is it true that there is life after death?	1
404287	404287	537928	537929	What is one coin?	What's this coin?	0
404288	404288	537930	537931	What is the approx annual cost of living while...	I am having little hairfall problem but I want...	0
404289	404289	537932	537933	What is like to have sex with cousin?	What is it like to have sex with your cousin?	0

404287 rows × 6 columns

Rysunek 1. Przykłady par pytań

[1] Quora - https://pl.quora.com/
[2] Quora Question Pairs - https://www.kaggle.com/competitions/quora-question-pairs/overview



Rysunek 2. Rozkład liczności klas (0 – pytania są różne, 1 - duplikat)

3. PRZETWORZENIE I TOKENIZACJA TEKSTÓW

Na początku wszystkie skróty zostały przekształcone na pełne wyrazy, np. „what's" zostało zmienione na „what is", a „'ve" na „have". Następnie zostały usunięte wyrazy nieznaczące (stop words), lista wyrazów została pobrana z biblioteki Natural Language Toolkit (NLTK) dla języka angielskiego. Potem słowa zostały przetworzone do bazowych postaci za pomocą SnowballStemmer'a. Przykład przetworzonych w ten sposób pytań przedstawia rysunek 3.

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	step step guid invest share market india	step step guid invest share market	0
1	1	3	4	stori kohinoor koh - - noor diamond	would happen indian govern stole kohinoor koh ...	0
2	2	5	6	increas speed internet connect use vpn	internet speed increas hack dns	0
3	3	7	8	mental lone solv	find remaind math 23 ^ 24 math divid 24 23	0
4	4	9	10	one dissolv water quik sugar salt methan carbo...	fish would surviv salt water	0
...	...	...	...	...	...	...
404285	404285	433578	379845	mani keyword racket program languag latest ver...	mani keyword perl program languag latest version	0
404286	404286	18840	155606	believ life death	true life death	1
404287	404287	537928	537929	one coin	coin	0
404288	404288	537930	537931	approx annual cost live studi uic chicago indi...	littl hairfal problem want use hair style prod...	0
404289	404289	537932	537933	like sex cousin	like sex cousin	0

404287 rows × 6 columns

Rysunek 3. Przykładowe pytania po obróbce tekstu

Do tokenizacji tekstów została wykorzystana klasa Tokenizer z biblioteki Keras. Słownik słów został ograniczony do 3000 najczęściej występujących słów. Do tokenizacji został użyty

algorytm TFIDF. W ten sposób przetworzone pytania zostały oddzielnie przedłożone na wejście sieci neuronowej.

4. STRUKTURA I HIPERPARAMETRY SIECI NEURONOWEJ

Do stworzenia i wytrenowania sieci neuronowej została wykorzystane biblioteki keras i tensorflow. Struktura sieci neuronowej:

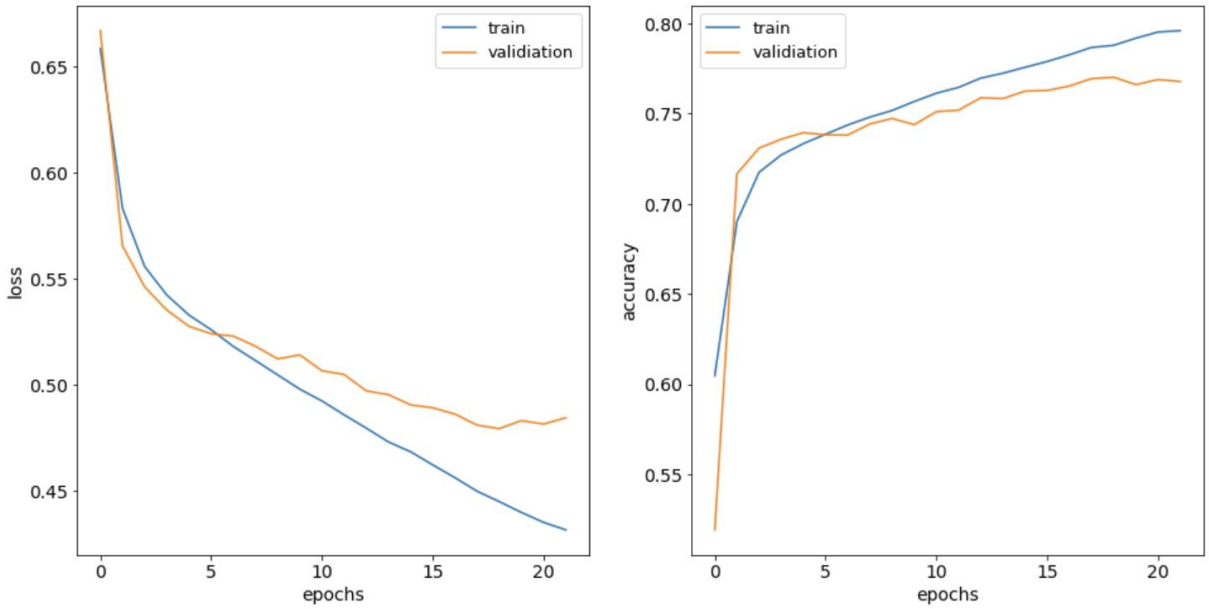
```
Layer (type)                 Output Shape              Param #     Connected to
==================================================================================================
 input_65 (InputLayer)       [(None, 3000)]            0           []

 input_66 (InputLayer)       [(None, 3000)]            0           []

 reshape_72 (Reshape)        (None, 1, 3000)           0           ['input_65[0][0]']

 reshape_73 (Reshape)        (None, 1, 3000)           0           ['input_66[0][0]']

 lstm_86 (LSTM)              (None, 1, 128)            1602048     ['reshape_72[0][0]']

 lstm_87 (LSTM)              (None, 1, 128)            1602048     ['reshape_73[0][0]']

 concatenate_31 (Concatenate)  (None, 1, 256)          0           ['lstm_86[0][0]',
                                                                   'lstm_87[0][0]']

 dropout_54 (Dropout)        (None, 1, 256)            0           ['concatenate_31[0][0]']

 batch_normalization_56 (BatchN  (None, 1, 256)        1024        ['dropout_54[0][0]']
 ormalization)

 lstm_88 (LSTM)              (None, 128)               197120      ['batch_normalization_56[0][0]']

 dropout_55 (Dropout)        (None, 128)               0           ['lstm_88[0][0]']

 batch_normalization_57 (BatchN  (None, 128)           512         ['dropout_55[0][0]']
 ormalization)

 dense_35 (Dense)            (None, 128)               16512       ['batch_normalization_57[0][0]']

 dropout_56 (Dropout)        (None, 128)               0           ['dense_35[0][0]']

 batch_normalization_58 (BatchN  (None, 128)           512         ['dropout_56[0][0]']
 ormalization)

 dense_36 (Dense)            (None, 64)                8256        ['batch_normalization_58[0][0]']

 dropout_57 (Dropout)        (None, 64)                0           ['dense_36[0][0]']

 batch_normalization_59 (BatchN  (None, 64)            256         ['dropout_57[0][0]']
 ormalization)

 dense_37 (Dense)            (None, 1)                 65          ['batch_normalization_59[0][0]']

==================================================================================================
Total params: 3,428,353
Trainable params: 3,427,201
Non-trainable params: 1,152
__________________________________________________________________________________________________
```

Na wejściu sieci znajdują się 2 oddzielne odnogi sieci, każda przyjmująca jedno z pytań. Obie odnogi przyjmują wektor liczb o długości 3000, który jest przetwarzany na matrycę za pomocą warstwy Reshape, aby mógł być następnie przekazany do warstwy LSTM o 128 neuronach.

W ten sposób oddzielnie przetworzone pytania są następnie łączone za pomocą warstwy Concatenate i jeszcze raz przekazywane do warstwy LSTM o 128 neuronach. Dalej znajduje się część feed-forward sieci z 2 warstwami o 128 i 64 neuronach. Wszystkie warstwy ukryte są aktywowane za pomocą funkcji ReLU, natomiast warstwa wyjściowa jest aktywowana funkcją sigmoidalną. Pomiędzy poszczególnymi warstwami sieci znajdują się warstwy normalizujące wartości wyjścia z warstw (BatchNormalization), które zapewniają lepszą jakość uczenia. Zastosowane zostały także warstwy Dropout w celu przeciwdziałania przeuczenia sieci.

5. TRENOWANIE I WALIDACJA

Jako funkcja straty została wybrana metryka binarnej entropi krzyżowej, która lepszą miarą niż dokładność w przypadku nierównych liczności klas. Uczenie zostało przeprowadzone dla 30 epok, z opcją wczesnego zatrzymania, jeżeli dla następnych 3 epok nie będzie poprawy wartości funkcji straty dla zbioru walidującego (20% danych trenujących). Opcja zatrzymania zakończyła trenowanie po 22 epokach.



Rysunek 4. Wartość funkcji straty i dokładności w zależności od epoki dla zbiorów trenującego i walidującego

Końcowy wynik trenowania:

```
Epoch 22/30
424/424 [==============================] - 90s 211ms/step - loss: 0.4315 - accuracy: 0.7960 - val_loss:
0.4844 - val_accuracy: 0.7677
```

Walidacja na zbiorze testowym:

```
261/261 [==============================] - 11s 41ms/step - loss: 0.4852 - accuracy: 0.7689
```

6. PODSUMOWANIE

W ramach projektu został wykonany projekt modelu sieci neuronowej do rozpoznawania duplikatów pytań. Pytania zostały przetworzone do postaci wektorów liczb, a następnie przedłożone na wejście zaprojektowanej sieci neuronowej. Został przedstawiony proces uczenia i testowania sieci neuronowej.

Postała sieć uzyskała dokładność 80% dla zbioru testującego oraz 77% dla walidującego i testowego. Z analizy procesu uczenia (rysunek 4) można wnioskować, że zjawisko przeuczenia zaczęło występować po około 20 epokach i warunek zatrzymujący poprawnie zakończył uczenia po 22 epokach.