Sztuczna inteligencja

Wytyczne projektu

Prowadzący ćwiczenia: Rafał Jaworski
Liczba punktów do zdobycia: 3

Opis projektu

Celem projektu jest napisanie programu komputerowego, do którego wejściem jest tekst opinii o wybranym produkcie, np.:

“Aparat dobrze sprawdza się w terenie. Uchwyci najmiejsze szczegóły. Godny polecenia”

“Mam niecały rok. Jak działał było nawet ok, jak na zapłaconą cenę. Niestety stało się to, co było do przewidzenia. Urządzenie padło. Zaczyna się opisywana na forach "trudna" droga do reklamacji. Brak gwarancji door to door, to juz wiem. Jednak lepiej nieco dopłacić i nabyć sprzęt, który mozna zanieść do sprzedawcy i reklamować.”

Na wyjściu powinna pojawić się ocena w skali 0-5, odzwierciedlająca poziom zadowolenia autora opinii z opisywanego produktu.

Ogólna metodologia rozwiązania

Do rozwiązania zadania należy użyć mechanizmu uczenia maszynowego. Powinien być on wytrenowany na faktycznych danych postaci komentarz-ocena, np.:

Komentarz	Ocena
“Aparat dobrze sprawdza się w terenie. Uchwyci najmiejsze szczegóły. Godny polecenia”	5/5
“Mam niecały rok. Jak działał było nawet ok, jak na zapłaconą cenę. Niestety stało się to, co było do przewidzenia. Urządzenie padło. Zaczyna się opisywana na forach "trudna" droga do reklamacji. Brak gwarancji door to door, to juz wiem. Jednak lepiej nieco dopłacić i nabyć sprzęt, który mozna zanieść do sprzedawcy i reklamować.”	2/5

Dokładność przewidywania oceny przez stworzony program należy przetestować w procesie ewaluacji.

Sposób pracy

Projekt można realizować w grupach maksymalnie 3 osobowych. Dozwolone jest zatem również pracowanie w parach oraz indywidualnie.
Efekty pracy należy opisać w raporcie i wysłać na adres rjawor@amu.edu.pl w terminie do 14 czerwca 2019 roku. Raport powinien zawierać:

- skład grupy,
- opis pozyskanych danych (ilość, skąd zostały pobrane),
- opis mechanizmu przewidującego (w szczególności - jakie cechy zostały użyte),
- wyniki ewaluacji.

Wszelkie pytania proszę kierować mailem na wskazany powyżej adres. Istnieje także możliwość umówienia się na dyżur w tygodniu (wtorek 12:00-13:00).

Szczegóły implementacji

Faza pierwsza - pozyskanie danych

W projekcie dane należy pozyskać samodzielnie. Sugeruję, aby użyć do tego popularnego portalu ceneo.pl. Należy wyszukać w nim ocenionych gwiazdkami opinii dotyczących wybranego produktu, a następnie je ściągnąć. Minimalna liczba opinii to 2000. Technicznie, aby ściągnąć zawartość strony można posłużyć się następującym kodem w Pythonie:

```
from urllib.request import urlopen html =
urlopen("http://www.stackoverflow.com/").read().decode('utf-8')
```

Aby na tak ściągniętej stronie wyszukać przydatnych informacji (tj. treści opinii oraz gwiazdek), najlepiej użyć wyrażeń regularnych, a w szczególności funkcji re.findall.

Faza druga - mechanizm zgadujący

Po pozyskaniu danych możemy użyć ich do zbudowania mechanizmu zgadującego ocenę opinii na podstawie tekstu opinii. W tym celu używamy oprogramowania Vowpal Wabbit (https://github.com/VowpalWabbit/vowpal_wabbit/wiki). Polecam instalację tego oprogramowania w systemie (działa pod Windows i Linux) i używanie go spod linii poleceń.
Program ten działa na plikach w następującym formacie:
`wartość | cecha1:wartość_cechy1 cecha2:wartość_cechy2 ....`

W każdej linii znajdują się dane dotyczące jednego obiektu (w naszym przypadku - jednej opinii). Nazwy cech wybieramy my. Po nazwie cechy następuje obowiązkowy dwukropek, a następnie liczbowa wartość tej cechy.

W przypadku opinii o produktach, wartością będzie liczba gwiazdek. Dobór cech zależy natomiast od nas. Możemy wybrać jakiekolwiek cechy opinii, które w naszym mniemaniu mogą korelować z liczbą gwiazdek, które uzyskała ta opinia. Np.:

```
4 | length:45 n_dobry:2 n_zly:0
2 | length:120 n_dobry:0 n_zly:3
```

Powyższy przykład opisuje dwie opinie. Pierwsza jest oceniona na 4 gwiazdki, jej długość (w znakach) wynosi 45. Występują w niej dwa użycia słowa “dobry” i nie ma w niej słowa “zły”. Druga opinia natomiast ma długość 120, słowo “dobry” w niej nie występuje, natomiast słowo “zły” występuje aż 3 razy.

Aby przetworzyć tekstową opinię do formatu Vowpal Wabbita należy napisać sobie program, którego zadaniem będzie ekstrakcja cech. Im więcej cech wybierzemy, tym lepiej. Normalną praktyką jest używanie kilkunastu, kilkudziesięciu, a nawet kilkuset cech. Ze względu na swoją specyfikę, Vowpal Wabbit będzie działał szybko nawet dla dużego zbioru cech. Ponadto, sam nauczy się, które cechy należy brać pod uwagę przy przewidywaniu oceny. Inne pomysły na cechy:

- Czy opinia jest pisana samymi wielkimi literami (0- nie, 1 - tak)
- Liczba emotikonów pozytywnych/negatywnych
- Liczba wystąpień wulgaryzmów (na Ceneo są one cenzurowane gwiadkami)
- ….

Kiedy stworzymy plik z danymi do trenowania train.txt, dokonujemy trenowania przy użyciu komendy:

```
vw -f comments.model train.txt
```

Powoduje to wytrenowanie modelu na pliku z danymi trenującymi i zapisanie modelu do pliku comments.model.

Po wytrenowaniu modelu można przystąpić do rozwiązania głównego problemu: przewidzenia oceny opinii na podstawie jej tekstu. W tym celu należy najpierw przetworzyć opinię na cechy, korzystając z tego samego mechanizmu, którego użyliśmy do przygotowania danych trenujących do Vowpal Wabbita. Mając na wejściu nową opinię, np:

“To nie jest całkowicie dobry produkt, ale też nie jest zły”, przetwarzamy ją na:

```
0 | length:58 n_dobry:1 n_zły:1
```

(przy założeniu, że ograniczyliśmy się tylko do 3 podanych wcześniej cech).

Wartość 0 w tym przypadku nie ma znaczenia, gdyż nie wiemy, jaką ocenę powinna mieć ta opinia. Ocena ta ma zostać wyliczona przez Vowpal Wabbita na podstawie wytrenowanego wcześniej modelu. Musimy jednak wpisać coś (np. 0), żeby zachować format danych Vowpala.

Jeśli cechy opinii, których oceny chcemy obliczyć, zapiszemy w pliku test.txt, wtedy uruchamiamy przewidywanie komendą:

```
vw -i comments.model -t test.txt -p /dev/stdout --quiet
```

Faza trzecia - ewaluacja

Ostatnim bardzo ważnym krokiem jest ewaluacja naszego programu zgadującego. Powinna ona przebiegać według następującego schematu. Po pierwsze, dzielimy zbiór ściągniętych opinii na dwie części w stosunku 90%-10%. 90% danych używamy do trenowania, resztę odkładając na bok.

Następnie, uruchamiamy mechanizm wytrenowany zbiorem 90% danych na pozostałych 10% danych. Otrzymujemy w ten sposób dane następującej postaci:

Opinia	Ocena prawdziwa (z danych)	Ocena podana przez Vowpala
opinia1	3.5	4.2
opinia2	5	4.7

Powinniśmy mieć kilkaset takich wierszy.

W każdym wierszu obliczamy różnicę pomiędzy faktyczną oceną opinii a tą podaną przez Vowpal Wabbit. W naszym przypadku mamy 0.7 w pierwszym wierszu i 0.3 w drugim. Wynikiem ewaluacji jest średnia tych różnic - w naszym przypadku wynosząca 0.5.

Byłoby dobrze, gdyby udało się Państwu uzyskać wynik ewaluacji w granicach 1.0.