

spis treści

wprowadzenie xvii

wstęp xiv

podziękowania xxvi

o książce xxiv

o autorach xxxii

o ilustracji na okładce xxxiv

CZĘŚĆ 1 ■ MÓWIĄCE MASZYNY 1

1. Pakiety myśli (przegląd NLP) 3

1.1. Język naturalny a język programowania 4

1.2. Magia 4

1.2.1. Maszyny prowadzące konwersację 5

1.2.2. Matematyka 6

1.3. Zastosowania praktyczne 8

1.4. Język widziany „oczyma” komputera 10

1.4.1. Język zamków 10

1.4.2. Wyrażenia regularne 11

1.4.3. Prosty chatbot 13

1.4.4. Inny sposób 16

- 1.5. Krótkie spojrzenie na hiperprzestrzeń 20
- 1.6. Kolejność słów i gramatyka 22
- 1.7. Potok języka naturalnego chatbota 23
- 1.8. Szczegóły przetwarzania 26
- 1.9. IQ języka naturalnego 28
- 2. Zbuduj swój słownik (tokenizacja słów) 32**
 - 2.1. Wyzwania (wprowadzenie do stemmingu) 34
 - 2.2. Tworzenie swojego słownika za pomocą tokenizatora 35
 - 2.2.1. Iloczyn skalarny 44
 - 2.2.2. Pomiar nakładania się wektorów BoW 44
 - 2.2.3. Poprawianie tokenów 45
 - Jak działają wyrażenia regularne 46 ■ Poprawione wyrażenia regularne do podziału słów 47 ■ Formy skrócone 50
 - 2.2.4. Rozszerzenie słownika za pomocą n -gramów 50
 - N-gramy 51 ■ Stop listy 54
 - 2.2.5. Normalizacja słownika 57
 - Ujednolicanie wielkości liter 57 ■ Stemming 59 ■ Lematyzacja 62
 - Przypadki użycia 64
 - 2.3. Wydźwięk 65
 - 2.3.1. VADER – analizator wydźwięku oparty na regułach 67
 - 2.3.2. Naiwny klasyfikator bayesowski 68
- 3. Matematyka na słowach (wektory TD-IDF) 73**
 - 3.1. Wektor BoW 74
 - 3.2. Wektoryzacja 79
 - 3.2.1. Przestrzenie wektorowe 82
 - 3.3. Prawo Zipfa 87
 - 3.4. Modelowanie tematyczne 89
 - 3.4.1. Powrót Zipfa 92
 - 3.4.2. Ranking trafności 94
 - 3.4.3. Narzędzia 96
 - 3.4.4. Inne możliwości 97

- 3.4.5. Okapi BM25 98
- 3.4.6. Co dalej 99

4. Odnajdowanie znaczenia w licznikach słów (analiza semantyczna)	101
4.1. Od liczników słów do wyników dla tematów	103
4.1.1. Wektory TF-IDF i lematyzacja	103
4.1.2. Wektory tematyczne	104
4.1.3. Eksperyment myślowy	105
4.1.4. Algorytm do oceny tematów	110
„Kuzyni” LSA	111
4.1.5. Klasyfikator LDA	111
Inny „kuzyn”	115
4.2. Analiza utajonych własności semantycznych (LSA)	116
4.2.1. Wasz eksperyment myślowy staje się prawdziwy	118
Gra w Mad libs	119
4.3. Rozkład według wartości osobliwej	121
4.3.1. U – lewostronne wektory osobliwe	123
4.3.2. S – wartości osobliwe	124
4.3.3. V^T – prawostronne wektory osobliwe	125
4.3.4. Orientacja macierzy SVD	126
4.3.5. Obcinanie tematów	127
4.4. Analiza głównych składowych	128
4.4.1. PCA dla wektorów 3D	130
4.4.2. Przestańmy szaleć i wróćmy do NLP	131
4.4.3. Stosowanie PCA do semantycznej analizy komunikatów SMS	134
4.4.4. Używanie obciętego SVD do analizy semantycznej komunikatu SMS	136
4.4.5. Jak dobrze działa LSA przy klasyfikacji spamu	137
Rozszerzenia LSA i SVD	139
4.5. Ukryta alokacja Dirichleta (LDiA)	140
4.5.1. Idea LDiA	141
4.5.2. Model tematyczny LDiA dla komunikatów SMS	143
4.5.3. LDiA + LDA = klasyfikator spamu	146
4.5.4. Uczciwsze porównanie: 32 tematy LDiA	148
4.6. Odległość i podobieństwo	149

- 4.7. Sterowanie za pomocą informacji zwrotnej 152
 - 4.7.1. Liniowa analiza dyskryminacyjna 153
- 4.8. Moc wektorów tematycznych 155
 - 4.8.1. Wyszukiwanie semantyczne 156
 - 4.8.2. Ulepszenia 159

CZĘŚĆ 2 ■ GŁĘBSZE UCZENIE SIĘ (SIECI NEURONOWE) 161

5. Sieci neuronowe krok po kroku (perceptrony i propagacja wsteczna) 163

- 5.1. Sieci neuronowe – lista składników 164
 - 5.1.1. Perceptron 165
 - 5.1.2. Perceptron numeryczny 165
 - 5.1.3. Zboczenie z drogi spowodowane odchyleniem 166
 - Neuron w Pythonie 168 ■ Klasa tkwi w sesji 169 ■ Uczenie się logiki to czysta frajda 170 ■ Następny krok 172 ■ Koniec drugiej zimy sztucznej inteligencji 175 ■ Propagacja wsteczna 176 ■ Zrózniczkujmy wszystko 179
 - 5.1.4. Poszusujemy – powierzchnia błędu 181
 - 5.1.5. Z wyciągu prosto na stok 182
 - 5.1.6. Udoskonalmy to nieco 182
 - 5.1.7. Keras: sieci neuronowe w Pythonie 184
 - 5.1.8. Wprzód i w głąb 187
 - 5.1.9. Normalizacja: stylowe wejście 188

6. Wnioskowanie przy użyciu wektorów słów (Word2vec) 190

- 6.1. Zapytania semantyczne i analogie 191
 - 6.1.1. Pytania o analogię 192
- 6.2. Wektory słów 193
 - 6.2.1. Wnioskowanie zorientowane wektorowo 197
 - Jeszcze więcej powodów, by korzystać z wektorów słów 199
 - 6.2.2. Jak obliczać reprezentacje Word2vec 200
 - Podejście skip-gram 201 ■ Czym jest softmax? 202 ■ W jaki sposób sieć uczy się reprezentacji wektorowej? 203 ■ Odnajdywanie wektorów słów za pomocą algebry liniowej 205 ■ Podejście CBoW 205 ■ Skip-gram a CBoW. Kiedy korzystać z którego podejścia? 207 ■ Triki obliczeniowe Word2vec 207 ■ Częste bigramy 207 ■ Podpróbkowanie często występujących tokenów 208 ■ Próbkowanie negatywne 209

- 6.2.3. Jak korzystać z modułu gensim.word2vec 209
- 6.2.4. Jak wygenerować własne reprezentacje wektorów słów 212
 - Kroki przetwarzania wstępnego 212 ■ Szkolenie dziedzinowego modelu Word2vec 213
- 6.2.5. Word2vec a GloVe (Global Vectors) 214
- 6.2.6. fastText 215
 - Jak korzystać z gotowych modeli fastText 216
- 6.2.7. Word2vec a LSA 216
- 6.2.8. Wizualizacja związków między słowami 217
- 6.2.9. Nienaturalne słowa 224
- 6.2.10. Doc2vec i podobieństwo dokumentów 225
 - Jak wyczytać wektory dokumentów 226
- 7. Kolejność słów i konwolucyjne sieci neuronowe (CNN) 228**
 - 7.1. Uczenie się znaczenia 230
 - 7.2. Zestaw narzędzi 232
 - 7.3. Konwolucyjne sieci neuronowe 233
 - 7.3.1. Elementy składowe 233
 - 7.3.2. Długość kroku 235
 - 7.3.3. Budowa filtra 235
 - 7.3.4. Uzupelnianie 237
 - Potok konwolucyjny 238
 - 7.3.5. Uczenie 238
 - 7.4. Zaiste, wąskie okna 239
 - 7.4.1. Implementacja w Kerasie: przygotowanie danych 240
 - 7.4.2. Architektura konwolucyjnej sieci neuronowej. 247
 - 7.4.3. Warstwa łącząca (*pooling*) 247
 - 7.4.4. Dropout 250
 - 7.4.5. Wisienka na torcie 251
 - Optymalizacja 252 ■ Dopasowanie (fit) 252
 - 7.4.6. Czas zabrać się za naukę (trening) 253
 - 7.4.7. Użycie modelu w potoku 255
 - 7.4.8. Gdzie pójdziecie dalej? 256

8. Zapętlone (rekurencyjne) sieci neuronowe (RNN) 259

- 8.1. Zapamiętywanie za pomocą sieci rekurencyjnych 262
 - 8.1.1. Propagacja wsteczna przez czas 267
 - Tl;Dr – Krótka rekapitulacja 269
 - 8.1.2. Kiedy i co aktualizować? 269
 - Czy jednak obchodzi was to, co wyszło z wcześniejszych kroków? 270
 - 8.1.3. Rekapitulacja 271
 - 8.1.4. Zawsze jest jakiś haczyk 272
 - 8.1.5. Rekurencyjne sieci neuronowe z Kerasem 272
- 8.2. Składanie w całość 277
- 8.3. Nauczmy się czegoś o przeszłości 279
- 8.4. Hiperparametry 279
- 8.5. Przewidywanie 283
 - 8.5.1. Stanowość 284
 - 8.5.2. Ulica dwukierunkowa 284
 - 8.5.3. Co to takiego? 286

9. Lepsza pamięć dzięki sieciom LSTM 288

- 9.1. LSTM 290
 - 9.1.1. Propagacja wsteczna przez czas 299
 - W praktyce 299
 - 9.1.2. Próba ognia 301
 - 9.1.3. Brudne dane 303
 - 9.1.4. Powrót do brudnych danych 306
 - 9.1.5. Słowa są trudne. Litery są prostsze 307
 - 9.1.6. Kolej na rozmowę 312
 - 9.1.7. Zwrot ku klarownej mowie 314
 - Zwiększanie użyteczności generatora 322
 - 9.1.8. Jak mówić i co mówić 322
 - 9.1.9. Inne rodzaje pamięci 322
 - 9.1.10. Idąc głębiej 323

10. Modele ciąg-ciąg i uwaga (attention) 326

- 10.1. Architektura koder-dekoder 327

10.1.1.	Dekodowanie myśli	328
10.1.2.	Wygląda znajomo?	330
10.1.3.	Konwersacja ciąg-ciąg	332
10.1.4.	Powtórzenie LSTM	332
10.2.	Składanie potoku ciąg-ciąg	334
10.2.1.	Przygotowanie naszego zbioru danych do szkolenia ciąg-ciąg	334
10.2.2.	Model ciąg-ciąg w Kerasie	335
10.2.3.	Koder ciągów	336
10.2.4.	Koder myśli	337
10.2.5.	Składanie sieci ciąg-ciąg	338
10.3.	Szkolenie sieci ciąg-ciąg	339
10.3.1.	Generowanie ciągów wyjściowych	340
10.4.	Budowanie chatbota przy użyciu sieci ciąg-ciąg	341
10.4.1.	Przygotowanie korpusu do szkolenia	342
10.4.2.	Budowanie słownika znaków	343
10.4.3.	Generowanie zbiorów treningowych zakodowanych metodą 1 z n	343
10.4.4.	Uczenie chatbota ciąg-ciąg	344
10.4.5.	Składanie modelu do generowania ciągów	345
10.4.6.	Przewidywanie ciągu	345
10.4.7.	Generowanie odpowiedzi	346
10.4.8.	Rozmowa z waszym chatbotem	347
10.5.	Ulepszenia	347
10.5.1.	Redukcja złożoności treningu za pomocą sortowania danych (<i>bucketing</i>)	347
10.5.2.	Uwaga (<i>attention</i>)	348
10.6.	W świecie rzeczywistym	350
CZĘŚĆ 3 ■ PRZEJŚCIE DO RZECZYWISTOŚCI (PRAWDZIWE PROBLEMY NLP)		353
11. Ekstrakcja informacji (rozpoznawanie jednostek nazewniczych i odpowiadanie na pytania)		
11.1.	Jednostki nazewnicze i relacje	356
11.1.1.	Baza wiedzy	356
11.1.2.	Ekstrakcja informacji	359
11.2.	Regularne wzorce	359

- 11.2.1. Wyrażenia regularne 360
- 11.2.2. Ekstrakcja informacji jako ekstrakcja cech z wykorzystaniem uczenia się maszyn 361
- 11.3. Informacje warte wyodrębnienia 363
 - 11.3.1. Ekstrakcja lokalizacji GPS 363
 - 11.3.2. Ekstrakcja dat 364
- 11.4. Wyodrębnianie relacji 369
 - 11.4.1. Znakowanie częściami mowy 370
 - 11.4.2. Normalizacja jednostek nazewniczych 374
 - 11.4.3. Normalizacja i wyodrębnianie relacji 375
 - 11.4.4. Wzorce słów 375
 - 11.4.5. Segmentacja 376
 - Segmentacja na zdania 377
 - 11.4.6. Dlaczego split('!?!') nie będzie działać? 377
 - 11.4.7. Segmentacja na zdania za pomocą wyrażeń regularnych 378
- 11.5. W prawdziwym świecie 380
- 12. Pogaduszki (silniki dialogowe) 382**
 - 12.1. Umiejętności językowe 383
 - 12.1.1. Nowoczesne podejścia 384
 - Systemy dialogowe odpowiadające na pytania 386 ■ Wirtualni asystenci 386
 - Chatboty konwersacyjne 387 ■ Chatboty marketingowe 388 ■ Zarządzanie społecznością 388 ■ Obsługa klienta 389 ■ Terapia 390
 - 12.1.2. Podejście hybrydowe 390
 - 12.2. Podejście polegające na dopasowaniu do wzorców 391
 - 12.2.1. Chatbot oparty na dopasowaniu do wzorca i AIML 392
 - AIML 1.0 393 ■ Interpreter AIML w Pythonie 394
 - 12.2.2. Sieciowe spojrzenie na dopasowanie do wzorców 399
 - 12.3. Oparcie na wiedzy 400
 - 12.4. Wyszukiwanie 402
 - 12.4.1. Problem kontekstu 403
 - 12.4.2. Przykładowy chatbot oparty na wyszukiwaniu danych 404
 - 12.4.3. Chatbot oparty na wyszukiwaniu 408
 - 12.5. Modele generatywne 410

- 12.5.1. Czat na temat NLPIA 411
- 12.5.2. Zalety i wady każdego podejścia 413
- 12.6. Napęd na cztery koła 414
 - 12.6.1. Will osiąga sukces 414
 - Instalowanie Willa 414 ■ Hello WILL 414
- 12.7. Proces projektowania 415
- 12.8. Sztuczki 418
 - 12.8.1. Zadawanie pytań z przewidywalnymi odpowiedziami 418
 - 12.8.2. Bycie zabawnym 419
 - 12.8.3. Gdy wszystko inne zawiedzie, trzeba wyszukać 419
 - 12.8.4. Bycie popularnym 419
 - 12.8.5. Być łącznikiem 420
 - 12.8.6. Stawanie się emocjonalnym 420
- 12.9. W świecie rzeczywistym 420
- 13. Skalowanie (optymalizacja, zrównoleglanie i przetwarzanie wsadowe) 422**
 - 13.1. Zbyt wiele dobrego (danych) 423
 - 13.2. Optymalizowanie algorytmów NLP 423
 - 13.2.1. Indeksowanie 424
 - 13.2.2. Zaawansowane indeksowanie 425
 - 13.2.3. Zaawansowane indeksowanie za pomocą Annoy 427
 - 13.2.4. Po co w ogóle stosować indeksy przybliżone? 432
 - 13.2.5. Obejście indeksowania: dyskretyzacja 433
 - 13.3. Algorytmy ze stałą pamięcią RAM 434
 - 13.3.1. Gensim 434
 - 13.3.2. Obliczenia graficzne 435
 - 13.4. Zrównoleglanie waszych obliczeń NLP 436
 - 13.4.1. Trenowanie modeli NLP na procesorach graficznych (GPU) 436
 - 13.4.2. Wynajem a kupno 438
 - 13.4.3. Opcje wynajmu GPU 438
 - 13.4.4. Jednostki przetwarzania tensorowego 439
 - 13.5. Zmniejszanie zużycia pamięci podczas trenowania modeli 440
 - 13.6. Uzyskiwanie wglądu w model za pomocą TensorBoard 442
 - 13.6.1. Jak wizualizować zanurzenia słów 443

dodatek A	Nasze narzędzia NLP	447
dodatek B	Swobodny Python i wyrażenia regularne	455
dodatek C	Wektory i macierze (podstawy algebry liniowej)	461
dodatek D	Narzędzia i techniki uczenia się maszyn	467
dodatek E	Ustawianie własnego AWS GPU	481
dodatek F	Mieszanie wrażliwe na lokalizację (LSH)	495
źródła		503
słownik		513
indeks		520
posłowie do wydania polskiego		535