Anliza sentymentu - specyfikacja

Aleksander Obuchowski

Listopad 2018

Spis treści

1 Wstęp **3**
- 1.1 Cel . . . 3
- 1.2 Zakres . . . 3
- 1.3 Definicje akronimy i skróty . . . 3
- 1.4 Referencje . . . 3
- 1.5 Krótki przegląd . . . 3

2 Opis ogólny **4**
- 2.1 Walory użytkowe i przydatność projektowanego systemu . . . 4
- 2.2 Ogólne możliwości projektowanego systemu . . . 4
- 2.3 Ogólne ograniczenia . . . 4
- 2.4 Środowisko operacyjne . . . 4
- 2.5 Charakterystyka użytkownika . . . 4

3 Specyficzne Wymagania **5**
- 3.1 Interefejs . . . 5
  - 3.1.1 preprocesing.py . . . 5
  - 3.1.2 learning.py . . . 5
  - 3.1.3 classifier.py.py . . . 5
  - 3.1.4 accuracy.py . . . 5
  - 3.1.5 matrix.log . . . 5
- 3.2 Wymagania dotyczące wydajności systemu . . . 5
- 3.3 Wymagania dotyczące zewnętrzych interfejsów . . . 5
- 3.4 Wymagania dotyczące zasobów . . . 6
  - 3.4.1 Wymagania zalecane . . . 6
- 3.5 Wymagania dotyczące weryfikacji . . . 6
- 3.6 Wymagania dotyczące sposób testowania . . . 6
- 3.7 Wymagania dotyczące dokumentacji . . . 6
- 3.8 Wymagania dotyczące ochrony . . . 6
- 3.9 Wymagania dotyczące przenośności . . . 6
- 3.10 Wymagania dotyczące jakości . . . 6
- 3.11 Wymagania dotyczące niezowodności . . . 6
- 3.12 Wymagania dotyczące bezpieczeństwa . . . 6

4 Dodatki **7**
- 4.1 Harmonogram prac nad projektem . . . 7

Streszczenie

Celem programu jest pozyskanie danych zawierających opisowe rencenzje filmów oraz ich ocenę w skali 1-10 a następnie na ich podstawie, metodami naiwnego klasyfikatora Bayesa storzenie słownika sentymentu wyrazów użytych w recenzjach, który w późniejszym etapie mógłby zostać użytu to oceniania podanych przez użytkownika zdań pod kątem ich pozytywności.

Rozdział 1

Wstęp

1.1 Cel

Bazowym celem oprogramowania jest stoworzenie słownika sentymentu dla języka polskiego mogącego posłużyć w badaniach nad analizą teksu.

1.2 Zakres

Oprogramowanie udostępnione zostanie w serwisie github.com

1.3 Definicje akronimy i skróty

1.4 Referencje

hackmd.io/Nz9iiJDVRCujYjQJ3HG7OA

1.5 Krótki przegląd

Dokument przedstawia ogólny opis oprogramowania, główne założenia, wymagania, oraz dokładny opis funckcji występujących w oprogramowaniu. Dodatkowo przedstawiony został harmonogram pracy nad oprogramowaniem.

Rozdział 2

Opis ogólny

2.1 Walory użytkowe i przydatność projektowanego systemu

Główną zaleta oprogramowania jest stowrzenie narzedzia do analizy wypowiedzi w języku polskim.

2.2 Ogólne możliwości projektowanego systemu

System jest w stanie pozyskać dane odnośnie recenzji filmów ze strony mediakrytyk.pl, zapisać je w formacie csv, nastęnie na ich podstawie stworzyć słownik w formacie json w kórym zawarte są inforamcje odnośnie liczby występowania słów w kontekście pozytywnym i negatywnym, po czym na tej postawie stworzyć słownik sentymentu w myśl naiwnego klasyfikatora Bayesa, po czym przy pomocy danych w nim zawartych określić czy zdanie wpisane przez użytkownieka jest pozytywne czy negoatywne. System posiada równierz metodę analizy klasyfikatora w formie tablicy pomyłek.

2.3 Ogólne ograniczenia

System w obecnej formie jest w stanie pozyskać dane jedynie ze strony mediakrytyk.pl.

2.4 Środowisko operacyjne

Oprogramowanie zostało wykonane w języku python w wersji 3.6 wykorzysując bublioteki io,json oraz csv.

2.5 Charakterystyka użytkownika

Oprogramowanie zaprojektowane jest z myślą o studentach prowadzących badania nad analizą tekstu w języku polskim.

Rozdział 3

Specyficzne Wymagania

3.1 Interefejs

Interfejs przedstawiony jest w formie kosoli. Użytkowik wywołuje kolejne programy przy pomocy wiersza poleceń.

3.1.1 preprocesing.py

preprocesing.py wykorzystuje dane zawarte w pliku data.csv i na ich podsatwie tworzy plik dictionary.json który zawiera informacje odnośnie tego ile razy dane słowo zostało użyte w kontekście pozytywnym i negatywnym.

3.1.2 learning.py

preprocesing.py wykorzystuje dane zawarte w pliku dictionary.json i na ich podstawie tworzy plik setiment.json który zawiera informacje o tym jak pozytywne i negatywne jest dane słowo.

3.1.3 classifier.py.py

classifier.py wykorzysuje dane zawarte w pliku sentiment.csv i na ich podsatwie klasyfikuje zadane mu zdanie pod kątem tego jak pozytywne jest.

3.1.4 accuracy.py

accuracy.py. określa poziom dopasowania klasyfikatora na podstawie opisanego wcześniej zbioru danych; tworzy plik matrix.log

3.1.5 matrix.log

Program accuracy.py zwraca macierz pomyłek w pliku matrix$_{l}og.txtzawiercyminforamcjeodnonieda$

3.2 Wymagania dotyczące wydajności systemu

Brak szczególnych wymagań.

3.3 Wymagania dotyczące zewnętrzych interfejsów

Nie dotyczy

3.4 Wymagania dotyczące zasobów

3.4.1 Wymagania zalecane

Ze względu na brak testów, wymagania podane zgodnie z maszyną na której tworzone jest oprogramowanie:
Procesor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, 1801 MHz, Rdzenie: 4, Procesory logiczne: 8 (lub odpowiadające)
RAM: 8GB
Karta Graficzna: NVIDIA GeForce® MX150 (lub odpowiadająca)

3.5 Wymagania dotyczące weryfikacji

Weryfikacja przeprowadzona zostanie w systemie cotygodniowych raportów o zmianach w oprogramowaniu.

3.6 Wymagania dotyczące sposób testowania

Oprogramowanie zostanie przetestowane korzystając z funcki acccuracy.py celem dobrania optymalnych parametrów klasyfikatora przyczyniających się do jak najlepszego wyniku w macierzy pomyłek

3.7 Wymagania dotyczące dokumentacji

Zmiany w kodzie oprogramowania widoczne będą na odpowiadającej mu stronie w serwisie github.com

3.8 Wymagania dotyczące ochrony

Nie dotyczy.

3.9 Wymagania dotyczące przenośności

Oprogramowanie zostanie opracowane w taki sposób aby każdy z odowiednim środowiskiem mógł je uruchomić.

3.10 Wymagania dotyczące jakości

Oprogramowanie powinno być zoptymalizowane

3.11 Wymagania dotyczące niezowodności

Oprogramowanie powinno być stabilne.

3.12 Wymagania dotyczące bezpieczeństwa

Nie dotyczy.

Rozdział 4

Dodatki

4.1 Harmonogram prac nad projektem

Data	Cel
28.11.18	Storzenie podstawowych funkcji oprogramowania
16.12.18	Usprawnienie klasyfikatora
20.12.18	Optymalizacja interfejsu
24.12.18	Testowanie klasyfikatora
31.12.18 godzina 24:00	Optymalizacja klasyfikatora na podstawie wcześniejszych testów
12.01.19	Końcowe testy i doporacowanie interfejsu