



DEITEL® DEVELOPER SERIES



Python®

dla programistów

z analizami przypadków
wprowadzającymi w tematykę
sztucznej inteligencji

- ▶ Przetwarzanie języka naturalnego
- ▶ Eksploracja danych Twittera®
- ▶ IBM® Watson™
- ▶ Uczenie maszynowe z użyciem scikit-learn®
- ▶ Głębokie uczenie z użyciem biblioteki Keras
- ▶ Big Data, Hadoop®, Spark™, NoSQL i usługi chmurowe
- ▶ Internet rzeczy (IoT)
- ▶ Standardowa biblioteka Pythona
- ▶ Biblioteki Data Science: NumPy, Pandas, SciPy, NLTK, TextBlob, Tweepy, Matplotlib, Seaborn, Folium i inne

PAUL DEITEL • HARVEY DEITEL

Helion

Tytuł oryginału: Python for Programmers: with Big Data and Artificial Intelligence Case Studies

Tłumaczenie: Andrzej Grażyński

ISBN: 978-83-283-6394-6

Authorized translation from the English language edition, entitled PYTHON FOR PROGRAMMERS, 1st Edition by DEITEL, PAUL J.; DEITEL, HARVEY, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2019 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Polish language edition published by Helion SA, Copyright © 2020.

Deitel and the double-thumbs-up bug are registered trademarks of Deitel and Associates, Inc.

Python logo courtesy of the Python Software Foundation.

Cover design by Paul Deitel, Harvey Deitel, and Chuti Prasertsith

Cover art by Agsandrew/Shutterstock

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/pytprs>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/pytprs.zip>

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Przedmowa	13
Zanim zaczniesz	33
1. Wprowadzenie do komputerów i języka Python	37
1.1. Wstęp	38
1.2. Podstawy technologii obiektowych	39
1.3. Język Python	41
1.4. Biblioteki	44
1.5. IPython i notatniki Jupytera — pierwsze spotkanie	47
1.6. Chmury i internet rzeczy	55
1.7. Big Data — ile to jest „Big”?	58
1.8. Analiza przypadku — mobilna aplikacja Big Data	66
1.9. Wprowadzenie do Data Science: gdy informatyka spotyka się z Data Science	68
1.10. Podsumowanie	71
2. Wprowadzenie do programowania w języku Python	73
2.1. Wstęp	74
2.2. Zmienne i instrukcje przypisania	74
2.3. Obliczenia arytmetyczne	75
2.4. Funkcja „print” i łańcuchy znaków	81
2.5. Łańcuchy z potrójnymi ogranicznikami	83
2.6. Wprowadzanie danych z klawiatury	84
2.7. Podejmowanie decyzji: instrukcja „if” i operatory porównania	86
2.8. Obiekty i typowanie dynamiczne	92
2.9. Wprowadzenie do Data Science — podstawowe statystyki opisowe	93
2.10. Podsumowanie	96
3. Instrukcje przepływu sterowania	97
3.1. Wstęp	98
3.2. Słowa kluczowe języka Python	99
3.3. Instrukcja „if”	99
3.4. Instrukcje „if ... else” i „if ... elif ... else”	101
3.5. Instrukcja „while”	104
3.6. Instrukcja „for”	104
3.7. Rozszerzone przypisania	107
3.8. Iterowanie po ciągach. Formatowane łańcuchy	107
3.9. Nadzorowane iterowanie	109
3.10. Wbudowana funkcja „range” — nieco dokładniej	112
3.11. Obliczenia finansowe — typ „Decimal”	112

3.12.	Instrukcje „continue” i „break”	116
3.13.	Operatory boolowskie	117
3.14.	Wprowadzenie do Data Science: miary tendencji centralnej	120
3.15.	Podsumowanie	122
4.	Funkcje	123
4.1.	Wstęp	124
4.2.	Funkcje definiowane w programie	124
4.3.	Funkcje z wieloma parametrami	127
4.4.	Generowanie liczb pseudolosowych	129
4.5.	Analiza przypadku — gra losowa	132
4.6.	Standardowa biblioteka Pythona	135
4.7.	Funkcje modułu „math”	136
4.8.	Wspomagane uzupełnianie kodu	138
4.9.	Domyślne wartości parametrów	140
4.10.	Argumenty kluczowe	141
4.11.	Zmienne listy parametrów	142
4.12.	Metody — funkcje należące do obiektów	144
4.13.	Zasięg definicji	145
4.14.	O importowaniu nieco dokładniej	147
4.15.	Przekazywanie argumentów — nieco szczegółów	149
4.16.	Rekurencja	152
4.17.	Funkcyjny styl programowania	156
4.18.	Wprowadzenie do Data Science: miary rozproszenia	158
4.19.	Podsumowanie	160
5.	Ciągi: listy i krotki	161
5.1.	Wstęp	162
5.2.	Listy	162
5.3.	Krotki	167
5.4.	Rozpakowywanie ciągów	170
5.5.	Wyodrębnianie podciągów	173
5.6.	Instrukcja „del”	176
5.7.	Listy jako argumenty wywołań funkcji	177
5.8.	Sortowanie list	179
5.9.	Multiplikacja ciągu	180
5.10.	Przeszukiwanie ciągów	180
5.11.	Inne metody listy	183
5.12.	Symulowanie stosu za pomocą listy	186
5.13.	Odwzorowywanie list	187
5.14.	Wyrażenia generatorowe	189
5.15.	Natywne filtrowanie, mapowanie i redukcja	189
5.16.	Inne funkcje do przetwarzania ciągów	192
5.17.	Listy dwuwymiarowe	193
5.18.	Wprowadzenie do Data Science: symulacje i ich statyczna wizualizacja	195
5.19.	Podsumowanie	205

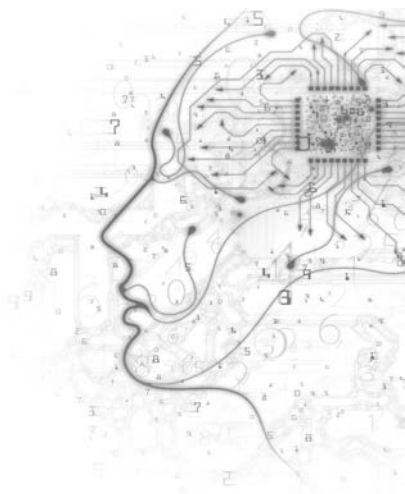
6.	Słowniki i zbiory	207
6.1.	Wstęp	208
6.2.	Słowniki	208
6.3.	Zbiory	218
6.4.	Wprowadzenie do Data Science: dynamiczna wizualizacja symulacji	224
6.5.	Podsumowanie	232
7.	Biblioteka NumPy i tablice ndarray	233
7.1.	Wstęp	234
7.2.	Tworzenie tablic na podstawie istniejących danych	235
7.3.	Atrybuty tablic	235
7.4.	Wypełnianie tablicy zadaną wartością	237
7.5.	Tworzenie tablicy na podstawie zakresu danych	238
7.6.	%timeit — porównanie efektywności tablic i list	240
7.7.	Inne „magiczne” polecenia IPython	241
7.8.	Operatory tablicowe	242
7.9.	Metody obliczeniowe biblioteki „NumPy”	244
7.10.	Funkcje uniwersalne biblioteki „NumPy”	245
7.11.	Indeksowanie i wyodrębnianie	247
7.12.	Widoki tablic jako płytkie kopie	249
7.13.	Głębokie kopiowanie	250
7.14.	Restrukturyzacja i transponowanie tablic	251
7.15.	Wprowadzenie do Data Science: szeregi i ramki danych biblioteki Pandas	255
7.16.	Podsumowanie	269
8.	Łańcuchy znaków	271
8.1.	Wstęp	272
8.2.	Formatowanie łańcuchów	272
8.3.	Konkatenowanie i zwielokrotnianie łańcuchów	278
8.4.	Usuwanie białych znaków otaczających łańcuch	278
8.5.	Zmiana wielkości liter w łańcuchu	279
8.6.	Operatory porównywania łańcuchów	279
8.7.	Wyszukiwanie podłańcuchów	280
8.8.	Zastępowanie podłańcuchów	283
8.9.	Dzielenie i składanie łańcuchów	283
8.10.	Testowanie specyficznych właściwości łańcucha i jego znaków	286
8.11.	Surowe łańcuchy	287
8.12.	Podstawy wyrażeń regularnych	288
8.13.	Wprowadzenie do Data Science: wyrażenia regularne i preparacja danych w bibliotece Pandas	298
8.14.	Podsumowanie	303
9.	Pliki i wyjątki	305
9.1.	Wstęp	306
9.2.	Pliki	307
9.3.	Przetwarzanie plików tekstowych	308

9.4.	Aktualizowanie plików tekstowych	311
9.5.	Serializacja obiektów w formacie JSON	313
9.6.	Niebezpieczny moduł „pickle”	315
9.7.	Dodatkowe uwagi o plikach	316
9.8.	Obsługa wyjątków	318
9.9.	Klauzula „finally”	323
9.10.	Jawne generowanie wyjątków	325
9.11.	Odwijanie stosu i ślad wykonania	326
9.12.	Wprowadzenie do Data Science: przetwarzanie plików CSV	328
9.13.	Podsumowanie	335
10.	Programowanie zorientowane obiektowo	337
10.1.	Wstęp	338
10.2.	Przykład: klasa „Account”	340
10.3.	Kontrolowanie dostępu do atrybutów	344
10.4.	Właściwości organizują dostęp do atrybutów. Przykład: klasa „Time”	345
10.5.	Symulowanie „prywatności” atrybutów	353
10.6.	Analiza przypadku: symulacja tasowania i rozdawania kart	354
10.7.	Dziedziczenie: klasy bazowe i podklasy	364
10.8.	Hierarchia dziedziczenia a polimorfizm	366
10.9.	„Kacze typowanie” a polimorfizm	374
10.10.	Przeciążanie operatorów	375
10.11.	Klasy wyjątków — hierarchia i definiowanie podklas	379
10.12.	Nazwane krotki	380
10.13.	Nowość wersji 3.7: klasy danych	381
10.14.	Testy jednostkowe przy użyciu łańcuchów dokumentacyjnych i modułu „doctest”	388
10.15.	Przestrzenie nazw i widoczność identyfikatorów	391
10.16.	Wprowadzenie do Data Science: szeregi czasowe i prosta regresja liniowa	394
10.17.	Podsumowanie	402
11.	Przetwarzanie języka naturalnego (NLP)	405
11.1.	Wstęp	406
11.2.	Klasa „TextBlob”	407
11.3.	Wizualizacja statystyki słów	422
11.4.	Ocena czytelności tekstu — biblioteka „Textatistic”	428
11.5.	Rozpoznawanie nazwanych encji — biblioteka „spaCy”	430
11.6.	Podobieństwo dokumentów	431
11.7.	Inne biblioteki i narzędzia NLP	432
11.8.	Zastosowanie NLP w uczeniu maszynowym i głębokim uczeniu	433
11.9.	Zbiory danych dla NLP	434
11.10.	Podsumowanie	434

12. Eksploracja masowych danych — Twitter	437
12.1. Wstęp	438
12.2. Ogólnie o API Twittera	440
12.3. Pierwszy krok — konto deweloperskie	441
12.4. Drugi krok — aplikacja i poświadczenia	441
12.5. Tweety jako obiekty	443
12.6. Biblioteka „Tweepy”	447
12.7. Uwierzytelnianie za pomocą biblioteki „Tweepy”	447
12.8. Informacja o koncie Twittera	449
12.9. Kursory biblioteki „Tweepy”	451
12.10. Przeszukiwanie niedawnych tweetów	454
12.11. Odkrywanie trendów — Twitter Trends API	458
12.12. Preparacja tweetów przed analizą	462
12.13. Strumieniowanie tweetów — Twitter Streaming API	464
12.14. Tweety i analiza sentymentu — klasa „SentimentListener”	469
12.15. Mapy i geokodowanie	474
12.16. Przechowywanie tweetów	483
12.17. Twitter a szeregi czasowe	483
12.18. Podsumowanie	484
13. Przetwarzanie poznawcze — IBM Watson	485
13.1. Wstęp	486
13.2. Konto IBM Cloud i konsola usług	487
13.3. Usługi platformy Watson	488
13.4. Przydatne narzędzia platformy Watson	492
13.5. Watson Developer Cloud Python SDK	493
13.6. Analiza przypadku: dwujęzyczna aplikacja konwersacyjna	494
13.7. Zasoby powiązane z IBM Watson	509
13.8. Podsumowanie	511
14. Uczenie maszynowe: klasyfikacja, regresja i klasteryzacja	513
14.1. Wstęp	514
14.2. Analiza przypadku: algorytm k najbliższych sąsiadów — klasyfikacja w zbiorze „Digits”. Część pierwsza	520
14.3. Analiza przypadku: algorytm k najbliższych sąsiadów — klasyfikacja w zbiorze „Digits”. Część druga	531
14.4. Analiza przypadku: prosta regresja liniowa na szeregu czasowym	539
14.5. Analiza przypadku: wielokrotna regresja liniowa na zbiorze „California Housing”	545
14.6. Analiza przypadku: uczenie nienadzorowane. Część pierwsza — redukcja wymiarowości	558
14.7. Analiza przypadku: uczenie nienadzorowane. Część druga — klasteryzacja za pomocą algorytmu k średnich	562
14.8. Podsumowanie	576

15. Głębokie uczenie	579
15.1. Wstęp	580
15.2. Pliki danych wbudowane w bibliotekę „Keras”	583
15.3. Alternatywne środowiska Anacondy	584
15.4. Sieci neuronowe	586
15.5. Tensory	588
15.6. Konwolucyjne sieci neuronowe i widzenie komputerowe: wieloklasyfikacja w zbiorze „MNIST”	590
15.7. TensorBoard — wizualizacja trenowania sieci	610
15.8. ConvNetJS: wizualizacja trenowania w oknie przeglądarki WWW	613
15.9. Rekurencyjne sieci neuronowe i ciągi danych: analiza sentymentu w zbiorze „IMDb”	614
15.10. Dostrajanie modeli głębokiego uczenia	622
15.11. Modele wstępnie wytrenowane	623
15.12. Podsumowanie	624
16. Big Data: Hadoop, Spark, NoSQL i IoT	627
16.1. Wstęp	628
16.2. Relacyjne bazy danych i język SQL	632
16.3. Bazy danych NoSQL i NewSQL dla Big Data — krótki rekonesans	644
16.4. Analiza przypadku: dokumenty JSON w bazie MongoDB	648
16.5. Hadoop	660
16.6. Spark	672
16.7. Strumieniowanie Sparka: zliczanie hashtagów przy użyciu „pyspark-notebook”	683
16.8. Internet rzeczy (IoT) i dashboardy	693
16.9. Podsumowanie	707

Przetwarzanie poznawcze — IBM Watson



W tym rozdziale opisujemy:

- repertuar usług platformy IBM Watson i możliwości ich darmowego użytkowania w ramach warstwy *Lite tier*,
- przykłady wykorzystywania wspomnianych usług,
- istotę przetwarzania poznawczego (*cognitive computing*) i możliwości wykorzystywania go w tworzonych aplikacjach,
- zakładanie konta IBM Cloud i uzyskiwanie dostępu do poszczególnych usług platformy IBM Watson,
- instalowanie pakietu *Watson Developer Cloud Python SDK*, umożliwiającego wykorzystywanie usług platformy IBM Watson w aplikacjach języka Python,
- budowę aplikacji-mashupu wykorzystującej trzy usługi platformy IBM Watson,
- dodatkowe zasoby związane z platformą IBM Watson, między innymi IBM Redbooks.

13.1. Wstęp	13.5. Watson Developer Cloud Python SDK
13.2. Konto IBM Cloud i konsola usług	13.6. Analiza przypadku: dwujęzyczna aplikacja konwersacyjna
13.3. Usługi platformy Watson	13.7. Zasoby powiązane z IBM Watson
13.4. Przydatne narzędzia platformy Watson	13.8. Podsumowanie

13.1. Wstęp

W rozdziale 1. opisywaliśmy kilka spektakularnych osiągnięć na polu sztucznej inteligencji, między innymi zwycięstwo IBM-owskiego superkomputera Watson w potyczce o milion dolarów z dwoma arcymistrzami teleturnieju Jeopardy! (firma IBM przeznaczyła wygraną na cele charytatywne). Zdolność Watsona do wykonywania równoległe kilkuset algorytmów przetwarzania języka naturalnego okazała się wystarczająca do odnajdywania właściwych odpowiedzi w repozytorium o rozmiarze 4 terabajtów, zawierającym ponad 200 milionów stron wiedzy, między innymi całą Wikipedię¹. Ten spektakularny sukces to zasługa badaczy z IBM, którzy wyszkolili Watsona, używając technik uczenia maszynowego (poświęcamy mu następny rozdział) oraz uczenia ze wzmocnieniem (*reinforcement-learning*)³.

Przygotowując niniejszą książkę i prowadząc związane z tym rozeznanie tematu, nie mogliśmy nie zainteresować się coraz większą popularnością superkomputera Watson; zapuściliśmy odpowiednią sondę w Google Alerts i otrzymaliśmy w rezultacie ponad 900 różnych artykułów, cytatów z dokumentacji i blogów, filmów i fotografii. Równocześnie postanowiliśmy przyjrzeć się nico dokładniej konkurencyjnym usługom, w porównaniu z którymi Watson okazał się (nomen omen) bezkonkurencyjny, jako jedna z najbardziej przyjaznych usług, z darmową warstwą *Lite tier*⁴, dzięki której wszyscy fascynujący się osiągnięciami sztucznej inteligencji mogą poeksperymentować z Watsonem bez wnoszenia jakichkolwiek opłat („no credit card required”).

IBM Watson to oparta na chmurze platforma **poznawczego przetwarzania danych** (*cognitive computing*), popularna w wielu zastosowaniach do szerokiego spektrum rozmaitych scenariuszy z rzeczywistego świata. Systemy jej przetwarzania poznawczego okazują się coraz bardziej sprawne w takich zadaniach, jak rozpoznawanie wzorców czy symulowanie ludzkiego mózgu w procesach decyzyjnych, w miarę jak konsumują coraz większe wolumeny danych treningowych^{5,6,7}. W tabeli 13.1 przedstawiamy arbitralnie wybraną listę obszarów, w jakich wiele firm, instytucji, organizacji i agencji z powodzeniem spożytkowuje możliwości Watsona.

Watson oferuje mnóstwo intrygujących możliwości, które programiści wykorzystywać mogą w tworzonych przez siebie aplikacjach. Gdy założysz konto IBM Cloud⁸, czytając ten rozdział, będziesz mógł na bieżąco implementować opisywane elementy aplikacji, na bazie darmowego

¹ <https://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>

² [https://pl.wikipedia.org/wiki/Watson_\(superkomputer\)](https://pl.wikipedia.org/wiki/Watson_(superkomputer))

³ <https://www.aaai.org/Magazine/Watson/watson.php>, AI Magazine, jesień 2010.

⁴ Polityka udostępniania usług ma to do siebie, że może zmieniać się z biegiem czasu, warto więc regularnie czytać jej postanowienia na stronach IBM.

⁵ <http://whatis.techtarget.com/definition/cognitive-computing>

⁶ https://en.wikipedia.org/wiki/Cognitive_computing

⁷ <https://www.forbes.com/sites/bernardmarr/2016/03/23/what-everyone-should-know-about-cognitive-computing>

⁸ Serwis *IBM Cloud* nosił poprzednio nazwę *bluemix*, która figuruje w wielu publikacjach, szczególnie w URL-ach; nie jest to problem, ponieważ użytkownicy korzystający z tych URL-i automatycznie przekierowywani są na nowe strony.

TABELA 13.1. Wybrane obszary zastosowań platformy IBM Watson

Analiza sentymentu, badanie opinii publicznej	Ochrona zdrowia	Rzeczywistość rozszerzona
Asysta personalna	Opracowywanie nowych leków	Rzeczywistość wirtualna
Bezpieczeństwo w miejscu pracy	Przydatne utrzymanie ruchu (PdM)	Samochody autonomiczne
Chatboty	Prognozowanie pogody	Sport
Diagnostyka medyczna i terapia	Przetwarzanie języka naturalnego	Sztuczna inteligencja
Diagnostyka obrazowa w medycynie	Przetwarzanie obrazów	Tłumaczenie między językami naturalnymi
Edukacja	Przetwarzanie poznawcze	Uczenie maszynowe
Generowanie napisów do filmów	Rekomendowanie produktów	Wsparcie dla klientów
Genetyka	Roboty i drony	Wykrywanie cybernetycznych aktów przemocy
Identyfikacja grupy docelowej odbiorców reklamy	Rozgrywanie gier	Wykrywanie nadużyć finansowych
Inteligentne domy	Rozpoznawanie i analiza głosu	Wykrywanie zagrożeń
Interfejsy konwersacyjne	Rozpoznawanie obiektów	Wykrywanie złośliwego oprogramowania
Internet rzeczy (IoT)	Rozpoznawanie twarzy	Zapobieganie przestępstwom
Muzyka	Rozszerzona inteligencja	Zarządzanie łańcuchem dostaw
	Rozumienie języka naturalnego	

dostępu (*Lite tier*) do kilku usług realizujących konwersję mowy na tekst pisany, syntetyzowanie mowy na podstawie tekstu pisanego i tłumaczenia tekstów między językami naturalnymi. Opiszemy ponadto kilka innych usług, związanych z próbami rozumienia języka naturalnego, chatbotami, analizowaniem wydźwięku emocjonalnego tekstu oraz rozpoznawaniem obrazów i materiałów wideo.

Do tworzenia aplikacji w języku Python, wykorzystujących usługi platformy Watson, będziesz musiał zainstalować pakiet *Watson Developer Cloud Python Software Development Kit* (SDK). Potem będziesz mógł na bieżąco implementować kolejne fragmenty aplikacji, umożliwiające konwersację dwojga rozmówców posługujących się różnymi językami, dzięki konwersji między mową i tekstem pisany oraz tłumaczeniu tekstu między językami angielskim i hiszpańskim.

IBM Watson to platforma wciąż ewoluująca i dynamicznie rozwijająca się. W czasie, gdy powstawała ta książka, dodano do tej platformy wiele usług, kilka zmodyfikowano, kilka też usunięto. Może się więc zdarzyć, że niektóre treści prezentowane w tym rozdziale okażą się nieadekwatne do aktualnego stanu platformy, choć były do niego adekwatne w momencie ich powstawania. W miarę możliwości będziemy zamieszczać na naszej stronie www.deitel.com informacje na temat istotniejszych zmian tego rodzaju.

13.2. Konto IBM Cloud i konsola usług

By uzyskać dostęp do darmowych usług *Lite tier* Watsona, musisz założyć konto IBM Cloud. Na stronach WWW zawierających opisy poszczególnych usług znajdziesz szczegółowe opisy korzystania z nich oraz informację o ograniczeniach wynikających z darmowego trybu tego korzystania. Choć usługi warstwy *Lite tier* charakteryzują się wieloma ograniczeniami, są wystarczające zarówno do celów edukacyjnych — czyli poznawania możliwości platformy — jak i do praktycznych zastosowań w aplikacjach. Wspomniane ograniczenia są specyficzne dla poszczególnych usług, mogą się zmieniać z upływem czasu, nie próbujemy więc ich tu wymieniać, odsyłając zainteresowanych Czytelników do stron WWW z opisami usług. Niestety, w trakcie pisania tej książki zaobserwowaliśmy, że w przypadku wielu usług ograniczenia te stawały się coraz bardziej restrykcyjne. Oczywiście na potrzeby tworzenia komercyjnych aplikacji IBM oferuje szeroką gamę płatnego dostępu do swych usług.

Rozpocznij procedurę zakładania konta, wchodząc na stronę

<https://cloud.ibm.com/docs/services/watson?topic=watson-about#about>

i postępuj zgodnie z zamieszczonymi tam instrukcjami. Kliknij link *Sign up for free*, wprowadź swój adres e-mail, wypełnij formularz rejestracyjny i kliknij przycisk *Create Account*. Kiedy to zrobisz, otrzymasz e-mail z linkiem do aktywowania konta; postępuj zgodnie z zawartymi w nim wskazówkami.

Po aktywowaniu konta możesz zalogować się do konsoli IBM Cloud, po czym na stronie **Watson dashboard** pod adresem

<https://cloud.ibm.com/developer/watson/dashboard>

możesz

- obejrzeć listę usług Watsona,
- połączyć się z usługami, w których się zarejestrowałeś,
- skorzystać z zasobów deweloperskich, między innymi dokumentacji Watsona, pakietów SDK (*Software Development Kits*) i różnych zasobów wspomagających naukę korzystania z platformy,
- przeglądać aplikacje, które stworzyłeś z udziałem platformy Watson.

W następnej kolejności możesz rejestrować się w poszczególnych usługach Watsona, używając dane uwierzytelniające warunkujące dostęp do nich; możesz przeglądać te dane i zarządzać nimi, wchodząc na stronę

<https://cloud.ibm.com/resources>

i rozwijając pozycję *Services* z lewej strony ekranu.

13.3. Usługi platformy Watson

W tej sekcji opisujemy podstawy kilku usług platformy Watson, wraz z odsyłaczami do szczegółowych informacji na ich temat. Na stronie

<https://cloud.ibm.com/developer/watson/documentation>

znajduje się wyszczególnienie dostępnych usług, wraz z odsyłaczami do ich dokumentacji i interfejsów API. Odsyłacze te reprodukujemy w przypisach towarzyszących poszczególnym sekcjom. Klikając link *Documentation* w sekcji danej usługi, zostaniesz przekierowany na jej stronę dokumentacyjną, zawierającą link do strony szczegółowej. Jeżeli chcesz uruchomić daną usługę, kliknij na tej stronie przycisk *Create*.

13.3.1. Usługa „Watson Assistant”

Usługa **Watson Assistant**⁹ jest podstawą konstrukcji wirtualnych asystentów i chatbotów¹⁰, z którymi użytkownik porozumiewa się za pomocą tekstu w języku naturalnym. IBM dostarcza interfejs webowy, umożliwiający *trenowanie* tej usługi pod kątem specyficznego scenariusza wynikającego z przeznaczenia aplikacji — na przykład odpowiadania na pytania w rodzaju „Czy jutro w Warszawie będzie padać?” lub inteligentnej obsługi użytkownika firmowego serwisu w wirtualnym biurze obsługi klienta. Kilka przykładowych interakcji tego rodzaju dostępnych jest pod adresem

<https://www.ibm.com/watson/services/conversation/demo/index.html#demo>

⁹ <https://cloud.ibm.com/catalog/services/watson-assistant>

¹⁰ <https://pl.wikipedia.org/wiki/Chatbot>

13.3.2. Usługa „Visual Recognition”

Usługa **Visual Recognition**¹¹ umożliwia lokalizowanie i interpretowanie informacji zawartych w obrazach i filmach: kolorów, obiektów, twarzy, tekstów, żywności, niewłaściwych treści itp. IBM dostarcza serię predefiniowanych modeli (dostępnych w demo usługi), możliwe jest też samodzielne trenowanie usługi (czego przykład przedstawimy w następnym rozdziale). Wchodząc na stronę

<https://watson-visual-recognition-duo-dev.ng.bluemix.net/>

możesz wypróbować możliwości usługi w odniesieniu do załączonych obrazów (oraz Twoich własnych).

13.3.3. Usługa „Speech to Text”

Zadaniem usługi **Speech to Text**¹² jest rozpoznawanie mowy zapisanej w pliku audio i jej konwersja na tekst pisany; wykorzystamy ją w przykładowej aplikacji. Usługa zostaje zaopatrzona w zestaw słów kluczowych, które powinna „wysłyszeć” w mówionym materiale i w przypadku rozpoznania któregoś z nich zasygnalizować ten fakt, podając miejsce jego wystąpienia (offset w pliku) oraz własną ocenę prawdopodobieństwa, że rozpoznanie to jest trafne. Usługa potrafi ponadto rozróżniać głos poszczególnych autorów wypowiedzi. Jest więc oczywiste, że idealnie nadaje się na potrzeby budowania wszelkich aplikacji sterowanych głosem oraz automatycznego rejestrowania przemówień w postaci tekstowej. Jej demo znajduje się na stronie

<https://speech-to-text-demo.ng.bluemix.net/>

gdzie możesz poeksperymentować z załączonymi plikami audio i własnymi plikami dźwiękowymi.

13.3.4. Usługa „Text to Speech”

Mechanizmem komplementarnym do rozpoznawanie mowy jest syntetyzowanie głosu na podstawie tekstu pisanego, implementowane przez automatycznego lektora — usługa **Text to Speech**¹³; ją także wykorzystamy w tym rozdziale. Tekst interpretowany przez usługę można wzbogacić o **znaczniki syntezy mowy**, składające się na język **SSML** (*Speech Synthesis Markup Language*), dostarczające wskazówek dotyczących (między innymi) fleksji, intonacji, rytmu i barwy głosu. Obecnie usługa ta zapewnia obsługę języków: angielskiego (w wydaniu brytyjskim i amerykańskim), francuskiego, niemieckiego, włoskiego, hiszpańskiego, portugalskiego i japońskiego. Możesz zaobserwować jej działanie dla plików tekstowych (również swoich własnych), uruchamiając jej demo dostępne na stronie

<https://text-to-speech-demo.ng.bluemix.net/>

13.3.5. Usługa „Language Translator”

Usługa **Language Translator**¹⁴ — ją również wykorzystamy w tym rozdziale — to automatyczny tłumacz tekstu pisanego, spełniający dwie zasadnicze funkcje:

- tłumaczenie tekstu pisanego ze wskazanego języka źródłowego na wskazany język docelowy,
- automatyczne rozpoznawanie języka tekstu źródłowego wśród ponad 60 języków.

¹¹ <https://cloud.ibm.com/catalog/services/visual-recognition>

¹² <https://cloud.ibm.com/catalog/services/speech-to-text>

¹³ <https://cloud.ibm.com/catalog/services/text-to-speech>

¹⁴ <https://cloud.ibm.com/catalog/services/language-translator>

Demo usługi dostępne jest na stronie

<https://language-translator-demo.ng.bluemix.net/>

13.3.6. Usługa „Natural Language Understanding”

Próba rozumienia tekstu w języku naturalnym to istota usługi **Natural Language Understanding**¹⁵ — pod pojęciem „rozumienia” kryją się głównie dwie rzeczy: całościowa ocena sentymentu i emocji tekstu oraz ocena istotności (relevantności) występowania wskazanych słów kluczowych. Usługa podejmuje próbę identyfikowania (między innymi):

- osób, miejsc, tytułów zawodowych i naukowych, organizacji, firm i wielkości fizycznych;
- kategorii, dziedzin i koncepcji — sztuki, sportu, organów władzy, polityki;
- części mowy i części zdania — rzeczowników, czasowników, podmiotów i orzeczeń.

Ponadto za pomocą *Watson Knowledge Studio*, można wyszkolić usługę do działania w bardziej zaawansowanych dziedzinach wiedzy i działalności człowieka (niebawem powrócimy do tej kwestii). Możesz sprawdzić działanie usługi w odniesieniu do różnych tekstów (gotowych i swoich własnych), uruchamiające jej demo dostępne na stronie

<https://natural-language-understanding-demo.ng.bluemix.net/>

13.3.7. Usługa „Discovery”

Usługa **Watson Discovery**¹⁶ posiada wiele cech wspólnych z usługą *Natural Language Understanding*, dodatkowo umożliwiając przedsiębiorstwom zarządzanie dużymi kolekcjami gromadzonych dokumentów, na podstawie ich treści wyrażonej w językach naturalnych. Demo usługi dostępne jest na stronie

<https://discovery-news-demo.ng.bluemix.net/>

13.3.8. Usługa „Personality Insights”

Na podstawie analizy tekstu pisanego można wiele wywnioskować na temat cech osobowości jego autora, wykorzystując wiedzę psychologiczną oraz opisywane w rozdziale 11. techniki przetwarzania języka naturalnego. Z tym ambitnym wyzwaniem postanowili zmierzyć się twórcy usługi **Personality Insight**¹⁷, która w założeniu umożliwić ma (tu cytuję z dokumentacji) „wgląd w sposób myślenia ludzi, motywacje i sposoby ich działania oraz odczuwania rzeczywistości — to wszystko z wykorzystaniem wiedzy z teorii osobowości oraz analiz lingwistycznych zmierzających do wnioskowania cech osoby na podstawie swobodnego tekstu jej autorstwa”. Wnioski ze wspomnianej analizy mają duże znaczenie praktyczne, na przykład w marketingu, pozwalają bowiem kierować reklamę danego produktu głównie do osób, które z dużym prawdopodobieństwem okażą się jego nabywcami. Na stronie

<https://personality-insights-livedemo.ng.bluemix.net/>

dostępne jest demo usługi, dzięki któremu możesz ją przetestować w odniesieniu do kolekcji tweetów pochodzących z różnych kont Twittera (także z Twojego, jeśli takowe posiadasz) oraz do innych dokumentów.

¹⁵ <https://cloud.ibm.com/catalog/services/natural-language-understanding>

¹⁶ <https://cloud.ibm.com/catalog/services/discovery>

¹⁷ <https://cloud.ibm.com/catalog/services/personality-insights>

13.3.9. Usługa „Tone Analyzer”

Usługa **Tone Analyzer**¹⁸ dokonuje oceny ogólnego charakteru tekstu pisanego, analizując go pod kątem trzech następujących elementów:

- emocji — złości, zniechęcenia, strachu, radości, smutku;
- nastawienia społecznego — otwartości, szczerości, ekstrawersji, sympatii i zabarwienia emocjonalnego;
- stylu języka — analitycznego, zdecydowanego, potocznego, niepewnego.

Demo usługi, dostępne pod adresem

<https://tone-analyzer-demo.ng.bluemix.net/>

możesz wypróbować na kolekcji tweetów, recenzji produktów, e-mailach i ewentualnie Twoich własnych plikach.

13.3.10. Usługa „Natural Language Classifier”

Usługa **Natural Language Classifier**¹⁹ to implementacja pojętnego ucznia, który po odpowiednim wyszkoleniu, za pomocą zdań i fraz z określonej dziedziny, dokonuje klasyfikacji podanego tekstu w odpowiedniej kategorii. Na przykład chciałbyś, aby zapytanie Twojego klienta „Na jakich warunkach mogę wydzierżawić produkt XXX” trafiło do kategorii „Pomoc dla klienta”, a zastrzeżenie „Na mojej fakturze jest błąd” do kategorii „Rozliczenia”. Wyszkolona usługa, wykorzystując mechanizmy przetwarzania poznawczego platformy Watson, próbuje zaklasyfikować otrzymane zdanie lub frazę w najbardziej prawdopodobnych kategoriach, podając przy tym ocenę prawdopodobieństwa każdej z propozycji. Usługa ta może oczywiście stanowić jeden z wielu etapów funkcjonalności aplikacji, na przykład nagrana wypowiedź klienta telefonującego na infolinię może zostać zapisana w postaci tekstu (za pomocą usługi *Speech to Text*), który przekazany zostanie do kwalifikacji zmierzającej do skierowania go do właściwego pracownika, oddziału czy wydziału firmy.

Niestety, *ta usługa nie oferuje żadnej możliwości darmowego dostępu*, dostępne jest jedynie jej demo na stronie

<https://natural-language-classifier-demo.ng.bluemix.net/>

gdzie możesz zadać pytanie dotyczące pogody, a klasyfikator rozstrzygnie, czy dotyczy ono temperatury, czy innych warunków meteorologicznych.

13.3.11. Usługa „Machine Learning”

Usługa **Watson Machine Learning**²⁰ umożliwia wyposażanie tworzonych aplikacji w mechanizmy predykcyjne, przez wykorzystywanie popularnych frameworków dedykowanych uczeniu maszynowemu, między innymi *Tensorflow*, *Keras* i *scikit-learn*. Dwa ostatnie z wymienionych wykorzystywać będziemy w następnych dwóch rozdziałach.

¹⁸ <https://cloud.ibm.com/catalog/services/tone-analyzer>

¹⁹ <https://cloud.ibm.com/catalog/services/natural-language-classifier>

²⁰ <https://cloud.ibm.com/catalog/services/machine-learning>

13.3.12. Usługa „Cognos Analytics”

Usługa **Cognos Analytics**²¹, dostępna za darmo przez 30 dni okresu próbnego, wykorzystuje sztuczną inteligencję i uczenie maszynowe do wykrywania i wizualizacji informacji zawartej w rozmaitych danych, *bez jakichkolwiek zabiegów programistycznych ze strony użytkownika*. Usługa oferuje także odpowiadanie na pytania zadawane w języku naturalnym, dotyczące wiedzy, jaką zebrała w procesie analizowania danych.

13.3.13. Synchroniczne i asynchroniczne wywołania usług

Wiele funkcji i usług wykorzystywanych w niniejszej książce wykonuje swą pracę w trybie **synchronicznym** — program dokonuje wywołania procedury lub usługi i oczekuje, aż ta zakończy swą pracę i zwróci sterowanie. Alternatywą jest wywołanie **asynchroniczne**: program inicjuje jedynie działanie usługi lub funkcji i powraca do swych głównych zadań; wywołana usługa lub funkcja działa równolegle z nim, a gdy zakończy pracę, program główny zostaje o tym *powiadomiony*. Większość usług Watsona może być uruchamianych w każdym z dwu wymienionych trybów.

Znakomitym przykładem asynchronicznego wywołania usługi jest demo usługi *Speech to Text*. Proces demo dokonuje próbkowania („sąplowania”) konwersacji dwojga (lub więcej) rozmówców, usługa na bieżąco tworzy transkrypcję tekstową tej konwersacji, a proces główny wypisuje ją — początkowo bez podziału na rozmówców, bo usługa jeszcze nie jest w stanie ich rozróżnić; wypisywany tekst poprzedzony jest komunikatem (Detecting speakers):. W pewnym momencie usługa dokonuje identyfikacji rozmówców i tekst wyprodukowany dotychczas przez usługę zostaje zwrócony ponownie, tym razem z podziałem na rozmówców — i w takim właśnie podziale wypisywane są odtąd wyniki transkrypcji.

We współczesnym świecie procesorów wielordzeniowych i klasterów wieloprocesorowych przetwarzanie zrównoleglone, czyli między innymi asynchroniczne wywołania usług, to rzecz naturalna, przyczyniająca się do poprawy efektywności aplikacji. Niestety, ten medal ma dwie strony: programowanie z wywołaniami asynchronicznymi jest bardziej skomplikowane i bardziej podatne na błędy niż programowanie z użyciem wyłącznie wywołań synchronicznych. Gdy w dalszej części rozdziału opisywać będziemy instalowanie pakietu *Watson Developer Cloud Python SDK*, zamieścimy link do repozytorium przykładowego kodu na GitHub, prezentującego zarówno synchroniczne, jak i asynchroniczne wersje różnych usług. Niezbędne szczegóły związane z poszczególnymi usługami dostępne są w ich *API Reference*.

13.4. Przydatne narzędzia platformy Watson

Poniżej opisujemy zaawansowane narzędzia platformy Watson, przydatne każdemu programiście.

13.4.1. Watson Studio

Watson Studio²² to nowy interfejs platformy Watson wspomagający tworzenie projektów z udziałem tej platformy i zarządzanie nimi oraz pracę zespołową nad tymi projektami. Dzięki temu możesz tworzyć dane, przygotowywać je do analizy, tworzyć notatniki Jupytera (patrz sekcja 1.5.3) dla interakcji z tymi danymi, tworzyć modele i trenować je, a także wykorzystywać oferowane przez Watsona możliwości w zakresie głębokiego uczenia.

Watson oferuje darmową wersję (*Lite tier*) Studio — *Watson Studio Lite*. W celu uzyskania dostępu do niej wejdź na stronę

²¹ <https://www.ibm.com/pl-pl/products/cognos-analytics>

²² <https://cloud.ibm.com/catalog/services/data-science-experience>

<https://cloud.ibm.com/catalog/services/data-science-experience>

i kliknij przycisk *Create*. Na nowo wyświetlonej stronie kliknij przycisk *Get Started*, a po wyświetleniu potwierdzenia utworzenia konta (okno *Done!*) zamknij je, klikając przycisk *Get Started*. Zatwierdź komunikat powitalny, klikając przycisk *Start tour*, zatwierdź wyświetlane po drodze informacje pomocnicze i ostatecznie, po wyświetleniu strony

<https://eu-gb.dataplatform.cloud.ibm.com/home?context=analytics>

kliknij przycisk *New project*, a następnie napis *Create a project* w wyświetlonym okienku. Watson zaproponuje Ci do wyboru:

- utworzenie nowego, pustego projektu (*Create an empty project*),
- utworzenie projektu na podstawie próbki lub pliku (*Create a project from a sample or a file*).

13.4.2. Knowledge Studio

Różne usługi platformy Watson zdolne są do współdziałania z *predefiniowanymi* modelami, lecz jednocześnie oferują także obsługę z nowo tworzonymi modelami, trenowanymi pod kątem określonych zastosowań. **Watson Knowledge Studio**²³ ułatwia tworzenie i trenowanie takich modeli, także w warunkach pracy zespołowej. Wytrenowane modele są gotowe do użycia przez usługi Watsona.

13.4.3. Knowledge Catalog

Watson Knowledge Catalog^{24 25} to zaawansowane narzędzie klasy enterprise dedykowane bezpiecznemu zarządzaniu, wyszukiwaniu i współdzieleniu danych organizacji. Wśród oferowanych możliwości wymienić należy:

- scentralizowany dostęp do danych firmowych, zarówno lokalnych, jak i tych w chmurach, oraz modeli uczenia maszynowego;
- wsparcie dla **Watson Studio**, dzięki któremu użytkownicy mogą łatwo odnajdywać i pobierać dane do wykorzystywania w projektach uczenia maszynowego;
- politykę bezpieczeństwa zapewniającą, że dostęp do określonych danych mają wyłącznie osoby, które mieć go powinny;
- dostępność ponad 100 operacji czyszczenia i preparowania danych przed ich analizą.

13.5. Watson Developer Cloud Python SDK

Zainstalujemy teraz oprogramowanie niezbędne do pełnej implementacji analizy przypadku opisywanej w następnej sekcji. IBM udostępnia programistom pakiet **Watson Developer Cloud Python SDK** (*software development kit*); wchodzący w jego skład moduł o nazwie `watson_developer_cloud` zawiera implementacje klas niezbędnych do interakcji z usługami Watsona — interakcja ta sprowadza się do tworzenia obiektów tych klas i wywoływania ich metod.

W celu zainstalowania SDK²⁶ uruchom²⁷

²³ <https://cloud.ibm.com/catalog/services/knowledge-studio>

²⁴ <https://medium.com/ibm-watson/introducing-ibm-watson-knowledge-catalog-cf42c13032c1>

²⁵ <https://dataplatform.cloud.ibm.com/docs/content/wsj/catalog/overview-wkc.html>

²⁶ Szczegółowy opis procedury instalacyjnej i użyteczne wskazówki na wypadek napotkania problemów dostępne są na stronie <https://github.com/watson-developer-cloud/python-sdk> w sekcji *README.md*.

- w Windows: Anaconda Prompt *jako administrator*,
- w Linuksie: Terminal lub shell,
- w MacOS: Terminal

wydaj polecenie

```
pip install --upgrade watson-developer-cloud
```

i poczekaj na zakończenie instalacji.

13.5.1. Moduły niezbędne do nagrywania i odtwarzania mowy

Potrzebne będą dodatkowo dwie biblioteki, do nagrywania dźwięku²⁸ (PyAudio) i odtwarzania plików dźwiękowych (PyDub). Aby je zainstalować, wydaj kolejno polecenia

```
pip install pyaudio
```

oraz

```
pip install pydub
```

13.5.2. Przykłady zastosowań pakietu

Aby ułatwić programistom korzystanie z usług platformy Watson, firma IBM udostępnia pliki z przykładowym kodem ilustrującym korzystanie z usług tej platformy, za pośrednictwem klas pakietu *Watson Developer Cloud Python SDK*. Pliki te dostępne są w serwisie GitHub, pod adresem

<https://github.com/watson-developer-cloud/python-sdk/tree/master/examples>

13.6. Analiza przypadku: dwujęzyczna aplikacja konwersacyjna

Wyobraź sobie, że podróżujesz po kraju, którego mieszkańcy posługują się językiem hiszpańskim, ty jednak nie znasz tego języka, a musisz porozumieć się z osobą, która nie zna języka angielskiego. No i nie ma w pobliżu tłumacza — ale na szczęście mamy XXI wiek i inteligentne aplikacje, które coraz lepiej się spisują w tej roli: uruchamiasz taką aplikację na swoim smartfonie, wypowiadasz swoją kwestię po angielsku, a z głośnika płynie hiszpańska wersja Twojej wypowiedzi. Interlokutor kiwa głową ze zrozumieniem — i odpowiada po hiszpańsku, a wspomniana aplikacja na bieżąco tłumaczy jego wypowiedź na angielski.

Nie pisałibyśmy o tym, gdyby historia ta mogła zaistnieć jedynie w sferze „wyobraź sobie”; przeciwnie — to scenariusz nie tylko realny, ale i realizowany na co dzień przy użyciu usług platformy Watson — co niebawem udowodnimy, budując przykładową aplikację o takiej właśnie funkcjonalności.

Wykorzystamy w tym celu trzy takie usługi, które w odpowiedni sposób ze sobą połączymy; takie połączenie dwóch lub więcej bytów (nie tylko programów, ale na przykład fragmentów literackich czy muzycznych) w celu uzyskania nowych wartości opatrywane jest nazwą **mas-hup**²⁹. W naszej aplikacji wykorzystamy także kilka standardowych operacji plikowych, które opisywaliśmy w rozdziale 9.

²⁷ W Windows może być konieczne uprzednie zainstalowanie *Microsoft's C++ build tools* po pobraniu ze strony <https://visualstudio.microsoft.com/visual-cpp-build-tools/>

²⁸ Instalacja PyAudio może wymagać podjęcia dodatkowych działań. Wskazówki znajdziesz na stronie domowej tego pakietu: <http://people.csail.mit.edu/hubert/pyaudio/> — przyp. red.

²⁹ [https://pl.wikipedia.org/wiki/Mashup_\(informatyka\)](https://pl.wikipedia.org/wiki/Mashup_(informatyka))

13.6.1. Zanim zaczniesz

W naszej aplikacji korzystać będziemy z trzech darmowych (*Lite tier*) usług platformy Watson, a do tego potrzebne będą dane uwierzytelniające dla każdej z nich. Zakładamy oczywiście, że posiadasz konto IBM Cloud i możesz realizować opisywane poniżej scenariusze ich uzyskiwania. Po ich otrzymaniu — dla każdej z usług — należy je zapisać w odpowiednich pozycjach pliku *keys.py*, znajdującego się w folderze z przykładami do rozdziału 13. *Pamiętaj o tym, że są to Twoje osobiste dane i nie powinieneś ich nikomu udostępniać.*

W każdym z poniższych scenariuszy wygenerowane dane uwierzytelniające składać się będą z dwóch elementów: klucza API oraz URL-a związanego z usługą. Każdy z tych URL-i jest domyślnym dla SDK i nie ma potrzeby go zmieniać ani kopiować, natomiast klucz API należy skopiować do odpowiedniej pozycji pliku *keys.py*. Właściwa zawartość tego pliku jest niezbędna do funkcjonowania aplikacji, której budowanie opisywać będziemy szczegółowo w sekcji 13.6.3.

13.6.1.1. Rejestracja w usłudze „Speech to Text”

Pierwszą z wykorzystywanych przez nas usług jest *Speech to Text*. Wykonuje ona konwersję mowy w języku angielskim i hiszpańskim na tekst w tych samych językach. Do interakcji z tą usługą konieczne jest podanie nazwy użytkownika i hasła oraz klucza API. Aby uzyskać ten klucz:

1. **Utwórz instancję usługi.** Wejdź na stronę

<https://cloud.ibm.com/catalog/services/speech-to-text>

i kliknij przycisk *Create*. Spowoduje to automatyczne wygenerowanie klucza API oraz wyświetlenie krótkiego tutoriala korzystania z usługi.

2. **Pobierz klucz API.** Kliknij *Manage* na panelu z lewej strony, a w sekcji *Credentials* kliknij *Show Credentials*. Skopiuj do schowka widoczny klucz API i wklej go (jako łańcuch) na pozycję `speech_to_text_key` w pliku *keys.py*.

13.6.1.2. Rejestracja w usłudze „Text to Speech”

Druga z usług naszej aplikacji — *Text to Speech* — syntetyzuje mowę na podstawie tekstu w języku angielskim lub hiszpańskim. Ta usługa także wymaga nazwy użytkownika i hasła.

1. **Utwórz instancję usługi.** Wejdź na stronę

<https://cloud.ibm.com/catalog/services/text-to-speech>

i kliknij przycisk *Create*. Spowoduje to automatyczne wygenerowanie klucza API oraz wyświetlenie krótkiego tutoriala korzystania z usługi.

2. **Pobierz klucz API.** Kliknij *Manage* na panelu z lewej strony, a w sekcji *Credentials* kliknij *Show Credentials*. Skopiuj do schowka widoczny klucz API i wklej go (jako łańcuch) na pozycję `text_to_speech_key` w pliku *keys.py*

13.6.1.3. Rejestracja w usłudze „Language Translator”

Ostatnia z wykorzystywanych przez nas usług — *Language Translator* — dokonuje tłumaczenia tekstu wyprodukowanego przez usługę *Speech to Text* na tekst, który interpretowany będzie przez usługę *Text to Speech*. To w ramach tej usługi dokonuje się zasadnicze niwelowanie fatalnych konsekwencji zbyt ambitnego projektu architektonicznego, jakim w starożytności okazała się Wieża Babel³⁰. Dostęp do usługi wymaga klucza API, który możesz otrzymać następująco:

³⁰ https://pl.wikipedia.org/wiki/Wie%C5%BCa_Babel

1. Utwórz instancję usługi. Wejdź na stronę

<https://cloud.ibm.com/catalog/services/language-translator>

i kliknij przycisk *Create*. Spowoduje to automatyczne wygenerowanie klucza API oraz wyświetlenie krótkiego tutoriala korzystania z usługi.

2. Pobierz klucz API. Kliknij *Manage* na panelu z lewej strony, a w sekcji *Credentials* kliknij *Show Credentials*. Skopiuj do schowka widoczny klucz API i wklej go (jako łańcuch) na pozycję `translate_key` w pliku *keys.py*.

13.6.1.4. Podgląd danych uwierzytelniających

Jeżeli nie zapisałeś kluczy API wygenerowanych dla poszczególnych usług lub utraciłeś je w jakikolwiek sposób, możesz je łatwo odczytać, wchodząc na stronę

<https://cloud.ibm.com/resources>

rozwijając na lewym panelu pozycję *Services* i klikając nazwę odpowiedniej usługi.

13.6.2. Aplikacja w akcji

Gdy już uzyskasz klucze API i prawidłowo wpiszesz je do pliku *keys.py*, możesz uruchomić skrypt *SimpleLanguageTranslator.py* znajdujący się w folderze z przykładami do rozdziału 13. Uruchom więc Anaconda Prompt (w Windows), Terminal (w MacOS i Linuksie) lub shell (w Linuksie), przejdź do folderu z przykładami do rozdziału 13., a następnie uruchom interpreter IPython w trybie pojedynczego skryptu³¹ za pomocą polecenia

```
ipython SimpleLanguageTranslator.py
```

Wykonywanie skryptu przebiega w dziesięciu opisywanych poniżej krokach, których zadania wyjaśnione są w komentarzach towarzyszących kodowi źródłowemu.

13.6.2.1. Przetwarzanie pytania

Po uruchomieniu aplikacji realizuje **krok 1.**, wypisując komunikat

Naciśnij Enter i zadaj pytanie po angielsku.

i oczekując na naciśnięcie klawisza *Enter*. Po jego naciśnięciu wypisuje kolejny komunikat

'Nagrywanie 5-sekundowe.'

a Ty masz możliwość nagrania (przez mikrofon) 5-sekundowego pytania. (My zapytaliśmy *Where is the closest bathroom?* — „Gdzie znajduje się najbliższa łazienka?”)³². Po upływie tego czasu aplikacja kończy nagrywanie i wypisuje komunikat

Nagrywanie zakończone.

W **kroku 2.** wywoływana jest usługa *Speech to Text*, która dokonuje konwersji nagranego pytania na tekst i wypisuje go:

³¹ Moduł *playback*, który wykorzystujemy w naszej aplikacji, wyświetli ostrzeżenie, gdy ją uruchomimy. Ostrzeżenie to nie ma związku z wykorzystywanymi przez nas funkcjami tego modułu i można je zignorować. Można również pozbyć się go całkowicie, instalując moduł *ffmpeg* ze strony <https://www.ffmpeg.org/> (w Windows, Linuksie i MacOS).

³² Byłoby sympatycznie, gdyby jedna ze stron prowadzących opisywaną konwersację posługiwała się językiem polskim; niestety, jak dotąd platforma Watson nie obsługuje naszego języka ojczystego, więc „spolszczając” opisywaną aplikację, z konieczności musieliśmy pozostawić treść dialogu w postaci oryginalnej — *przyp. red. wydania polskiego*.

Angielski: where is the closest bathroom

W **kroku 3.** usługa *Language Translator* dokonuje tłumaczenia angielskiego tekstu na język hiszpański i wypisuje rezultat tego tłumaczenia:

Hiszpański: ¿Dónde está el baño más cercano?

W **kroku 4.** z przetłumaczonego tekstu syntetyzowana jest mowa — rolę syntezatora spełnia usługa *Text to Speech* — a wynik tej syntezy zapisywany jest do pliku *spanish.wav*.

Krok 5. to odtworzenie zawartości pliku *spanish.wav*, za pomocą funkcji importowanych z modułu *pydub*.

Zakończona została pierwsza część scenariusza aplikacji — hiszpański rozmówca usłyszał pytanie w swoim języku.

13.6.2.2. Przetwarzanie odpowiedzi

Teraz inicjatywę powinien przejąć hiszpański rozmówca. Aplikacja, realizując **krok 6.**, wypisuje komunikat

Naciśnij Enter i udziel odpowiedzi po hiszpańsku.

i oczekuje na naciśnięcie klawisza *Enter*. Po jego naciśnięciu wypisuje komunikat

'Nagrywanie 5-sekundowe.'

dając rozmówcy możliwość udzielenia 5-sekundowej³³ odpowiedzi, która zapisana zostaje w pliku *spanishresponse.wav*. Nie posługujemy się biegle językiem hiszpańskim, użyliśmy więc uprzednio przygotowanego klipu *spokenResponse.wav*, w którym zaprzyjaźniona lektorka nagrała odpowiedź *El baño más cercano está en el restaurante* — „Najbliższa łazienka znajduje się w restauracji” — i odtworzyliśmy ów klip do mikrofonu komputera, z odpowiednio wysokim poziomem głośności.

W **kroku 7.** zarejestrowana odpowiedź konwertowana jest na hiszpański tekst

Odpowiedź po hiszpańsku: el baño más cercano está en el restaurante

W **kroku 8.** ponownie wywoływana jest usługa *Language Translator*, tym razem w celu przetłumaczenia tekstu hiszpańskiego na język angielski:

Odpowiedź po angielsku: The nearest bathroom is in the restaurant

W **kroku 9.** usługa *Text to Speech* syntetyzuje angielską odpowiedź na podstawie tego tekstu i zapisuje ją w pliku *englishresponse.wav*.

Krok 10. to ostatni akt dialogu — odtworzenie zawartości pliku *englishresponse.wav* na użytek angielskiego rozmówcy.

13.6.3. Analiza kodu aplikacji — skrypt „SimpleLanguageTranslator.py”

Zobaczymy teraz, jak opisywana funkcjonalność aplikacji przekłada się na jej kod źródłowy, zawarty w pliku *SimpleLanguageTranslator.py*. Przeanalizujemy ów kod zgodnie z zasadami analizy zstępującej — od ogółu do szczegółów (ang. *top-down analysis*). „Ogół” sformułowaliśmy już na wstępie, przypomnijmy:

³³ Dla prostoty ustaliliśmy w kodzie czas nagrywania na 5 sekund (jest to wartość zmiennej *SECONDS* w funkcji *record_audio*). W profesjonalnej aplikacji pożądanym byłoby bardziej inteligentne rozwiązanie — automatyczne rozpoczęcie nagrywania w momencie rozpoczęcia przemowy i jego zakończenie po upływie predefiniowanego okresu ciszy. Nie chcieliśmy jednak komplikować zbytnio przykładu, dedykowanego przede wszystkim zastosowaniu usług platformy Watson.

Ułatwienie konwersacji dwojga rozmówców, z których jeden posługuje się tylko językiem angielskim, a drugi tylko językiem hiszpańskim.

Pierwszy poziom uszczegółowienia jest dość oczywisty:

Przetłumaczenie wypowiedzi rozmówcy anglojęzycznego na mowę w języku hiszpańskim.

Przetłumaczenie wypowiedzi rozmówcy hiszpańskojęzycznego na mowę w języku angielskim.

Na kolejnym poziomie szczegółowości każdy z tych etapów przekłada się na pięć punktów:

Przetłumaczenie wypowiedzi rozmówcy anglojęzycznego na mowę w języku hiszpańskim.

Sygnal do zadania pytania w języku angielskim i nagrywanie głosu do pliku audio.

Konwersja mowy angielskiej na tekst.

Tłumaczenie tekstu angielskiego na język hiszpański.

Synteza głosu na podstawie tekstu hiszpańskiego.

Odtworzenie pliku audio ze zszyntetyzowaną mową hiszpańską.

Przetłumaczenie wypowiedzi rozmówcy hiszpańskojęzycznego na mowę w języku angielskim.

Sygnal do udzielenia odpowiedzi w języku hiszpańskim i nagrywanie głosu do pliku audio.

Konwersja mowy hiszpańskiej na tekst.

Tłumaczenie tekstu hiszpańskiego na język angielski.

Synteza głosu na podstawie tekstu angielskiego.

Odtworzenie pliku audio ze zszyntetyzowaną mową angielską.

Kroki 2. i 7. związane są z usługą *Speech to Text*, kroki 3. i 8. z usługą *Language Translator*, a kroki 4. i 9. — z usługą *Text to Speech*.

Przedstawiony podział, stanowiący przejaw strategii „dziel i zwyciężaj”, ułatwia projektowanie i analizowanie aplikacji, umożliwia bowiem skoncentrowanie się w danej chwili tylko na wąskim aspekcie bardziej złożonego problemu.

13.6.3.1. Importowanie klas z pakietu „Watson SDK”

W wierszach 14. – 16. na listingu 13.1a importowane są z modułu `watson_developer_cloud` klasy zapewniające interakcję z usługami platformy Watson. Każda z tych klas uzyskuje łączność z daną usługą przy użyciu poświadczeń (kluczy API) uzyskanych podczas tworzenia instancji poszczególnych usług:

- klasa `SpeechToTextV1`³⁴ przekazuje plik audio do usługi *Speech to Text*, w celu jego konwersji na tekst, i odbiera wynikowy tekst w postaci dokumentu JSON³⁵;

³⁴ Człon „V1” w nazwie klasy oznacza „wersję nr 1” usługi. W miarę, jak firma IBM modyfikuje swe usługi, dodaje do modułu `watson_developer_cloud` nowe wersje klas, zamiast modyfikowania klas już w tym module obecnych. W konsekwencji istniejące aplikacje, stworzone z użyciem wspomnianego modułu, stają się niewrażliwe na rozwijanie usług platformy Watson. Obecnie (początek roku 2020) klasy dla usług *Speech to Text* i *Text to Speech* istnieją w wersji 1 (V1), natomiast klasa współdziałająca z usługą *Language Translator* posiada już trzy wersje (najnowsza to V3).

³⁵ Notację JSON i jej dokumenty opisywaliśmy w rozdziałach 9. i 12.

- klasa `LanguageTranslatorV3` przekazuje tekst źródłowy do usługi *Language Translator* i odbiera przetłumaczony tekst wynikowy w postaci dokumentu JSON;
- klasa `TextToSpeechV1` przekazuje tekst pisany do usługi *Watson Text to Speech* i odbiera wynik jego konwersji (we wskazanym języku) na mowę.

LISTING 13.1A. Program „SimpleLanguageTranslator” — część pierwsza (importowanie klas do interakcji z platformą Watson)

```

1 # SimpleLanguageTranslator.py
2 #-*- coding: 1250 -*-
3
4 """ Aplikacja ułatwiająca porozumiewanie się dwóch rozmówców,
5     z których jeden posługuje się językiem angielskim, a drugi
6     hiszpańskim. Wykorzystuje API następujących usług platformy
7     IBM Watson:
8
9         Speech to Text
10        Text to Speech
11        Language Translator
12 """
13
14 from watson_developer_cloud import SpeechToTextV1
15 from watson_developer_cloud import LanguageTranslatorV3
16 from watson_developer_cloud import TextToSpeechV1

```

13.6.3.2. Importowanie pozostałych modułów

W wierszu 17. na listingu 13.1b importowane są z pliku *keys.py* klucze API warunkujące dostęp do usług platformy Watson, natomiast moduły importowane w wierszach 18. – 21. związane są z przetwarzaniem materiału audio w aplikacji:

- moduł `pyaudio` umożliwia nagrywanie głosu przez mikrofon;
- moduły `pydub` i `pydub.playback` umożliwiają pobieranie i odtwarzanie zawartości plików audio;
- moduł `wave` z biblioteki standardowej Pythona umożliwia zapisywanie materiału audio w plikach w formacie WAV (*Waveform Audio File Format*), opracowanym przez Microsoft i IBM. W aplikacji używamy go do tworzenia plików *.wav*, które następnie przekazywane są jako dane wejściowe do usługi *Speech to Text*.

LISTING 13.1B. Program „SimpleLanguageTranslator” — część druga (importowanie pozostałych modułów)

```

17 import keys # zawiera klucze API dla poszczególnych usług
18 import pyaudio # wykorzystywany do nagrywania przez mikrofon
19 import pydub # wykorzystywany do ładowania pliku WAV
20 import pydub.playback # wykorzystywany do odtwarzania pliku WAV
21 import wave # wykorzystywany do zapisywania w pliku WAV

```

13.6.3.3. Program główny — funkcja „run_translator”

Zasadnicza funkcjonalność aplikacji skupiona jest w funkcji `run_translator` — to ona zostaje wywołana, gdy uruchamiamy plik skrypt *SimpleLanguageTranslator.py* jako skrypt. Zgodnie z wcześniejszym opisem funkcjonalnym, podzieliliśmy treść tej funkcji na 10 kroków, które wyraźnie zaznaczyliśmy za pomocą komentarzy.

W **kroku 1.** (wiersze 27. – 30. na listingu 13.1c) funkcja zachęca użytkownika do naciśnięcia klawisza *Enter* i zadania krótkiego pytania po angielsku; gdy użytkownik naciśnie rzeczywisty klawisz, wywołana funkcja `record_audio` przez 5 sekund rejestruje dźwięk przez mikrofon, zapisując materiał dźwiękowy w pliku *english.wav*.

LISTING 13.1C. Program „SimpleLanguageTranslator” — część trzecia (nagrywanie angielskiego pytania)

```
23 def run_translator():
24     """Wywołuje funkcje realizujące interakcję
25         z usługami IBM Watson"""
26
27     # Krok 1. Sygnał do zadania pytania w języku angielskim
28     #     i nagrywanie głosu do pliku audio
29     input('Naciśnij Enter i zadaj pytanie po angielsku.')
30     record_audio('english.wav')
```

W **kroku 2.** plik z nagraniem pytaniem przekazywany jest do usługi *Speech to Text* w celu skonwertowania go na tekst pisany w języku angielskim (listing 13.1d) przy użyciu predefiniowanego modelu `en-US_BroadbandModel`³⁶. Wynikowy tekst jest następnie wypisywany:

LISTING 13.1D. Program „SimpleLanguageTranslator” — część czwarta (konwersja angielskiego pytania na tekst)

```
32     # Krok 2. Konwersja mowy angielskiej na tekst
33     english = speech_to_text(
34         file_name='english.wav', model_id='en-US_BroadbandModel')
35     print('Angielski: ', english)
```

Uzyskany w ten sposób tekst angielski jest w **kroku 3.** tłumaczony na wersję hiszpańską (listing 13.1e). Tłumaczenie wykonywane jest przez usługę *Language Translator* według predefiniowanego modelu `en-es` — `en` oznacza język angielski, `es` hiszpański. Wynik tłumaczenia jest następnie wypisywany.

LISTING 13.1E. Program „SimpleLanguageTranslator” — część piąta (tłumaczenie tekstu angielskiego)

```
37     # Krok 3. Tłumaczenie tekstu angielskiego na język hiszpański
38     spanish = translate(text_to_translate=english, model='en-es')
39     print('Hiszpański: ', spanish)
```

Uzyskany w wyniku tłumaczenia tekst hiszpański konwertowany jest w **kroku 4.** na mowę (listing 13.1f); w wywołaniu usługi *Text to Speech* wybrano barwę głosu `es-US_SofiaVoice` i zapisanie wyniku konwersji do pliku audio *spanish.wav*.

LISTING 13.1F. Program „SimpleLanguageTranslator” — część szósta (konwersja tekstu hiszpańskiego na mowę)

```
41     # Krok 4. Synteza głosu na podstawie tekstu hiszpańskiego
42     text_to_speech(text_to_speak=spanish, voice_to_use='es-US_SofiaVoice',
43         file_name='spanish.wav')
```

³⁶ Dla większości języków usługa *Speech to Text* oferuje dwa modele — szerokopasmowy (*broadband*) i wąskopasmowy (*narrowband*), zależnie od jakości źródłowego materiału audio — dla dźwięku próbkowanego z częstotliwością 16 kHz lub większą, IBM zaleca model szerokopasmowy. W naszej aplikacji próbujemy dźwięk z częstotliwością 44,1 kHz.

Zawierający wynik syntezy plik *spanish.wav* jest w **kroku 5**, odtwarzany za pomocą funkcji `play_audio` (listing 13.1g) — rozmówcy słyszą hiszpańską wersję oryginalnego pytania.

LISTING 13.1G. Program „SimpleLanguageTranslator” — część siódma
(odtworzenie pytania w wersji hiszpańskiej)

```
45 # Krok 5. Odtworzenie pliku audio ze zszyntetyzowaną
46 # mową hiszpańską
47 play_audio(file_name='spanish.wav')
```

W ten sposób zrealizowana została pierwsza część konwersacji — hiszpańskojęzyczny rozmówca usłyszał w swoim języku treść pytania zadanego przez anglojęzycznego interlokutora. Druga część konwersacji — udzielenie odpowiedzi — przebiega według analogicznego scenariusza, lecz tłumaczenie tekstu przebiega w odwrotnym kierunku, z języka hiszpańskiego na angielski (listing 13.1h):

- w kroku 6. nagrywana jest odpowiedź w języku hiszpańskim;
- w kroku 7. utworzony plik audio konwertowany jest (przez usługę *Speech to Text*) na tekst, według predefiniowanego modelu `es-ES_BroadbandModel`;
- krok 8. to tłumaczenie wynikowego tekstu hiszpańskiego na język angielski, za pomocą usługi *Language Translator*, według predefiniowanego modelu `es-en`;
- w kroku 9. przetłumaczony tekst konwertowany jest przez usługę *Text to Speech* na mowę o barwie głosu `en-US_AllisonVoice`; wynik konwersji zapisywany jest do pliku *englishresponse.wav*;
- w kroku 10. plik *englishresponse.wav* jest odtwarzany za pomocą funkcji `play_audio` — rozmówcy słyszą angielską wersję odpowiedzi.

LISTING 13.1H. Program „SimpleLanguageTranslator” — część ósma (przekazywanie odpowiedzi)

```
49 # Krok 6. Sygnał do udzielenia odpowiedzi w języku hiszpańskim
50 # i nagrywanie głosu do pliku audio
51 input('Naciśnij Enter i udziel odpowiedzi po hiszpańsku.')
52 record_audio('spanishresponse.wav')
53
54 # Krok 7. Konwersja mowy hiszpańskiej na tekst
55 spanish = speech_to_text(
56     file_name='spanishresponse.wav', model_id='es-ES_BroadbandModel')
57 print('Odpowiedź po hiszpańsku: ', spanish)
58
59 # Krok 8. Tłumaczenie tekstu hiszpańskiego na język angielski
60 english = translate(text_to_translate=spanish, model='es-en')
61 print('Odpowiedź po angielsku: ', english)
62
63 # Krok 9. Synteza głosu na podstawie tekstu angielskiego
64 text_to_speech(text_to_speak=english,
65     voice_to_use='en-US_AllisonVoice',
66     file_name='englishresponse.wav')
67
68 # Krok 10. Odtworzenie pliku audio ze zszyntetyzowaną
69 # mową angielską
70 play_audio(file_name='englishresponse.wav')
```

Tak oto rozmówcy posługujący się różnymi językami wymienili między sobą pierwszą informację.

Przyjrzyjmy się teraz szczegółom implementacji funkcji wywoływanych w krokach 1. – 10.

13.6.3.4. Funkcja „speech_to_text”

W celu wywołania usługi *Speech to Text* funkcja `speech_to_text` (wiersze 73. – 105. na listingu 13.2a) tworzy obiekt klasy `SpeechToTextV1` i przypisuje jego referencję do zmiennej `stt` (skrót od *speech-to-text*); argumentem wywołania konstruktora jest klucz API zapewniający dostęp do usługi. W ramach instrukcji `with` (wiersze 81. – 84.) zostaje otwarty plik audio, o nazwie określonej przez parametr `file_name`, do odczytu binarnego (tryb `rb`) i powiązany zostaje ze zmienną plikową `audio_file`. Następnie, w wierszach 83. – 84. wywoływana jest metoda `recognize` obiektu `SpeechToTextV1` reprezentowanego przez zmienną `stt`. W wywołaniu tym występują trzy argumenty kluczowe:

- `audio` oznacza zmienną plikową `audio_file` reprezentującą otwarty plik z materiałem dźwiękowym;
- `content_type` reprezentuje typ treści zapisanej w pliku audio — `audio/wav` oznacza dźwięk w formacie `wav`³⁷;
- `model` identyfikuje model języka mówionego, który usługa wykorzystywać będzie do konwersji mowy na tekst. Używamy dwóch modeli szerokopasmowych: `en-US_BroadbandModel` (dla języka angielskiego) i `es-ES_BroadbandModel` (dla języka hiszpańskiego).

LISTING 13.2A. Funkcje uruchamiające usługi platformy Watson — część pierwsza
(funkcja „speech_to_text”)

```
73 def speech_to_text(file_name, model_id):
74     """Wywołanie usługi Watson Speech to Text w celu
75         skonwertowania głosu na tekst pisany"""
76
77     # utworzenie klienta usługi
78     stt = SpeechToTextV1(iam_apikey=keys.speech_to_text_key)
79
80     # otwarcie pliku audio
81     with open(file_name, 'rb') as audio_file:
82         # przekazanie zawartości pliku do konwersji
83         result = stt.recognize(audio=audio_file,
84                               content_type='audio/wav', model=model_id).get_result()
85
86     # Pobranie listy wyników. Może ona zawierać, oprócz wyniku
87     # końcowego, także wyniki pośrednie, zależnie od żądanej metody
88     # rozpoznawania głosu. W tym przypadku żądamy tylko końcowego
89     # wyniku, więc lista zawiera tylko jeden element.
90
91     results_list = result['results']
92
93     # Pobierz końcowy wynik rozpoznawania
94     speech_recognition_result = results_list[0]
```

³⁷ Typ treści oznaczany był dawniej akronimem **MIME** (*Multipurpose Internet Mail Extensions*) — to standard zdefiniowany na potrzeby identyfikowania typu załączników do poczty elektronicznej.

```

95
96 # Pobranie listy wariantów transkrypcji. Jej liczebność
97 # zależy od żądanej metody rozpoznawania głosu.
98 # W tym przypadku lista zawiera tylko jeden element.
99 alternatives_list = speech_recognition_result['alternatives']
100 first_alternative = alternatives_list[0]
101
102 # Pobranie tekstu stanowiącego wynik transkrypcji
103 # i zwrócenie go jako wyniku
104 transcript = first_alternative['transcript']
105 return transcript

```

Metoda `recognize` zwraca wynik w postaci obiektu klasy `DetailedResponse`. Jego metoda `get_result` zwraca dokument JSON zawierający tekst będący wynikiem konwersji, który zapisujemy w zmiennej `result`. Wspomniany dokument ma strukturę podobną do przedstawionej na rysunku 13.1, jego dokładna treść zależy od treści pytania.

```

{
  "results": [
    {
      "alternatives": [
        {
          "confidence": 0.983,
          "transcript": "where is the closest bathroom "
        }
      ],
      "final": true
    }
  ],
  "result_index": 0
}

```

Wiersz 91
Wiersz 94
Wiersz 99
Wiersz 100
Wiersz 104

RYSUNEK 13.1. Struktura przykładowego dokumentu JSON zwracanego przez usługę „Speech to Text”

Jak widzimy, dokument ten to w istocie *zagnieżdżone* słowniki i listy. Aby uczynić nawigowanie po tej strukturze bardziej przejrzystym, w wierszach 91. – 104. podzieliśmy scenariusz tej nawigacji na instrukcje cząstkowe, aż do uzyskania tekstu `where is the closest bathroom`, który zwrócony zostaje jako wynik. Znaczenie wspomnianych instrukcji cząstkowych jest następujące:

- w wierszu 91. zwracana jest lista, skojarzona w słowniku z kluczem `'results'`:

```
results_list = result['results']
```

Zależnie od argumentów wywołania metody `recognize`, lista ta może zawierać, oprócz ostatecznego wyniku, także wyniki pośrednie — te mogą być użyteczne przy transkrypcji materiału dźwiękowego pozyskiwanego „na żywo”, na przykład wiadomości lub reportażu. Jako że zażądaliśmy tylko wyniku końcowego, wspomniana lista zawiera tylko jeden element³⁸, który w wierszu 94. przypisywany jest do zmiennej `speech_recognition_result` jako jedyny wynik:

```
speech_recognition_result = results_list[0]
```

³⁸ Szczegóły dotyczące argumentów wywołania metody `recognize` i struktury zwracanego przez nią dokumentu JSON opisane są na stronie <https://cloud.ibm.com/apidocs/speech-to-text/speech-to-text?code=python>

- Na końcowy wynik konwersji może składać się kilka wariantów tekstu, tworzących listę opatrzoną kluczem 'alternatives', każdy opatrzony subiektywną oceną jakości dopasowania, w skali 0,0 – 1,0. Instrukcja w wierszu 99.

```
alternatives_list = speech_recognition_result['alternatives']
```

przypisuje tę listę do zmiennej `alternatives_list`. Ponieważ żądamy tylko jednego wariantu, wspomniana lista zawiera tylko jeden element, który w wierszu 100. przypisywany jest do zmiennej `first_alternative`:

```
first_alternative = alternatives_list[0]
```

- Każdy z wariantów konwersji jest słownikiem, w którym pod kluczem 'transcript' kryje się łańcuch, zawierający tekst stanowiący wynik konwersji. Przypisujemy go do zmiennej roboczej, którą następnie zwracamy jako wynik (wiersze 104. – 105.):

```
transcript = first_alternative['transcript']
```

```
return transcript
```

Wiersze 91. – 104. można zapisać w postaci pojedynczej instrukcji

```
return result['results'][0]['alternatives'][0]['transcript']
```

dla celów dydaktycznych zdekomponowaliśmy ją jednak na kilka prostszych instrukcji.

13.6.3.5. Funkcja „translate”

Funkcja `translate` (wiersze 108. – 132. na listingu 13.2b), pośrednicząca w wywołaniu usługi *Language Translator*, posiada dwa parametry: łańcuch zawierający tekst źródłowy oraz identyfikator modelu, jaki usługa *Laanguage Translator* wykorzystywać będzie w celu zrozumienia tekstu źródłowego i wyprodukowania jego odpowiednika w języku docelowym. W naszym przykładzie wykorzystujemy dwa predefiniowane modele IBM: en-es do tłumaczenia z języka angielskiego na hiszpański oraz es-en do tłumaczenia w przeciwnym kierunku. Funkcja rozpoczyna swoją pracę od utworzenia obiektu klasy *LanguageTranslatorV3* i przypisania jego referencji do zmiennej `language_translator` (wiersze 115. – 116.). Parametrami konstruktora są: wersja usługi ('2018-05-31'³⁹) i klucz API zapewniający dostęp do usługi. Uruchomienie usługi następuje w wyniku wywołania metody `translate` tegoż obiektu (wiersze 119. – 120.). Metoda ta wywoływana jest z dwoma argumentami kluczowymi, identycznymi z parametrami metody `translate`:

- `text` reprezentuje tekst źródłowy;
- `model_id` jest identyfikatorem modelu tłumaczeniowego.

LISTING 13.2B. Funkcje uruchamiające usługi platformy Watson — część druga (funkcja „translate”)

```
108 def translate(text_to_translate, model):
109     """Wywołanie usługi Watson Language Translator w celu
110     przetłumaczenia tekstu angielskiego na hiszpański
111     (model en-es) albo hiszpańskiego na angielski
112     (model es-en)"""
113
```

³⁹ Zgodnie z *API Reference* usługi *Language Translator*, wersja 2018-05-31 jest wersją najnowszą w czasie, gdy piszemy tę książkę. IBM zmienia oznaczenie wersji usługi tylko wtedy, gdy wskutek dokonanych w niej zmian staje się ona niekompatybilna z aplikacjami wykorzystującymi jej dotychczasową wersję. Dzięki temu aplikacje zyskują gwarancję, że niezależnie od rozwijania danej usługi przez IBM, korzystać będą zawsze z jej właściwej wersji, czyli tej, dla której zostały stworzone. Szczegółowe informacje na ten temat znajdują Czytelnicy na stronie <https://www.ibm.com/watson/developercloud/language-translator/api/v3/python.html?python-versioning>

```

114 # utworzenie klienta usługi
115 language_translator = LanguageTranslatorV3(version='2018-05-31',
116     iam_apikey=keys.translate_key)
117
118 # wykonanie tłumaczenia
119 translated_text = language_translator.translate(
120     text=text_to_translate, model_id=model).get_result()
121
122 # Pobranie listy wynikowych łańcuchów — każdy z nich
123 # odpowiada jednemu łańcuchowi źródłowemu. W tym przypadku
124 # przekazujemy tylko jeden łańcuch źródłowy, więc lista
125 # wyników zawiera tylko jeden element.
126 translations_list = translated_text['translations']
127 first_translation = translations_list[0]
128
129 # pobranie przetłumaczonego tekstu i zwrócenie go
130 # jako wyniku
131 translation = first_translation['translation']
132 return translation

```

Wynik zwracany przez usługę *Language Translator*, a w konsekwencji przez metodę `LanguageTranslatorV3.translate`, ma postać dokumentu JSON o strukturze podobnej do przedstawionej na rysunku 13.2.

```

{
  "translations": [
    {
      "translation": "¿Dónde está el baño más cercano? "
    }
  ],
  "word_count": 5,
  "character_count": 30
}

```

RYСУNEK 13.2. Struktura przykładowego dokumentu JSON zwracanego przez usługę „Language Translator”

Konkretna treść dokumentu JSON zależy jest od zadanego pytania, pod względem strukturalnym jest on kombinacją zagnieżdżonych słowników i list:

- W najbardziej zewnętrznym słowniku, reprezentowanym przez zmienną `translated_text` (wiersz 119.), pod kluczem `'translations'` ukrywa się lista fragmentów tekstu wynikowego, każdy fragment odpowiada jednemu łańcuchowi tekstu źródłowego (wiersz 126.).
`translations_list = translated_text['translations']`
- Jako że przekazaliśmy tylko jeden łańcuch tekstu źródłowego, wspomniana lista zawiera tylko jeden element, który przypisujemy do zmiennej `first_translation` (wiersz 127.).
`first_translation = translations_list[0]`
- Jedyny fragment tekstu wynikowego, kryjący się pod zmienną `first_translation`, jest słownikiem, pod którego kluczem `'translation'` ukrywa się interesujący nas łańcuch (wiersz 131.) zwracany ostatecznie jako wynik funkcji (wiersz 132.):
`translation = first_translation['translation']`
`return translation`

Wiersze 119. – 132. można zapisać w postaci pojedynczej instrukcji

```
return translated_text['translations'][0]['translation']
```

ponownie jednak dla celów dydaktycznych zdekomponowaliśmy ją na kilka prostszych.

13.6.3.6. Funkcja „text_to_speech”

Usługa *Text to Speech* uruchamiana jest w wyniku wywołania funkcji `text_to_speech` (wiersze 134. – 146. na listingu 13.2c). Jej parametry reprezentują kolejno tekst konwertowany na mowę (`text_to_speak`), barwę głosu produkowanego materiału audio — jedną z predefiniowanych (`voice_to_use`) — i nazwę pliku (`file_name`), w którym zapisany zostanie materiał audio wyprodukowany przez usługę. Funkcja rozpoczyna swą pracę od utworzenia obiektu klasy `TextToSpeechV1` i przypisania jego referencji do zmiennej `tts` (skrót od *text-to-speech*); argumentem wywołania konstruktora jest klucz API zapewniający dostęp do usługi. W ramach instrukcji `with` (wiersze 144. – 146.) zostaje utworzony wspomniany wcześniej plik binarny (tryb `wb`) o nazwie określonej przez parametr `file_name` i powiązany zostaje ze zmienną plikową `audio_file`.

LISTING 13.2C. Funkcje uruchamiające usługi platformy Watson — część trzecia (funkcja „text_to_speech”)

```
134 def text_to_speech(text_to_speak, voice_to_use, file_name):
135     """Wywołanie usługi Watson Text to Speech w celu
136         zsyntetyzowania mowy, z żądaną barwą głosu, i zapisanie
137         wyniku syntezy w pliku WAV o żądanej nazwie."""
138
139     # utworzenie klienta usługi
140     tts = TextToSpeechV1(iam_apikey=keys.text_to_speech_key)
141
142     # utworzenie pliku, do którego zapisany zostanie
143     # wynik syntezy
144     with open(file_name, 'wb') as audio_file:
145         audio_file.write(tts.synthesize(text_to_speak,
146             accept='audio/wav', voice=voice_to_use).get_result().content)
```

W wierszach 145. – 146. wywoływane są dwie metody. Najpierw uruchamiana jest usługa *Text to Speech* poprzez wywołanie metody `synthesize` obiektu `TextToSpeechV1` z trzema argumentami:

- `text_to_speak` jest powieleniem pierwszego parametru funkcji, czyli łańcuchem zawierającym tekst konwertowany na mowę;
- `accept` określa format, w jakim usługa wyprodukować ma materiał audio — jak wcześniej wyjaśnialiśmy, `audio/wav` oznacza format `wav`;
- `voice` jest powieleniem drugiego parametru funkcji, czyli określeniem barwy głosu produkowanego materiału audio. W naszym przykładzie używamy dwóch predefiniowanych głosów: `en-US_AllisonVoice` dla mowy angielskiej i `es-US_SofiaVoice` dla mowy hiszpańskiej. Platforma Watson oferuje wiele głosów męskich i żeńskich dla różnych języków⁴⁰.

⁴⁰ Kompletna ich lista dostępna jest pod adresem <https://cloud.ibm.com/docs/services/text-to-speech?topic=text-to-speech-voices#languageVoices>

Zachęcamy do eksperymentowania!

Metoda `synthesize` zwraca wynik w postaci obiektu klasy `DetailedResponse`. Jego metoda `get_result` udostępnia strukturę reprezentującą wynikowy materiał dźwiękowy; właściwość `content` tej struktury reprezentuje strumień bajtów, który zapisujemy do przygotowanego pliku.

13.6.3.7. Funkcja „record_audio”

Nagrywanie mowy przez mikrofon i rejestrowanie dźwięku w pliku WAV jest zadaniem funkcji `record_audio` (wiersze 148. – 186. na listingu 13.3a). Parametrem funkcji jest nazwa docelowego pliku. Implementacja rejestratora importowana jest z modułu `pyaudio`; fizyczne detale rejestracji określone są przez zestaw zmiennych klasowych w wierszach 153. – 158. — zgodnie z dokumentacją modułu, ich znaczenie jest następujące:

- `FRAME_RATE` — liczba ramek tworzonych w ciągu sekundy: 44 100, czyli częstotliwość 44,1 kHz, to wartość typowa dla dźwięku nagrywanego na płycie CD;
- `CHUNK` — liczba ramek w buforze: tworzone ramki są buforowane, po wypełnieniu bufora jego zawartość przekazywana jest jednorazowo do aplikacji — w tym przypadku w buforze mieszczą się 1024 ramki);
- `FORMAT` — identyfikator formatu ramki: `paInt16` oznacza ramkę w postaci 16-bitowej liczby całkowitej;
- `CHANNELS` — liczba kanałów rejestrowanych w każdej ramce, w tym przypadku stereofonia 2-kanałowa;
- `SECONDS` — całkowity czas nagrania w sekundach (5).

LISTING 13.3A. Funkcje pomocnicze — część pierwsza (funkcja „record_audio”)

```
148 def record_audio(file_name):
149     """Zapisanie krótkiego (do 5 sekund) nagrania
150         w pliku WAV o podanej nazwie, z wykorzystaniem
151         modułu pyaudio"""
152
153     FRAME_RATE = 44100 # liczba ramek na sekundę
154     CHUNK = 1024 # liczba ramek odczytywanych jednorazowo
155     FORMAT = pyaudio.paInt16 # każda ramka jest 16-bitową
156                             # liczbą całkowitą
157     CHANNELS = 2 # 2 sample na ramkę
158     SECONDS = 5 # całkowity czas nagrania
159
160     # otwarcie i zamknięcie strumienia audio
161     recorder = pyaudio.PyAudio()
162
163     # skonfigurowanie i otwarcie strumienia audio
164     # do nagrywania (input=True)
165     audio_stream = recorder.open(format=FORMAT, channels=CHANNELS,
166                                  rate=FRAME_RATE, input=True, frames_per_buffer=CHUNK)
167
168     # Zapisywanie surowych bajtów z wejścia mikrofonowego
169     audio_frames = []
170     print('Nagrywanie 5-sekundowe.')
171
172     # odczytywanie nagrania w porcjach określonych przez CHUNK
173     for i in range(0, int(FRAME_RATE * SECONDS / CHUNK)):
174         audio_frames.append(audio_stream.read(CHUNK))
175
176     print('Nagrywanie zakończone.')
```

```

177     audio_stream.stop_stream() # zatrzymanie nagrywania
178     audio_stream.close()
179     recorder.terminate() # zwolnienie zasobów używanych przez pyaudio
180
181     # zapisanie ramek audio w pliku WAV
182     with wave.open(file_name, 'wb') as output_file:
183         output_file.setnchannels(CHANNELS)
184         output_file.setsampwidth(recorder.get_sample_size(FORMAT))
185         output_file.setframerate(FRAME_RATE)
186         output_file.writeframes(b''.join(audio_frames))

```

W wierszu 161. tworzony jest obiekt klasy PyAudio implementujący funkcje rejestratora. Wywołanie metody open w wierszach 165. – 166. powoduje otwarcie strumienia audio i skonfigurowanie go zgodnie z parametrami FORMAT, CHANNELS, FRAME_RATE i CHUNK. Argument input=True oznacza odbieranie przez strumień sygnału z wejścia audio. Wynikiem zwracanym przez metodę (zapisywanym w zmiennej audio_stream) jest obiekt klasy Stream zapewniający interakcję ze wspomnianym strumieniem.

W wierszu 169. tworzona jest lista, w której kolekcjonowane będą *porcje ramek*, odczytywane ze strumienia przez metodę read obiektu audio_stream w wierszu 174. Wyrażenie

```
int(FRAME_RATE * SECONDS / CHUNK)
```

w wierszu 173. określa całkowitą liczbę takich porcji.

W wierszach 176. – 179. następuje zakończenie nagrywania: zamknięcie strumienia oraz zwolnienie obiektu rejestratora i używanych przez niego zasobów.

Na tym etapie dysponujemy więc listą audio_frames, której elementami są porcje ramek, i naszym zadaniem jest zapisanie jej zawartości w pliku audio. Zapis ten odbywa się pod kontrolą instrukcji with (wiersze 182. – 186.). W wierszu 182. funkcja open z modułu wave powoduje utworzenie binarnego pliku o nazwie tożsamej z argumentem wywołania funkcji record_audio i powiązanie tego pliku ze zmienną output_file. Zmienna ta umożliwia określenie (w wierszach 183. – 185.) konfiguracji zawartości pliku: liczby kanałów, rozmiaru próbki (zwracanego przez metodę get_sample_size rejestratora) i częstotliwości próbkowania. Po skonfigurowaniu nagłówka pliku, zapisujemy do niego (za pomocą metody writeframes) zawartość ramek z listy audio_frames — wyrażenie

```
b''.join(audio_frames)
```

oznacza konkatenację porcji ramek w postaci łańcucha bajtowego (*byte string*); przedrostek b powoduje, że wyrażenie '' traktowane jest jako pusty *łańcuch bajtów*, nie *łańcuch znaków*.

13.6.3.8. Funkcja „play_audio”

Do odtworzenia materiału audio wyprodukowanego przez usługę *Text to Speech* wykorzystujemy moduły pydub i pydub.playback. Funkcja play_audio rozpoczyna swoją pracę od utworzenia obiektu klasy AudioSegment i załadowania doń zawartości pliku audio za pomocą metody from_wav (wiersz 191. na listingu 13.3b). Pozostaje już tylko odtworzyć tę zawartość (wiersz 192).

LISTING 13.3B. Funkcje pomocnicze — część druga (funkcja „play_audio”)

```

188 def play_audio(file_name):
189     """Odtworzenie zawartości pliku WAV o podanej nazwie
190         za pomocą modułu pydub"""
191     sound = pydub.AudioSegment.from_wav(file_name)
192     pydub.playback.play(sound)

```

13.6.3.9. Wykonanie skryptu — wywołanie funkcji „run_translator”

Gdy plik *SimpleLanguageTranslator.py* wywoływany jest jako skrypt do wykonania, globalny symbol `__name__` ma wartość `'__main__'` i wówczas wywołana zostaje funkcja `run_translator`, opisywana szczegółowo w sekcji 13.6.3.3, która tym samym spełnia zadanie głównej funkcji modułu (listing 13.4).

LISTING 13.4. Wywołanie głównej funkcji modułu („run_translator”)

```
194 if __name__ == '__main__':  
195     run_translator()
```

Niniejsza analiza przypadku, oprócz niewątpliwie praktycznego znaczenia wynikającego z drobiazgowej prezentacji wywoływania usług platformy Watson w języku Python, ma ponadto walor dydaktyczny, ilustruje bowiem zastosowanie uniwersalnej strategii „dziel i zwyciężaj”, dzięki której dość skomplikowany program staje się zrozumiały dzięki dekompozycji jego kodu na prostsze fragmenty, łatwiejsze w zrozumieniu i analizie. Niejako przy okazji jest też prezentacją tego, jak w praktyce buduje się mashupy, o których wspominaliśmy na początku sekcji 13.6.

13.7. Zasoby powiązane z IBM Watson

Firma IBM zapewnia wszechstronne wsparcie dla deweloperów, którzy tworzą aplikacje na bazie oferowanych przez nią bibliotek, usług i narzędzi — zwłaszcza dla tych deweloperów, którzy dopiero zaczynają swoją przygodę w pewnym obszarze tematycznym.

13.7.1. Dokumentacja usług

Studiowanie dokumentacji usług platformy Watson najprościej rozpocząć od strony pod adresem

<https://cloud.ibm.com/developer/watson/documentation>

na której zgrupowane są sekcje poświęcone poszczególnym usługom. W każdej z takich sekcji znajdują się dwa linki, prowadzące (odpowiednio) do dokumentacji usługi (*Documentation*) i szczegółowego opisu jej interfejsu API (*API Reference*). Dokumentacja każdej z usług obejmuje zazwyczaj:

- tutorial *Getting started with...*,
- instruktażowy film ilustrujący korzystanie z tutoriala,
- link do demo zastosowania usługi,
- linki do dodatkowych informacji i wskazówek specyficznych dla danej usługi,
- linki do przykładowych aplikacji wykorzystujących usługę,
- linki do dodatkowych zasobów — specyficznych tutoriali, wideo, blogów itp.

W opisie interfejsu API usługi znajdziemy między innymi szczegóły korzystania z usług w różnych językach i środowiskach, na przykład w Pythonie i pakiecie *Watson Developer Cloud Python SDK* — na stronie opisu API należy kliknąć link *More* u góry prawego panelu i w rozwiniętym menu wybrać pozycję *Python* (lub link *Python*, jeśli jest widoczny od razu). Dla każdej usługi dostępne są: szczegółowy opis możliwych opcji jej wywoływania, możliwe typy zwracanych odpowiedzi, przykładowe odpowiedzi — i wiele innych przydatnych informacji.

13.7.2. Pakiety SDK dla platformy Watson

Do utworzenia skryptu opisywanego w niniejszym rozdziale wykorzystaliśmy pakiet *Watson Developer Cloud Python SDK*. Podobne pakiety SDK dostępne są dla wielu innych języków i platform, kompletna ich lista znajduje się pod adresem

<https://cloud.ibm.com/developer/watson/sdks-and-tools>

13.7.3. Zasoby edukacyjne

Na stronie *Learning Resources* pod adresem

<https://cloud.ibm.com/developer/watson/learning-resources>

znajdziesz linki prowadzące do następujących zasobów:

- postów blogerów na temat sztucznej inteligencji, platformy Watson i jej usług oraz ich wykorzystywania w przemyśle,
- repozytorium GitHub platformy Watson, zawierającego narzędzia dla deweloperów, pakiety SDK i przykładowy kod,
- kanału YouTube platformy Watson (patrz sekcja 13.7.4),
- wzorców kodu, które firma IBM określa mianem „map drogowych umożliwiających sprostanie skomplikowanym wyzwaniom programistycznym” (*roadmaps for solving complex programming challenges*). Wiele z nich zaimplementowanych jest w języku Python, lecz inne, zaimplementowane w innych językach, także mogą służyć za wzór dla kodu tworzonego w Pythonie.

13.7.4. Materiały wideo

W kanale YouTube platformy Watson, dostępnym pod adresem

<https://www.youtube.com/user/IBMWatsonSolutions>

znajdziesz kilkaset filmów i wideoklipów na temat różnych aspektów korzystania z tej platformy.

13.7.5. IBM Redbooks

Wymienione poniżej „czerwone księgi IBM” (*IBM Redbooks*), składające się na wszechstronny i szczegółowy opis chmury IBM i usług platformy Watson, stanowią cenny materiał przyciągający się do rozwijania umiejętności programistycznych i tworzenia coraz lepszych aplikacji.

- „Essentials of Application Development on IBM Cloud”, <http://www.redbooks.ibm.com/abstracts/sg248374.html>
- „Building Cognitive Applications with IBM Watson Services”, Volume 1: „Getting Started”, <http://www.redbooks.ibm.com/abstracts/sg248387.html>
- „Building Cognitive Applications with IBM Watson Services”, Volume 2: „Conversation (now called Watson Assistant)”, <http://www.redbooks.ibm.com/abstracts/sg248394.html>
- „Building Cognitive Applications with IBM Watson Services”, Volume 3: „Visual Recognition”, <http://www.redbooks.ibm.com/abstracts/sg248393.html>
- „Building Cognitive Applications with IBM Watson Services”, Volume 4: „Natural Language Classifier”, <http://www.redbooks.ibm.com/abstracts/sg248391.html>
- „Building Cognitive Applications with IBM Watson Services”, Volume 5: „Language Translator”, <http://www.redbooks.ibm.com/abstracts/sg248392.html>

- „Building Cognitive Applications with IBM Watson Services”, Volume 6: „Speech to Text and Text to Speech”, <http://www.redbooks.ibm.com/abstracts/sg248388.html>
- „Building Cognitive Applications with IBM Watson Services”, Volume 7: „Natural Language Understanding”, <http://www.redbooks.ibm.com/abstracts/sg248398.html>

13.8. Podsumowanie

Ten rozdział stanowi wprowadzenie do platformy przetwarzania poznawczego (*cognitive computing*), jaką jest IBM Watson. Oferuje ona szeroką gamę usług, przeznaczonych do integrowania z tworzonymi aplikacjami. Zasadniczo IBM Watson jest platformą komercyjną, firma IBM zachęca jednak również do nauki i eksperymentowania ze swoją platformą, oferując wiele z jej usług, z pewnymi ograniczeniami, w ramach darmowej warstwy *Lite tier*. Usługi IBM Watson koncentrują się wokół sztucznej inteligencji, między innymi przetwarzania języka naturalnego, tłumaczenia tekstów między językami, konwersji mowy na tekst i tekstu na mowę, identyfikowania znaczeń tekstów, analizy ich sentymentów, rozpoznawaniu obrazów i innych wzorców — i wielu innych aspektów rzeczywistości kojarzonych z inteligentnym działaniem.

Dostęp do usług i narzędzi platformy Watson wymaga założenia konta *IBM Cloud*. Jego właściciel otrzymuje dostęp do bogatej dokumentacji, tutoriali, wskazówek przykładowego kodu i oczywiście możliwość korzystania z usług nakreślonych warunkach (w tym warstwy *Lite tier*).

W języku Python programistyczny dostęp do usług platformy IBM Watson możliwy jest dzięki pakietowi *Watson Developer Cloud Python SDK*. Łącząc odpowiednio kilka usług, skonstruowaliśmy aplikację-mashup ułatwiającą konwersację dwojga rozmówców, z których jeden posługuje się tylko językiem angielskim, a drugi tylko językiem hiszpańskim. Pierwszy rozmówca zadaje pytanie w języku angielskim, a usługa *Speech to Text* konwertuje jego wypowiedź na tekst pisany. Usługa *Language Translator* tłumaczy angielski tekst na odpowiednik w języku hiszpańskim, a usługa *Text to Speech* na podstawie tego tekstu syntetyzuje mowę, dzięki czemu drugi rozmówca słyszy odnośne pytanie w języku hiszpańskim. I w tym języku udziela odpowiedzi, po czym wymieniony łańcuch usług doprowadza do wyemitowania głosu wypowiadającego tę odpowiedź w języku angielskim.

Platforma IBM Watson jest niezwykle bogato udokumentowana, towarzyszą jej także liczne zasoby: blogi, filmy i wideoklipy, rozległe repozytorium kodu w serwisie GitHub oraz „czerwone książki” IBM Redbooks.

PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Python® dla programistów

Możliwości

- Ponad 500 praktycznych przykładów ze świata rzeczywistego do uruchamiania i eksperymentowania — od krótkich fragmentów po kompletne analizy przypadków
- Kodowanie w interpreterze IPython i notatnikach Jupytera
- Wykorzystywanie bibliotek: standardowej biblioteki Pythona i bibliotek Data Science, do realizacji złożonych zadań przy minimalnym udziale kodowania
- Obszerny opis języka Python: instrukcje sterujące, funkcje, łańcuchy znaków, pliki, serializacja w notacji JSON, pliki i wyjątki
- Programowanie proceduralne, w stylu funkcyjnym i zorientowane obiektowo
- Kolekcje: listy, krotki, słowniki, zbiory, tablice NumPy, szeregi i ramki danych Pandas
- Wizualizacje — statyczne, dynamiczne i animowane
- Doświadczenia z danymi pochodzącymi z rzeczywistego świata
- Wprowadzenie do Data Science: sztuczna inteligencja, podstawowe statystyki, symulacje, animacje, zmienne losowe, preparacje danych do analizy, regresja
- Analizy przypadków: sztuczna inteligencja, Big Data, Data Science w chmurach: NLP, eksploracja Twittera, IBM Watson™, uczenie maszynowe, głębokie uczenie, widzenie komputerowe, Hadoop, Spark™, NoSQL, IoT
- Biblioteki open source: NumPy, Pandas, Matplotlib, Seaborn, Folium, SciPy, NLTK, TextBlob, spaCy, Textastic, Tweepy, scikit-learn™, Keras i wiele innych

Kontakt z autorami

- e-mail: deitel@deitel.com
- media społecznościowe:
LinkedIn®: bit.ly/DeitelLinkedIn
Facebook®: facebook.com/DeitelFan
Twitter®: twitter.com/deitel
YouTube™: youtube.com/DeitelTV
- kod źródłowy i aktualizacje:
deitel.com

informit.com/deitel

Dietlowy przewodnik po Pythonie dla profesjonalistów

Książka, napisana z myślą o programistach posiadających doświadczenie w programowaniu w języku wysokopoziomym, jest podręcznikiem do tworzenia aplikacji z zakresu najnowszych, fascynujących technologii za pomocą języka Python — jednego z najpopularniejszych i najszybciej rozwijających się współczesnych języków programowania. Więcej szczegółów podano w przedmowie do książki.

Na podstawie ponad 500 przykładów zaczerpniętych z rzeczywistego świata — od małych fragmentów kodu po 40 obszernych skryptów i kompletnych implementacji analiz przypadków — wykorzystując interaktywny interpreter IPython i notatniki Jupytera szybko nauczysz się programować z użyciem najnowszych idiomów Pythona. Po lekturze części wprowadzającej złożonej z rozdziałów 1 – 5 oraz niektórych fragmentów rozdziałów 6 i 7, przejdiesz do ćwiczeń praktycznych z rozdziałów 11 – 16, obejmujących różne dziedziny współczesnych zastosowań informatyki i sztucznej inteligencji: przetwarzanie języka naturalnego, eksplorację danych Twittera® między innymi w celu analizy sentymentu tweetów, przetwarzanie poznawcze z udziałem platformy IBM® Watson™, nadzorowane uczenie maszynowe w zastosowaniu do klasyfikacji i regresji, nienadzorowane uczenie maszynowe na przykładzie klasteryzacji, widzenie komputerowe oraz realizację głębokiego uczenia za pomocą sieci neuronowych — konwolucyjnych i rekurencyjnych. Poznasz także współczesną infrastrukturę przetwarzania Big Data: platformy Hadoop® i Spark™, bazy danych NoSQL i internet rzeczy (IoT). Zyskasz też możliwość praktycznego poeksperymentowania (bezpośrednio lub pośrednio) z kilkoma usługami chmurowymi, między innymi Google Translate®, IBM Watson, Microsoft® Azure®, OpenMapQuest i PubNub.

Paul Deitel i Harvey Deitel, założyciele Deitel & Associates, Inc., to znani na całym świecie autorzy publikacji i kursów z zakresu języków programowania. Miliony użytkowników zdobywa wiedzę i doświadczenie dzięki ich książkom (drukowanym i e-bookom), lekcjom wideo LiveLessons, treningom online w serwisie Safari (<https://www.safaribooksonline.com/search/?query=deitel>) oraz interaktywnym kursom multimedialnym Revel™ doskonałym umiejętności programowania w popularnych językach, takich jak Python, Java, C++, C, C#, na popularnych platformach (Swift™, iOS, Android™) oraz tworzenia aplikacji internetowych i webowych.

KOMENTARZE RECENZENTÓW

Znakomite wprowadzenie do Pythona! Gorąco polecam tę książkę jako wprowadzenie zarówno do Pythona, jak i do Data Science — Shyamal Mitra, starszy wykładowca, University of Texas at Austin

Rozdział o IBM Watson jest fascynujący. Przykłady integrujące ze sobą różne usługi Watsona są naprawdę świetnie pomyślane — Daniel Chen, specjalista Data Science, Lander Analytics

Zajmujące i interesujące przykłady z rzeczywistego świata zachęcają czytelnika do przeprowadzania konstruktywnych analiz danych. Wyjaśnienie podstawowych koncepcji Data Science — najlepsze, jakie kiedykolwiek napotkałem. Wysoce użyteczne wprowadzenie do modeli uczenia maszynowego, wskazówki dotyczące wyboru najlepszego modelu i wyjaśnienie, co to znaczy „najlepszy”. Bardzo dobry opis technologii związanych z Big Data, z reprezentatywnymi przykładami — Jamie Whitacre, niezależna konsultantka Data Science

Wspaniałe wprowadzenie do głębokiego uczenia — Alison Sanchez, Uniwersytet San Diego

Najlepsza, jaką kiedykolwiek widziałem, książka wprowadzająca w tematykę Pythona i Data Science — Roland DePratti, adiunkt informatyk, Centralny Uniwersytet Stanowy Connecticut

Lubię tę nową kombinację informatyki, Data Science i statystyki — Lance Bryant, Uniwersytet Shippensburg

Podjęcie zastosowane w książce z pewnością zainteresuje czytelników. Fantastyczne wprowadzenie w różne koncepcje uczenia maszynowego, bez zbytniego angażowania czytelnika w szczegóły matematyczne — Garrett Dancik, profesor nadzwyczajny informatyki i bioinformatyki, Uniwersytet Stanowy Eastern Connecticut

(ciąg dalszy wewnątrz okładki)



helson.pl

HELION SA
ul. Kościuszki 10
44-100 Gliwice
tel. 32 220 98 83
helson@helson.pl

INFORMATYKA W NAJLEPSZYM WYDANIU

KOD KORZYŚCI
Sięgnij po więcej!



Cena: 129,00 zł

ISBN 978-83-283-6394-6



9 788328 363946

 Pearson