Chatboty
które działają!

Praktyczny przewodnik po konwersacyjnej sztucznej inteligencji

Andrew Freed
Enikő Rózsa
Cari Jacobs

Przedmowa: Jesus Mantas

Helion

Tytuł oryginału: Effective Conversational AI: Chatbots that work

Tłumaczenie: Piotr Rajca

ISBN: 978-83-289-3600-3

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
helion.pl/user/opinie/chatbo
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

Spis treści

2 Tworzenie konwersacyjnej sztucznej inteligencji

Zagadnienia omawiane w tym rozdziale

- Budowa konwersacyjnej sztucznej inteligencji typu FAQ.
- Budowa konwersacyjnej sztucznej inteligencji opartej na procesach.
- Wykorzystanie generatywnej sztucznej inteligencji w systemach konwersacyjnych.

W środowisku produkcyjnym konwersacyjna sztuczna inteligencja może być dość złożona. W tej książce omówimy wiele technik, które pozwalają rozwiązać rzeczywiste problemy, z jakimi możesz się spotkać podczas budowania i wdrażania własnych rozwiązań. W tym rozdziale zbudujemy Cake Bot — system konwersacyjnej sztucznej inteligencji łączący elementy kilku różnych typów botów konwersacyjnych. Zapewni nam to solidne podstawy do zrozumienia struktury systemów konwersacyjnych.

Będziemy śledzić losy fikcyjnej małej piekarni z Ohio o nazwie Cake Shop. Firma produkuje torty na zamówienie i przyjmuje zlecenia z dostawą lub odbiorem osobistym. Właściciele firmy chcą dodać do swojej strony internetowej system konwersacyjnej sztucznej inteligencji, który pomoże ich klientom. Ponieważ nigdy wcześniej nie budowali bota, zamierzają zacząć od czegoś prostego,

ale mają nadzieję szybko rozszerzyć zakres i możliwości swojego rozwiązania. Postanawiają rozpocząć od systemu, który odpowiada na najczęściej zadawane pytania.

Wiele zadań z tego rozdziału *można by* wykonać za pomocą dużych modeli językowych. Jednak właściciele tej piekarni są ostrożni. Szczególnie zależy im na kontroli sposobu formułowania odpowiedzi dla pytania kilku określonych typów. Dlatego ich rozwiązanie będzie łączyć techniki tradycyjne i generatywne.

W tym rozdziale pokażemy proces budowania z wykorzystaniem platformy konwersacyjnej sztucznej inteligencji (IBM watsonx Assistant), a później dołączymy do tworzonego rozwiązania platformę generatywnej sztucznej inteligencji (IBM watsonx.ai). Kluczowe koncepcje przedstawione w tym rozdziale można zastosować na wielu różnych platformach sztucznej inteligencji. Nic także nie stoi na przeszkodzie, byś do tworzenia swoich rozwiązań używał innej wybranej przez siebie platformy konwersacyjnej i generatywnej AI.

2.1. Tworzenie bota FAQ

Większość twórców konwersacyjnej sztucznej inteligencji zaczyna od botów odpowiadających na pytania. Te rozwiązania AI, znane również jako boty FAQ, udzielają odpowiedzi bezpośrednio na pytanie użytkownika, często bez zadawania pytań uzupełniających. Użytkownik zadaje pytanie, bot zwraca odpowiedź, a rozmowa kończy się, gdy użytkownik przestaje zadawać pytania. Te boty sprawdzają się szczególnie dobrze, gdy mamy do czynienia z niewielką liczbą (często zadawanych) pytań.

W tym podrozdziale zbudujemy bota FAQ dla piekarni Cake Shop. Niektóre pytania będą miały statyczną odpowiedź, która będzie taka sama niezależnie od sposobu zadania pytania. Z kolei odpowiedzi na inne pytania będą dynamiczne — ich treść będzie się zmieniać w zależności od informacji zawartych w pytaniu. Zanim jednak wytrenujemy naszego bota, najpierw przygotujemy podstawową strukturę rozwiązania.

2.1.1. Podstawy bota FAQ

Każda konwersacyjna sztuczna inteligencja musi umieć rozpocząć rozmowę i zareagować, gdy nie wie, co robić. Większość platform konwersacyjnej AI zapewnia tę funkcjonalność domyślnie podczas tworzenia nowego chatbota. Warto szybko sprawdzić te konfiguracje i dostosować je do swoich potrzeb.

Firma Cake Shop rozpoczyna budowanie swojej konwersacyjnej AI (którą dalej będziemy nazywać „asystentem”) i nadaje jej nazwę „Cake Bot”. Z głównego menu konwersacyjnej AI ich programista przechodzi do sekcji *Actions* (Akcje), która wyświetla wszystkie możliwości asystenta. Pierwsza lista nosi tytuł *Created by you* (Utworzone przez Ciebie) i jest pusta; druga lista to *Created*

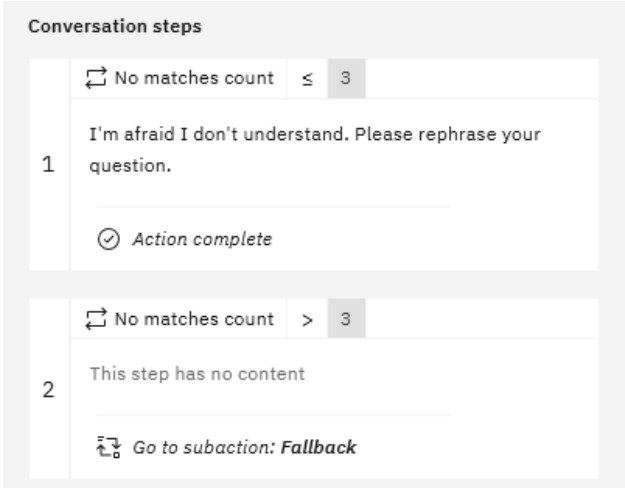
by assistant (Ustawione przez asystenta) i zawiera domyślne możliwości przedstawione w tabeli 2.1.

Tabela 2.1. *Domyślne możliwości w nowym asystencie*

Możliwość	Wykonywana, gdy
Greet customer (Powitanie klienta)	Asystent jest po raz pierwszy otwierany lub uruchamiany. Otwarcie asystenta rozpoczyna rozmowę.
No action matches (Brak dopasowania akcji)	Żadna akcja nie może zostać dopasowana do wiadomości użytkownika (wiadomość nie jest zrozumiana). Inne platformy mogą nazywać to „intencją awaryjną”.
Trigger word detected (Wykryto słowo wyzwalające)	Wykryte zostają słowa kluczowe, takie jak wulgaryzmy.
Fallback (Awaryjna)	Użytkownik musi opuścić chatbota.

Dostosowanie pierwszej z tych możliwości jest najważniejsze, ponieważ daje nam pierwszą szansę na personalizację asystenta. Domyślny tekst brzmi *Welcome, how can I assist you?* (Witamy, jak mogę Ci pomóc?). Zespół Cake Shop zmienia ten tekst na „Witam w Cake Bot. Jak mogę Ci pomóc?”. To minimalny poziom dostosowania — lepiej byłoby dołączyć dodatkowe informacje, takie jak to, co bot może zrobić dla użytkowników. Jednak bot nie ma jeszcze żadnych możliwości, więc zespół Cake Shop pozostawia tę wiadomość bez zmian.

Następnie należy przejrzeć akcję *No action matches* (Brak dopasowania akcji). Ta akcja zostanie wywołana, gdy bot nie zrozumie użytkownika. Ponieważ bot nie został jeszcze w żaden sposób wytrenowany, ta akcja będzie domyślną odpowiedzią na każde dane wejściowe wpisane przez użytkownika. Domyślna konfiguracja jest pokazana na rysunku 2.1.



Rysunek 2.1. *Domyślna konfiguracja akcji No action matches*

Powyższa konfiguracja jest następująca:

1. Akcja zlicza, ile razy została wywołana podczas konwersacji.
2. Jeśli trzy razy lub mniej, domyślna odpowiedź brzmi: *I'm afraid I don't understand. Please rephrase your question* (Obawiam się, że nie rozumiem. Proszę przeformułować pytanie).
3. Jeśli cztery razy lub więcej, system przekierowuje do procedury awaryjnej (przy czym domyślną procedurą awaryjną jest zaproponowanie kontaktu z konsultantem).

Zespół Cake Shop postanawia obniżyć ten próg, zmieniając 3 na 1. Dzięki temu użytkownicy na pewno nie utkną w pętli nieporozumień.

Procedura awaryjna i połączenie z konsultantem

Większość platform konwersacyjnej sztucznej inteligencji oferuje bardzo proste sposoby integracji, wymagające napisania minimalnego kodu lub takie, w których pisanie kodu w ogóle nie jest konieczne (określane, odpowiednio, integracjami *low-code* i *no-code*), umożliwiające połączenie użytkowników z konsultantem przez czat lub rozmowę głosową. Nie będziemy tutaj opisywać szczegółowo tych rozwiązań, ponieważ zależą one od konkretnej platformy. Wystarczy powiedzieć, że jest to powszechnie znany wzorzec. W tym rozdziale skoncentrujemy się na projektowaniu konwersacji i trenowaniu sztucznej inteligencji.

W tym momencie mamy już chatbota, który wykonuje trzy czynności:

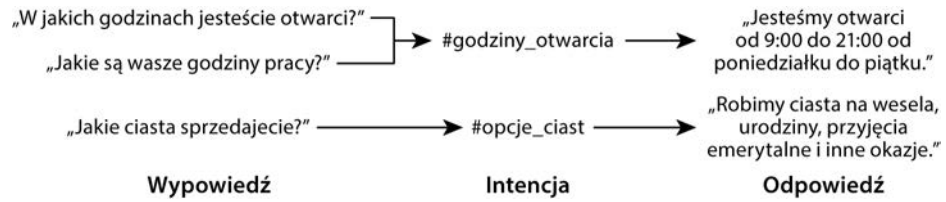
1. Gdy użytkownik otwiera czat, wita go komunikatem „Witam w Cake Bot. Jak mogę Ci pomóc?".
2. Bez względu na to, co użytkownik napisze następnie, chatbot odpowiada, że nie rozumie.
3. Niezależnie od tego, co użytkownik na to odpowie, chatbot oferuje kontakt z prawdziwym konsultantem.

Nuda! Nauczmy zatem tego bota odpowiadać na pytania jak należy.

2.1.2. Statyczne pytania i odpowiedzi

Zacznijmy od koncepcyjnego modelu komponentów chatbota zaangażowanych w odpowiadanie na pytania.

Na niektórych platformach można bezpośrednio łączyć pytania z odpowiedziami. Na innych wprowadza się dodatkową warstwę, która kategoryzuje podobne pytania w grupy zwane **intencjami** (ang. *intent*). System odpowiadania na pytania oparty na intencjach daje twórcy pełną kontrolę nad odpowiedziami generowanymi przez konwersacyjną AI. Uogólniona wersja tego projektu, bazująca na przykładzie chatbota Cake Bot, została pokazana na rysunku 2.2.



Rysunek 2.2. *Boty odpowiadające na pytania odwzorowują wypowiedzi użytkowników na intencje, które z kolei są odwzorowywane na odpowiedzi*

Przejrzyjmy terminologię zastosowaną na tym diagramie:

- **Wypowiedź** — to dane wejściowe przekazane do chatbota. W przypadku bota odpowiadającego na pytania są to pytania.
- **Intencja** — to logiczne grupowanie wypowiedzi o podobnym znaczeniu.
- **Odpowiedź** — to wynik wygenerowany przez chatbota. W przypadku bota odpowiadającego na pytania są to właśnie odpowiedzi.

W przypadku Twojego pierwszego chatbota intencje pozwalają zaoszczędzić dużo czasu. Zwróć uwagę, że jako twórca nie musisz rozróżniać pytań o podobnym znaczeniu. „W jakich godzinach jesteście otwarci?" i „Jakie są wasze godziny pracy?" odnoszą się do godzin działania Twojej firmy. Nie jest istotne, aby bot rozróżniał te pytania. Dzięki zastosowaniu intencji `#godziny_otwarcia` nadajemy im to samo „znaczenie". Pytanie „Jakie ciasta sprzedajecie?" ma inne znaczenie i dlatego kojarzymy je z inną intencją:`#opcje_ciast`.

Dla każdej intencji obsługiwanej przez Twojego bota system jest trenowany za pomocą przykładowych wypowiedzi. Nowoczesne systemy oparte na intencjach wymagają zaledwie pięciu przykładowych wypowiedzi na intencję. To całkiem niezły kompromis — istnieje niemal nieskończona liczba sposobów zapytania o godziny otwarcia sklepu, a podając kilka przykładów, możesz dobrze wytrenować swojego bota.

Systemy odpowiadające na pytania bazujące na intencjach to jednocześnie błogosławieństwo i przekleństwo: dla każdej intencji, którą trenujesz, możesz kontrolować odpowiedź, co niesie ze sobą zarówno zalety, jak i wady.

Zalety:

- Masz pełną kontrolę nad projektem odpowiedzi. Możesz ją redagować, formatować tekst, a nawet dodawać elementy graficzne — innymi słowy, dokładnie znasz treść odpowiedzi.
- W przypadku niewielkiej liczby intencji można to zrobić szybko. Swojego pierwszego chatbota możesz skonfigurować w zaledwie godzinę.

Wady:

- Wraz ze wzrostem liczby intencji wytrenowanie chatbota tak, by rozpoznawał je wszystkie, staje się coraz trudniejsze.

- Trudno jest dostosować odpowiedzi do niuansów w pytaniach użytkowników. Na pytanie „Czy jesteście dziś otwarci?” bot nadal odpowiada ogólnikowo: „Jesteśmy otwarci codziennie”.
- Niedokładne lub źle dostrojone odpowiedzi wywołują u użytkownika przykre wrażenie, że chatbot go nie rozumie.

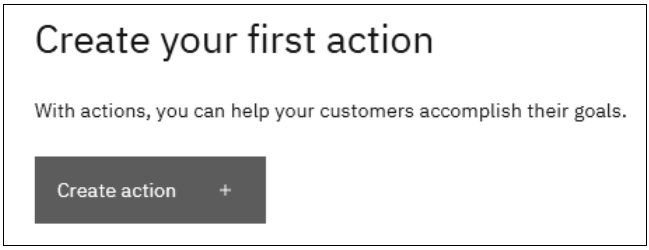
W kolejnych rozdziałach zajmiemy się kilkoma problemami związanymi z botami odpowiadającymi na pytania: jak zebrać odpowiednie dane do trenowania bota (rozdział 4.), jak wykorzystać te dane do tworzenia lepszych intencji (rozdział 5.), jak uzupełnić te intencje odpowiedziami pochodzącymi z dokumentów oraz tworzonymi przez generatywną AI (rozdział 6.) oraz jak wykorzystać generatywną AI do realizacji kilku dodatkowych zadań związanych z trenowaniem i testowaniem swoich rozwiązań (rozdział 7.).

Zacznijmy od nauczenia naszego chatbota pierwszych umiejętności odpowiadania na pytania. Dla każdej z tych umiejętności potrzebujemy intencji, zestawu powiązanych wypowiedzi użytkownika oraz odpowiedzi. Pierwszy zestaw pytań i odpowiedzi będzie dotyczył historii Cake Shop, godzin otwarcia sklepów, rodzajów oferowanych tortów, przybliżonych kosztów tortów oraz informacji o Klubie Tortowym. Te odpowiedzi bazujące na intencjach pokazano w tabeli 2.2.

Tabela 2.2. *Początkowy zestaw intencji dla bota odpowiadającego na pytania wraz z powiązanymi wypowiedziami i odpowiedziami*

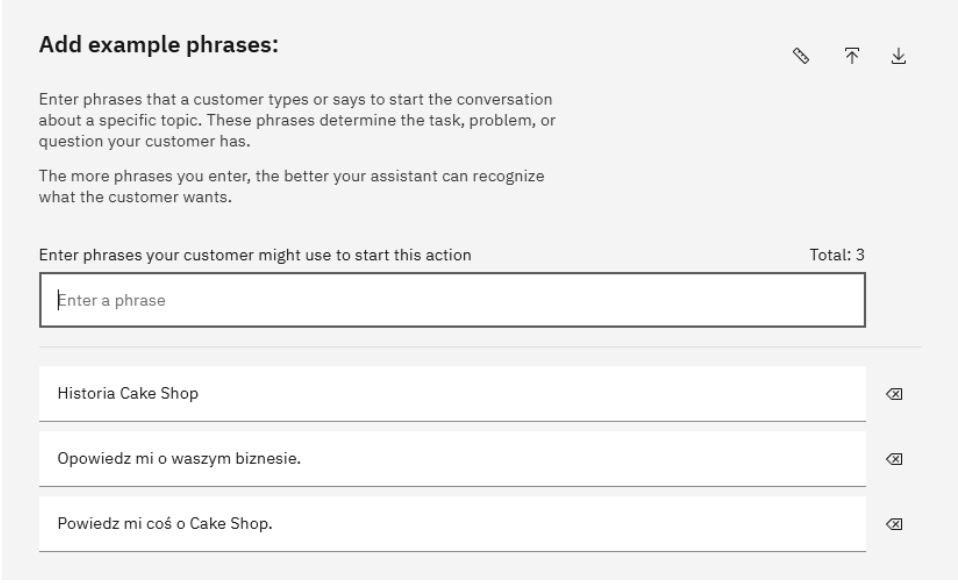
Intencja	Przykładowe wypowiedzi	Odpowiedź
`#historia`	Historia Cake Shop Opowiedz mi o waszym biznesie. Powiedz mi coś o Cake Shop.	Założona przez Babcię Cake w 1980 roku, zrobiliśmy już ponad 10 000 tortów dla lokalnych mieszkańców!
`#godziny_otwarcia`	Godziny otwarcia Jakie są wasze godziny otwarcia? Kiedy jesteście otwarci?	Jesteśmy otwarci od poniedziałku do piątku, od 9:00 do 21:00.
`#opcje_tortów`	Opcje tortów Czy robicie torty weselne? Jakie rodzaje tortów sprzedajecie?	Oferujemy torty na różne okazje, takie jak wesela, urodziny, rocznice, emerytury i torty na każdą okazję.
`#koszt`	Ile kosztuje tort? Czy jest minimalna wartość zamówienia? Czy trzeba dopłacać za dostawę?	Nasze torty kosztują zazwyczaj około 30 USD, a opłata za dostawę wynosi 5 USD.
`#klub_tortowy`	Nagrody tortowe Klub Tortowy Jakieś specjalne promocje lub zniżki?	Nasz program lojalnościowy Klub Tortowy daje Ci bon podarunkowy o wartości 10 dolarów za każde dziesięć kupionych tortów.

W asystencie definiujemy akcję, która wykrywa intencję i udziela odpowiedzi — akcję odpowiadania na pytania. To najprostszy rodzaj akcji, jaki możemy zdefiniować na dowolnej platformie konwersacyjnej sztucznej inteligencji. Rysunek 2.3 pokazuje interfejs użytkownika, który rozpoczyna definiowanie akcji.



Rysunek 2.3. *Interfejs użytkownika rozpoczynający tworzenie pierwszej akcji*

Dla każdej z takich akcji musimy skonfigurować sposób uruchamiania (wypowiedzi użytkownika) i to, co robi (odpowiada). Zauważysz, że informacje te odpowiadają środkowej oraz prawej kolumnie z tabeli 2.2. Niektóre platformy konwersacyjnej sztucznej inteligencji używają również etykiety intencji dla akcji; w naszym przypadku etykiety te zostały określone na podstawie jednej lub kilku wypowiedzi wyzwalających wykonanie danej akcji. Rozpoczynamy naszą podróż po definiowaniu wypowiedzi wyzwalających akcję `#historia`, której sam początek został przedstawiony na rysunku 2.4.

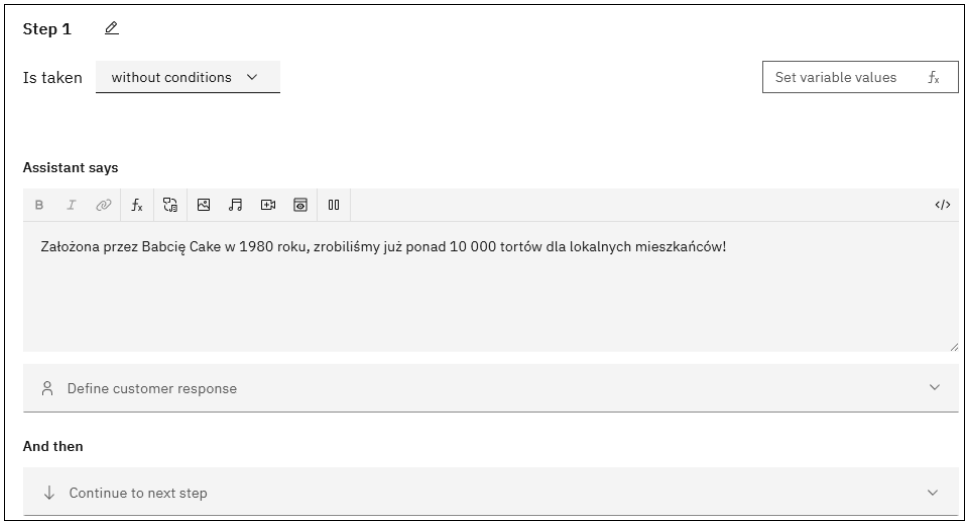


Rysunek 2.4. *Definiowanie wypowiedzi uruchamiających akcję*

Zauważ, że interfejs użytkownika wskazuje, że rozpoznawanie tej akcji przez chatbota poprawi się wraz z większą liczbą przykładów. Na potrzeby naszej demonstracji użyjemy trzech przykładów na akcję, co na początek powinno wystarczyć. W kolejnych rozdziałach pokażemy różne sposoby znajdowania dodatkowych przykładów treningowych.

Nasza akcja odpowiadania na pytania jest prawie gotowa. Mamy już pytania, które ją uruchamiają; teraz musimy zdefiniować odpowiedź chatbota. Odpowiedź dla naszej akcji `#historia` została przedstawiona na rysunku 2.5. Ta akcja składa się z trzech części, którymi są:

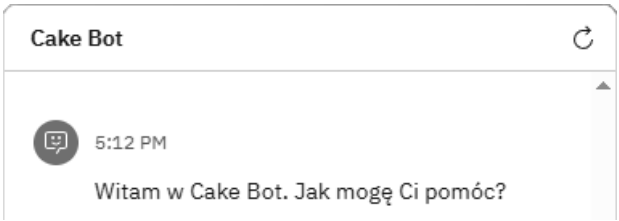
- *Logika warunkowa* — w przypadku statycznej akcji odpowiadania na pytania nie jest potrzebna żadna logika. Akcja będzie wykonana tylko wtedy, gdy zostanie wykryta intencja.
- *Odpowiedzi* — pole *Assistant says* (Asystent mówi) to odpowiedź dla użytkownika. W naszym przypadku jest to zwykły tekst.
- *Następny krok* — w przypadku statycznej akcji odpowiadania na pytania nie jest potrzebny następny krok. Udzielenie odpowiedzi kończy akcję.



Rysunek 2.5. *Definiowanie odpowiedzi dla akcji odpowiadania na pytania. Najprostsza forma ma tylko jeden krok po wykryciu intencji — udzielenie odpowiedzi*

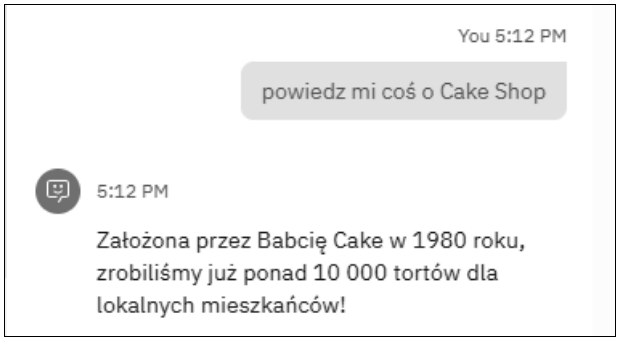
Te kroki tworzenia akcji trzeba powtórzyć dla każdej z pięciu intencji. Wszystkie akcje są trenowane na przykładach, które je uruchamiają, oraz na odpowiedzi, której chatbot powinien udzielić. Każda z nich składa się tylko z jednego kroku i kończy się po udzieleniu odpowiedzi.

Po zdefiniowaniu wszystkich pięciu akcji będziemy gotowi do przetestowania naszego chatbota. Interfejs, który do tego służy, został przedstawiony na rysunku 2.6.



Rysunek 2.6. *Strona podglądu chatu*

Spróbujmy zatem zadać kilka pytań! Rysunek 2.7 pokazuje wyniki testów dla przykładowego pytania.



Rysunek 2.7. *Przykładowa odpowiedź systemu Cake Bot na zadane pytanie*

Zauważ, że zadane pytanie nie odpowiada dokładnie żadnemu z podanych wcześniej przykładów treningowych. Oznacza to, że bot nauczył się rozumieć znaczenie zawarte w przykładach. Wyniki zamieszczone na listingu 2.1 przedstawiają dodatkowe testy naszego bota.

Listing 2.1. Dodatkowe pytania zadawane chatbotowi Cake Bot wraz z odpowiedziami

```
Użytkownik: godziny otwarcia?
Bot: Jesteśmy otwarci od poniedziałku do piątku, od 9:00 do 21:00.
Użytkownik: dlaczego kura przeszła przez jezdnię?
Bot: Obawiam się, że nie rozumiem. Proszę inaczej sformułować pytanie.
Użytkownik: cena tortu?
Bot: Nasze torty kosztują zazwyczaj około 30 USD, a opłata za dostawę wynosi 5 USD.
```

To świetny początek dla naszego chatbota. Możemy go wytrenować na większej liczbie intencji i zwiększyć jego dokładność — wystarczy zapewnić mu więcej przykładów dla tych intencji. Przyjrzyjmy się jednak czemuś innemu.

Wszystkie akcje odpowiadania na pytania, które do tej pory stworzyliśmy, były akcjami jednoetapowymi. Użytkownik otrzymuje tę samą odpowiedź bez względu na to, o co pyta. W następnym punkcie dowiesz się, jak przekształcić taką statyczną odpowiedź w odpowiedź dynamiczną, opartą na dodatkowych informacjach.

2.1.3. Dynamiczne odpowiadanie na pytania

Cukiernia Cake Shop obecnie dysponuje sklepami położonymi w czterech lokalizacjach: Columbus, Dublin, Westerville i Grandview. Kiedy bot był tworzony po raz pierwszy, wszystkie cztery lokalizacje miały te same godziny otwarcia: od 9:00 do 21:00 w dni robocze. Sytuacja się zmieniła — sklep w Columbus musi być otwierany i zamykany godzinę wcześniej (od 8:00 do 20:00). Jedna odpowiedź chatbota nie obejmuje zatem już wszystkich sklepów. Teraz, gdy użytkownik pyta o godziny otwarcia, musimy ustalić, o który sklep chodzi. Jeśli użytkownik tego nie sprecyzuje, musimy zadać pytanie doprecyzowujące.

Przykładowa konwersacja przedstawiona na listingu 2.2 pokazuje, jak chcemy, żeby bot obsługiwał pytania o godziny otwarcia.

Listing 2.2. Przykładowa konwersacja dotycząca godzin otwarcia, zależna od lokalizacji użytkownika

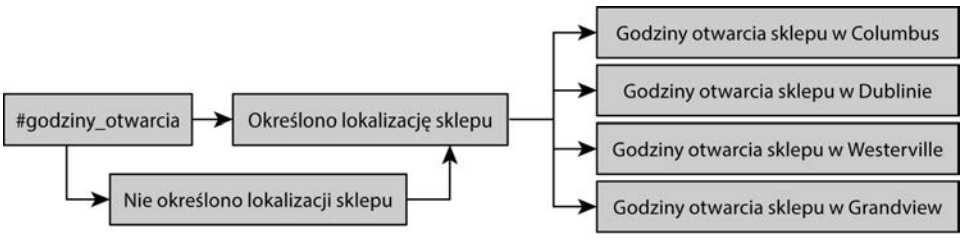
```
Użytkownik: godziny otwarcia?
Bot: Aby sprawdzić godziny otwarcia naszego sklepu, wybierz lokalizację.
Bot: (Columbus, Dublin, Westerville, Grandview)
Użytkownik: Columbus
Bot: Nasz sklep w Columbus jest otwarty od poniedziałku
    do piątku, 8:00-20:00.

Użytkownik: godziny otwarcia?
Bot: Aby sprawdzić godziny otwarcia naszego sklepu,
    wybierz lokalizację.
Bot: (Columbus, Dublin, Westerville, Grandview)
Użytkownik: Dublin
Bot: Nasz sklep w Dublin jest otwarty od poniedziałku
    do piątku, 9:00-21:00.

Użytkownik: godziny otwarcia dla Grandview?
Bot: Nasz sklep w Grandview jest otwarty od poniedziałku do piątku, 9:00-21:00.
```

Teraz niejednoznaczne pytanie jest doprecyzowywane przed udzieleniem odpowiedzi, natomiast pytanie jednoznaczne otrzymuje bezpośrednią odpowiedź

Możemy również narysować diagram przepływu obejmujący te przykładowe rozmowy. Przykład takiego diagramu został przedstawiony na rysunku 2.8. Tworzenie diagramów przepływu i przykładowych rozmów jest przydatne, gdy konwersacja ma charakter dynamiczny. Niektórzy członkowie zespołu będą preferować diagramy, inni rozmowy, a jeszcze inni będą potrzebować ich obu.



Rysunek 2.8. *Przepływ procesu dla zależnej od lokalizacji intencji #godziny_otwarcia*

Przepływ „godziny otwarcia” można zaimplementować w trzech krokach:

1. Wyświetlamy komunikat „Aby sprawdzić godziny otwarcia naszego sklepu, wybierz lokalizację” wraz z listą dostępnych lokalizacji. W odpowiedzi użytkownik musi wskazać lokalizację sklepu.
2. Jeśli krok 1. = „Columbus”, wyświetlamy godziny otwarcia sklepu w Columbus i kończymy akcję.
3. Wyświetlamy godziny otwarcia sklepu podanego w kroku 1. i kończymy akcję.

To działa, ponieważ w naszej platformie sterowanie „przepływa” pomiędzy kolejnymi krokami. A oto jak może wyglądać przebieg kilku przykładowych rozmów:

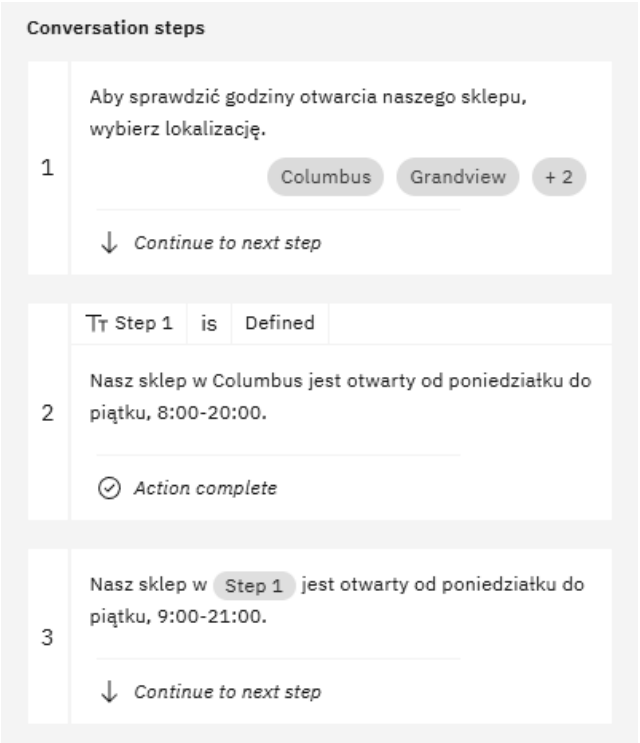
- Użytkownik wpisuje „godziny otwarcia” i zostaje uruchomiony krok 1. Użytkownik wybiera „Columbus”, zostaje uruchomiony krok 2., po czym akcja kończy działanie.
- Użytkownik wpisuje „godziny otwarcia” i zostaje uruchomiony krok 1. Użytkownik wybiera „Grandview”, warunek kroku 2. nie jest spełniony. Zostaje uruchomiony krok 3., po czym akcja kończy działanie.
- Użytkownik wpisuje „godziny sklepu w Columbus”. Warunki wyjścia z kroku 1. są spełnione, więc zostaje uruchomiony krok 2. i po czym akcja kończy działanie.
- Użytkownik wpisuje „godziny otwarcia sklepu w Grandview”. Warunki wyjścia z kroku 1. są spełnione, a warunek kroku 2. nie jest spełniony. Zostaje uruchomiony krok 3., po czym akcja kończy działanie.

Rysunek 2.9 pokazuje sposób implementacji tych kroków w naszym asystencie.

To doskonały początek prac nad naszym chatbotem Cake Bot. Bot potrafi odpowiadać na podstawowe pytania o Cake Shop i wykazuje się nawet dynamicznym charakterem. Właścicielka nie będzie musiała odpowiadać przez telefon na tyle powtarzających się pytań! Jednak Cake Bot nie może jeszcze podejmować żadnych działań w imieniu użytkowników. Temu zagadnieniu przyjrzymy się dokładniej w następnym podrozdziale.

Ćwiczenia

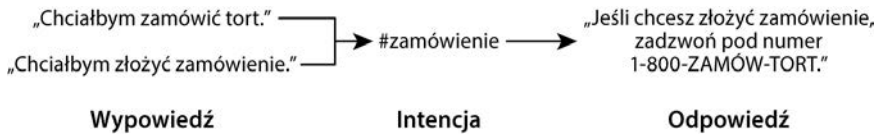
1. Pobierz kod chatbota prezentowanego w tym rozdziale książki z repozytorium w serwisie GitHub: *https://github.com/andrewrfreed/EffectiveConversationalAI*; spolonizowane przykłady można znaleźć na serwerze FTP wydawnictwa Helion: *https://ftp.helion.pl/przyklady/chatbo.zip*. Załaduj chatbota w watsonx Assistant i użyj panelu podglądu, aby przetestować przepływy pytań i odpowiedzi chatbota.
2. Alternatywnie zaimplementuj Cake Bot na preferowanej platformie konwersacyjnej AI:
   - Zdefiniuj wiadomość powitalną.
   - Zdefiniuj intencję zapasową i/lub wiadomość zapasową.
   - Zaimplementuj pięć intencji z tabeli 2.2.



Rysunek 2.9. *Trzy kroki dla akcji #godziny_otwarcia*

2.2. Agenty kierujące i boty zorientowane na procesy

Nie wszystkie boty to mechanizmy odpowiadające na pytania. Boty tego typu świetnie radzą sobie z udzielaniem odpowiedzi, ale co w sytuacji, gdy użytkownik potrzebuje czegoś więcej niż tylko odpowiedzi — co, jeśli potrzebuje, żeby bot wykonał jakąś akcję? W przypadku naszej przykładowej cukierni bardzo chcielibyśmy, żeby klienci mogli zamawiać torty bezpośrednio przez bota. Jeśli ograniczymy się wyłącznie do możliwości odpowiadania na pytania, to jedynym, co będziemy mogli zaoferować użytkownikom, będzie konwersacja taka jak ta z rysunku 2.10.



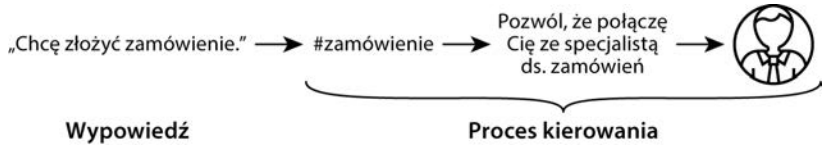
Rysunek 2.10. *Proces składania zamówień w cukierni Cake Shop jako system pytań i odpowiedzi. Ale tak naprawdę nie odpowiada na pytania!*

Użytkownik chce zrealizować proces, ale nie może tego zrobić w ramach bota. Otrzymuje jedynie *instrukcje*, jak przeprowadzić ten proces. Bot typu pytanie-odpowiedź jest więc często wczesną wersją bardziej zaawansowanego rozwiązania.

2.2.1. Agenty kierujące

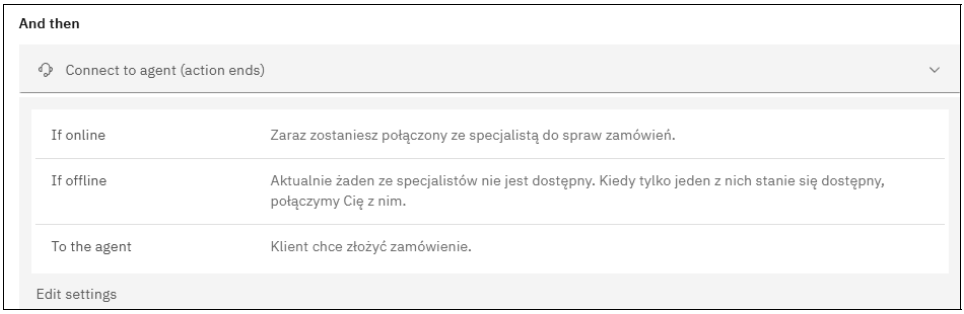
Cukiernia Cake Shop oferuje szeroką gamę tortów z różnymi opcjami smaków i dekoracji. Dostępne są pakiety dekoracyjne na wesela, ukończenie studiów, urodziny i inne okazje. Wśród opcji smakowych znajdziemy torty waniliowe, czekoladowe i truskawkowe. Do tego dochodzą różne metody płatności i dostawy. Biorąc pod uwagę wszystkie te możliwości, można założyć, że użytkownik będzie chciał lub potrzebował przedyskutować ten proces z człowiekiem.

Dla wielu twórców chatbotów następną logiczną iteracją ich bota jest agent kierujący (ang. *routing agent*). Agent kierujący wykrywa intencję zawartą w wypowiedzi użytkownika i określa, kto najlepiej może pomóc w jej realizacji. Rysunek 2.11 przedstawia naszego chatbota Cake Bot uzupełnionego o możliwości agenta kierującego.



Rysunek 2.11. *Agent kierujący wykrywa intencje użytkownika i kieruje go do odpowiedniego specjalisty*

W przypadku początkowych zapytań typu pytanie-odpowiedź bot działa tak jak wcześniej. Jednak gdy użytkownik chce zamówić tort, bot w ogóle nie próbuje odpowiedzieć na pytanie — po prostu kieruje rozmowę do odpowiedniego specjalisty. Nasze rozwiązanie zostało przedstawione na rysunku 2.12. Po określeniu intencji sama akcja składa się tylko z jednego kroku: przekierowania użytkownika do specjalisty.



Rysunek 2.12. *Konfiguracja agenta kierującego dla intencji #zamówienie. Gdy tylko intencja zostanie wykryta, użytkownik zostanie przekierowany do specjalisty*

Ten agent kierujący jedynie segreguje przychodzące zapytania, które następnie mogą zostać przekierowane do konsultantów lub do wyspecjalizowanych rozwiązań AI. Konsultanci mogą korzystać z telefonu lub czatu na stronie internetowej. W tej książce będziemy ogólnie nazywać takie osoby *agentami centrum obsługi klienta*.

Naciśnij 1, aby umówić wizytę...

Prawdopodobnie dzwoniłeś kiedyś do systemu interaktywnej odpowiedzi głosowej (IVR), który odczytuje menu opcji i prosi o wybranie jednej z nich („naciśnij 1, aby umówić wizytę"). To również jest agent kierujący. Wadą takich systemów jest długi czas potrzebny na odczytanie całego menu. Konwersacyjny agent kierujący AI pozwala wypowiedzieć swoją intencję, co jest znacznie wygodniejsze niż słuchanie długiego menu dostępnych opcji.

Agenty kierujące pozwalają wdrażać rozwiązania konwersacyjnej sztucznej inteligencji stopniowo, eliminując tym samym konieczność obsługi wszystkiego naraz.

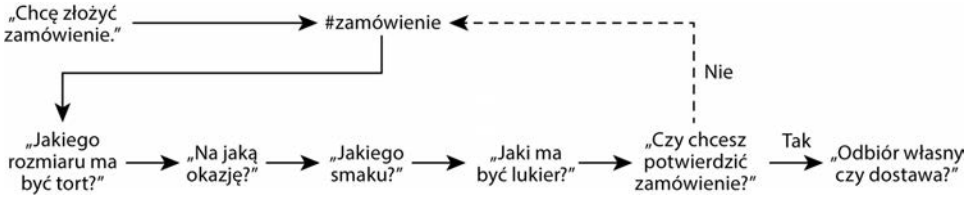
Konsultanci w systemach z agentami kierującymi często wiedzą, jakiego typu zapytanie chce zadać użytkownik, ale niewiele więcej. Na rysunku 2.12 powiedziano im tylko, że użytkownik chce zamówić tort. W przypadku niektórych procesów o wysokim stopniu złożoności i wrażliwości może to być idealne rozwiązanie. Na przykład intencja „zgłoś oszustwo" prawdopodobnie powinna od razu spowodować nawiązanie połączenia z człowiekiem.

W innych scenariuszach wczesne przekierowanie użytkownika do konsultanta jest nudne dla agenta i kosztowne dla pracodawcy. W systemach ubezpieczeniowych obsługujących statusy roszczeń przed przejściem do bardziej wartościowych zadań, takich jak wyjaśnienie, co się stało z roszczeniem, należy zebrać identyfikator członka i datę zgłoszenia roszczenia. Tutaj asystent AI mógłby najpierw zebrać identyfikator członka i datę zgłoszenia roszczenia, a dopiero potem przekierować rozmowę do człowieka.

Tak więc kolejną ewolucją agenta kierującego jest przeniesienie większej części pracy do bota. Spróbujmy zrobić to w naszym chatbocie Cake Bot.

2.2.2. Przejście od agenta kierującego do bota zorientowanego na proces

Uogólniony przepływ procesu zamawiania tortów został przedstawiony na rysunku 2.13. Obejmuje on cztery kroki wyjaśniające szczegóły dotyczące zamawianego tortu, następnie krok potwierdzenia i wreszcie realizację. (Dla zwięzłości w tej części rozdziału pominiemy szczegóły realizacji — przykładowy kod jest dostępny w serwisie GitHub, na stronie repozytorium przykładów do książki: *https://github.com/andrewrfreed/EffectiveConversationalAI*, z kolei spolonizowane przykłady można znaleźć na serwerze FTP wydawnictwa Helion: *https://ftp.helion.pl/przyklady/chatbo.zip*).



Rysunek 2.13. *Diagram procesu zamawiania tortu w Cake Shop*

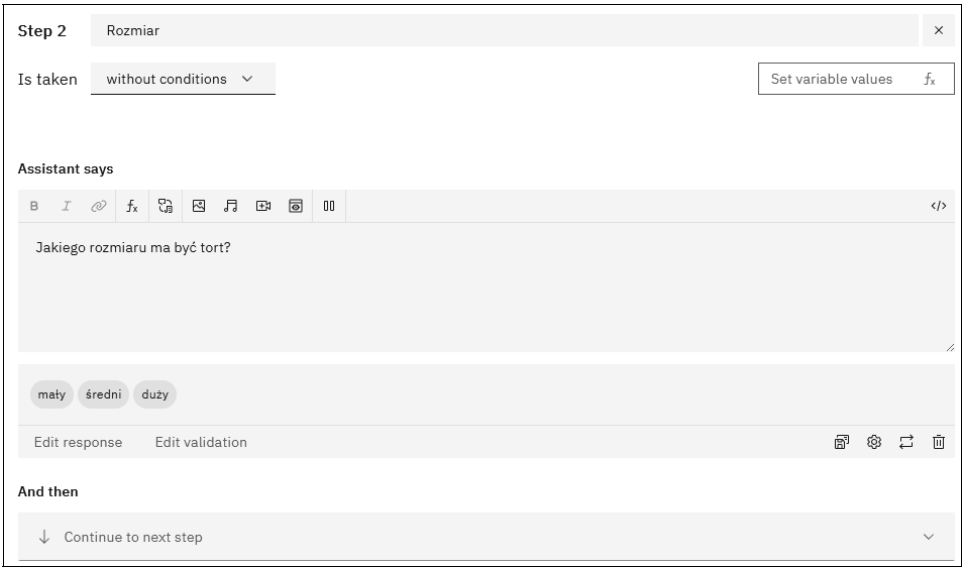
Po zaprojektowaniu pełnego diagramu procesu możemy przejść od agenta kierującego do bota zorientowanego na proces. Cake Bot będzie obsługiwał część procesu zamawiania tortu, obejmującą gromadzenie kilku szczegółowych informacji przed przekierowaniem użytkownika do ludzkiego agenta w celu dokończenia procesu. Rysunek 2.14 przedstawia projekt pierwszej wersji tego procesu Cake Bota.

Rysunek 2.14. *Przekształcanie agenta kierującego w bota zorientowanego na proces. Teraz, przed przekazaniem obsługi zamówienia do człowieka, bot zbiera dwie informacje*

Początkowo nasz proces składał się z jednego kroku (jak pokazano na rysunku 2.12), teraz są cztery:

1. Bot rozpocznie proces poprzez wyświetlenie odpowiedzi: „Mogę Ci pomóc w zamówieniu tortu".
2. Następnie bot zapyta o rozmiar tortu i przedstawi opcje do wyboru (mały, średni, duży).
3. W dalszej kolejności zapyta o okazję, z której tort będzie wręczany, i przedstawi dostępne opcje (urodziny, wesele, rocznica, emerytura, uniwersalny).
4. Bot przekieruje użytkownika do konsultanta. To jest pierwszy i jedyny krok naszego początkowego agenta kierującego, przy czym aktualnie komunikat przekazywany konsultantowi zmienił się z „Użytkownik chce zamówić tort" na „Użytkownik chce zamówić <rozmiar> tort na <okazję>". Jak widać, asystent wstawia do komunikatu odpowiedzi podane przez użytkownika w krokach 2. i 3.

Wszystkie te kroki są wykonywane sekwencyjnie. Rysunek 2.15 pokazuje krok 2.



Rysunek 2.15. *W kroku zbierania informacji o rozmiarze tortu oferowane są użytkownikowi konkretne opcje do wyboru*

Zwróć uwagę, że bot może oferować użytkownikowi różne opcje w formie przycisków, informując go w ten sposób o dostępnych możliwościach wyboru. Bot może również pozwolić użytkownikowi na własnoręczne wpisanie odpowiedzi, jeśli woli taką formę interakcji. Warto dokładniej przyjrzeć się tym opcjom, korzystając z kodu dostępnego w przykładach do książki.

Warto również zauważyć, że każda z tych opcji będzie obsługiwać „przepływ sterowania". Jeśli użytkownik rozpocznie rozmowę od słów „Chcę zamówić tort", zostanie zapytany o rozmiar i okazję. Jeśli z kolei napisze: „Chcę zamówić duży tort rocznicowy", chatbot pominie pytania o rozmiar i okazję, ponieważ informacje te zostały już podane.

Implementację pozostałej części procesu zamawiania tortu, odpowiadającą krokom opisanym w tym punkcie rozdziału, można znaleźć w przykładach do książki w serwisie GitHub. Listing 2.3 przedstawia przykładową rozmowę z naszym botem.

Listing 2.3. Przykładowa rozmowa dotycząca zamawiania tortu

```
Użytkownik: Chcę zamówić tort.
Bot: Mogę przyjąć zamówienie na tort.
Bot: Jakiego rozmiaru ma być tort?
Bot: (mały, średni, duży)
Użytkownik: duży
Bot: (urodzinowy, weselny, rocznicowy, emerytalny, uniwersalny)
Użytkownik: urodzinowy
Bot: Jaki ma mieć smak?
Bot: (waniliowy, czekoladowy, marmurkowy)
```

```
Użytkownik: waniliowy
Bot: Jaką polewę chciałbyś?
Użytkownik: truskawkową
Bot: Dobrze, zapisałem Cię na mały waniliowy tort urodzinowy
    z truskawkową polewą. Czy to się zgadza?
Użytkownik: tak
Bot: Świetnie. Twoje zamówienie zostało przyjęte.
Wartość zamówienia wynosi 30 USD.
Bot: Czy chciałbyś zaplanować odbiór osobisty, czy dostawę?
    Pamiętaj, że koszt dostawy wynosi 5 USD.
Bot: (odbiór osobisty, dostawa)
Użytkownik: dostawa
(szczegóły realizacji zostały pominięte)
```

Komunikat potwierdzający w kroku 7. odtwarza informacje zebrane w poprzednich krokach

Potwierdzenie zamówienia w kroku 8. uruchamia logikę warunkową dla ceny tortu

Nasz Cake Bot staje się coraz bardziej zaawansowany. Ma możliwość statycznego odpowiadania na pytania o torty, dynamicznego odpowiadania na pytania o godziny otwarcia sklepów oraz jest zorientowany na procesy przepływu zamawiania tortów. Zespół Cake Shop wdraża swojego chatbota i jest zadowolony z rezultatów (a użytkownicy są zadowoleni ze swoich tortów!). W następnym podrozdziale zmierzymy się z ostatnim wyzwaniem, które chcieliśmy przedstawić w tym rozdziale: dodaniem możliwości generatywnej sztucznej inteligencji poprzez integrację dużego modelu językowego (LLM).

Ćwiczenia

1. Skorzystaj z prezentowanego w tym rozdziale kodu chatbota, który pobrałeś z repozytorium książki w serwisie GitHub (*https://github.com/andrewrfreed/EffectiveConversationalAI).* Załaduj chatbota w watsonx Assistant i użyj panelu *Preview*, aby przetestować przepływ zamawiania tortów przez chatbota.
2. Alternatywnie możesz zaimplementować proces przyjmowania zamówień przez Cake Bota na wybranej przez siebie platformie konwersacyjnej AI:
   - Wykryj intencję zamówienia tortu.
   - Zbierz wszystkie cztery parametry tortu i zakończ podsumowaniem.
   - Przekieruj intencję do konsultanta.

2.3. Odpowiadanie użytkownikowi za pomocą generatywnej sztucznej inteligencji

Jak na razie nasz Cake Bot wykorzystuje wyłącznie tradycyjną technologię konwersacyjnej sztucznej inteligencji. Odpowiadanie na pytania odbywa się za pomocą klasyfikatora opartego na intencjach. Proces składania zamówień realizowany jest przez sekwencyjną serię reguł. To podejście sprawdzało się dotychczas dobrze w przypadku potrzeb sklepu Cake Shop.

Gdy zespół Cake Shop analizuje wydajność bota Cake Bot, zauważa nietypowy trend. Użytkownicy pytają bota o przepisy na dania, które zamierzają podać na obiad przed ciastem. Prośby o przepisy nie wykazują żadnego innego wzorca —

pojawiają się zapytania o zapiekanki, sałatki, dania z patelni i wiele innych. Zespół cieszy się z różnorodności swoich użytkowników, ale nie wie, jak obsłużyć te zapytania w bocie Cake Bot. Jak można wykryć wszystkie te różne rodzaje przepisów, a co dopiero na nie odpowiedzieć?

To doskonałe miejsce, aby zespół Cake Shop włączył do swojego rozwiązania elementy generatywnej sztucznej inteligencji. Mogą wykorzystać istniejący mechanizm intencji do wykrywania próśb o przepisy, a następnie przekierowywać je do dużego modelu językowego (LLM), który wygeneruje odpowiedź. Będą musieli zintegrować LLM ze swoim chatbotem i wysyłać do niego konkretne zapytania.

Zobaczmy, jak mogą to zrobić.

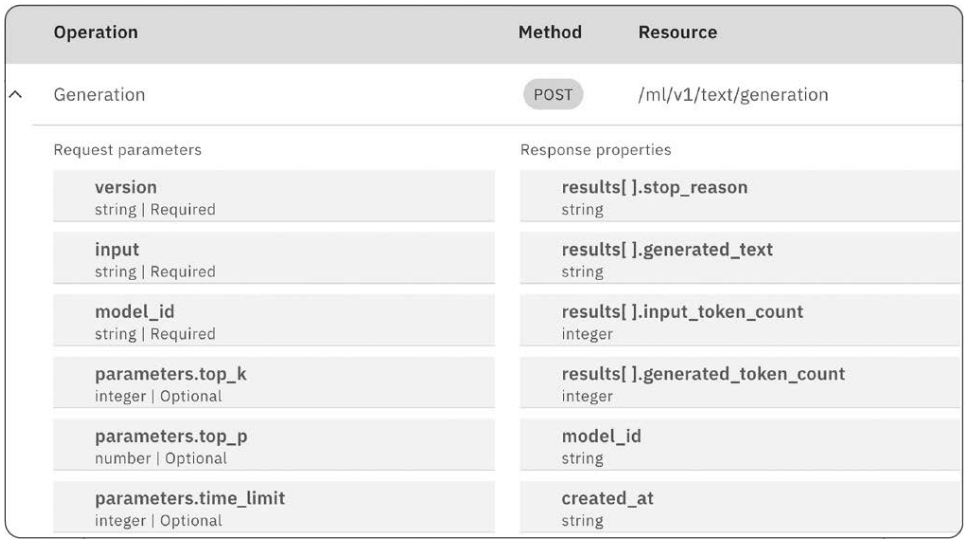
2.3.1. Integracja z dużym modelem językowym

W przypadku wielu platform konwersacyjnej sztucznej inteligencji podstawowym sposobem integracji z systemami zewnętrznymi są interfejsy programowania aplikacji (API). To powszechne wzorce integracji, które na szczęście są obsługiwane przez szeroką gamę platform generatywnej AI udostępniających duże modele językowe. Konkretny sposób integracji API z konwersacyjną sztuczną inteligencją różni się w zależności od platformy. Na niektórych platformach integracja ta odbywa się za pomocą kodu, inne oferują interfejsy typu *low-code*, które prawie nie wymagają programowania, a jeszcze inne — rozwiązania wizualne. Różne platformy mają różne nazwy dla swoich możliwości integracyjnych, takie jak rozszerzenia (ang. *extensions*), integracje (ang. *integrations*) czy realizacje (ang. *fulfillments*). Wiele z nich umożliwia integrację API za pomocą specyfikacji OpenAPI.

W tym punkcie rozdziału dodamy platformę generatywnej sztucznej inteligencji jako rozszerzenie służące do generowania tekstu i korzystające z modelu LLM. Dodanie rozszerzenia na naszej platformie wymaga wykonania czterech kroków (ich szczegółowy opis można znaleźć w przykładach dołączonych do książki):

1. Z menu *Integrations* (Integracje) wybierz opcję *Build a Custom Extension* (Zbuduj niestandardowe rozszerzenie).
2. Podaj nazwę i opis, na przykład *Wywołanie API platformy generatywnej AI*.
3. Dostarcz plik specyfikacji OpenAPI. Ten plik opisuje możliwości rozszerzenia, w tym metody, które udostępnia, wymagane i opcjonalne parametry oraz odpowiedzi, które zapewnia. Pliki specyfikacji OpenAPI to popularny format dokumentacji dla API. Są zazwyczaj dostarczane przez platformy generatywnej AI.
4. Podaj szczegóły połączenia i uwierzytelniania, takie jak adres URL implementacji API oraz klucz API potrzebny do uzyskania dostępu.

Dodajemy rozszerzenie, po czym możemy wizualnie zbadać je z wnętrza asystenta. Rysunek 2.16 pokazuje rozszerzenie dla API generowania tekstu przy użyciu modelu LLM na naszej platformie.

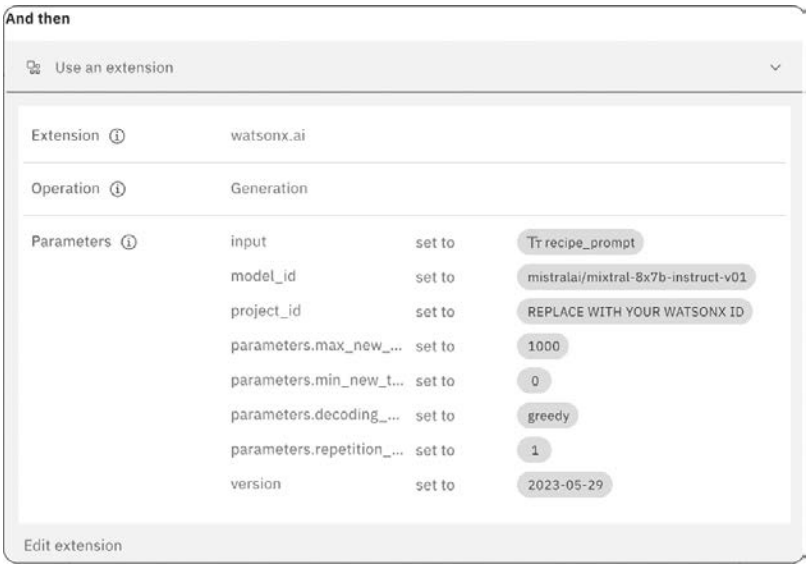


Rysunek 2.16. *Specyfikacja OpenAPI dla naszego API do generowania tekstu z użyciem modelu LLM wraz z widocznym fragmentem dostępnych parametrów żądania*

W momencie pisania niniejszej książki nasze API generowania tekstu obejmuje 15 parametrów wejściowych i 6 parametrów wyjściowych — więcej niż mieści się na rysunku 2.16! Dostępnych jest również kilka parametrów bez żadnych dostosowań, takich jak kod statusu HTTP odpowiedzi. Inne platformy generatywnej sztucznej inteligencji będą miały podobny zestaw parametrów, choć być może będą miały inne nazwy lub położenie. Przyjrzyjmy się najważniejszym parametrom:

- `input` (żądanie) — monit dla LLM; będzie zawierał instrukcje, kontekst i dane dla LLM; część tych danych może pochodzić bezpośrednio od użytkownika;
- `model_id` (żądanie) — identyfikator LLM do użycia w zadaniu; większość platform generatywnej AI pozwala wybierać spośród kilku modeli;
- `parameters` (żądanie) — pary klucz-wartość, które dostrajają zachowanie modelu LLM; obejmują one metodę dekodowania (zachłanne lub próbkujące), liczbę tokenów wyjściowych do wygenerowania oraz kilka innych parametrów;
- `generated_text` (odpowiedź) — wynik wygenerowany przez model LLM.

Możemy użyć rozszerzenia z dowolnego kroku w dowolnej akcji. Wcześniej w tym rozdziale używaliśmy możliwości/funkcji takich jak *Assistant says* (Asystent mówi), *Continue to next Step* (Przejdź do następnego kroku) oraz *Connect to Agent* (Połącz z agentem). W przypadku rozszerzeń funkcja, której należy użyć, nazywa się *Use an extension* (Użyj rozszerzenia). Rysunek 2.17 pokazuje, jak wygląda takie wywołanie rozszerzenia dla naszej akcji generowania przepisu. Inne zadania używające modelu LLM wyglądałyby podobnie, ale miałyby różne wartości konfiguracyjne. Ten zestaw parametrów jest dostosowany do generowania przepisów.

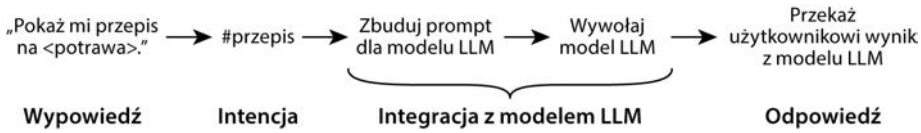


Rysunek 2.17. *Wywołanie API generowania tekstu przy użyciu modelu LLM z akcji w asyście AI*

Zobaczmy teraz, jak możemy połączyć to wszystko w chatbocie Cake Bot.

2.3.2. Kierowanie żądań do modelu LLM

Diagram przepływu przedstawiony na rysunku 2.18 pokazuje sposób generowania przepisów w naszym bocie Cake Bot. Najpierw utworzymy nową akcję. Podobnie jak w przypadku akcji odpowiadających na pytania, zaczynamy od przykładowych wypowiedzi, które spowodują wyzwolenie akcji. W naszym przykładzie pierwszymi trzema z tych wypowiedzi będą: „Pokaż mi przepis na", „Jak mogę przygotować" oraz „Powiedz mi, jak upiec". Ze względu na ogromną różnorodność możliwych przepisów nie uwzględniamy nazw potraw, a jedynie sposób, w jaki prawdopodobnie użytkownicy mogą prosić o przepisy.



Rysunek 2.18. *Diagram przepływu dla generowania przepisów w bocie Cake Bot przy wykorzystaniu modelu LLM*

Krok 1. nowej akcji polega na zapisaniu całej oryginalnej wypowiedzi użytkownika (ze zmiennej systemowej `input.text`) w zmiennej o nazwie `recipe_query_text`. To technika, której nie stosowaliśmy w poprzednich krokach. W przypadku akcji zamawiania tortów każda opcja miała jawny i skończony zestaw odpowiedzi. Nawet jeśli użytkownik powiedział „duże ciasto, proszę", chcieliśmy zapisać tylko „duże". W przypadku prośby o przepis nie mamy pojęcia, co napisze użytkownik, dlatego też rejestrujemy całą jego wypowiedź.

Krok 2. akcji polega na zdefiniowaniu promptu dla modelu LLM. Łączymy prosty prompt systemowy z żądaniem użytkownika. Następny listing — 2.4 — przedstawia wyrażenie używane do budowania zmiennej `recipe_prompt`.

Listing 2.4. Tworzenie promptu z prośbą o wygenerowanie przepisu, który jest przechowywany w zmiennej recipe_prompt

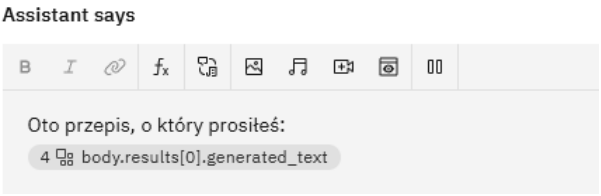
```
"Jesteś przydatnym asystentem kulinarnym. Przygotuj przepis zgodnie z instrukcjami
podanymi przez użytkownika.\n\nDane wejściowe: ".append(recipe_query_text)
↪.append("\n\nWyniki: ")
```

Etap 3. polega na wywołaniu modelu LLM. Parametry tego wywołania zostały przedstawione wcześniej na rysunku 2.17, ale przyjrzyjmy się teraz kilku wybranym spośród nich oraz ich wartościom:

- `input` — jako danych wejściowych używamy wartości zmiennej `recipe_prompt`; to powoduje wstawienie żądania przepisu podanego przez użytkownika do uogólnionego formatu promptu pokazanego na listingu 2.4;
- `model_id` — dostępnych jest wiele modeli, ale w momencie pisania niniejszej książki dobre wyniki w zadaniu generowania przepisów zapewniał model `mistralai/mixtral-8x7b-instruct-v01`;
- `project_id` — to jest identyfikator pochodzący z instancji projektu platformy generatywnej sztucznej inteligencji;
- `min_tokens` i `max_tokens` — są ustawione, odpowiednio, na `0` i `1000`, zwiększone z domyślnych wartości `0` i `200`, ponieważ przepisy zwykle bywają dość długie;
- `decoding_method` — ustawiony na `greedy` dla ułatwienia debugowania, więc model odpowiada identycznie, jeśli podane są te same dane wejściowe;

- `repetition_penalty` — ustawiony na 1 (bez kary), ponieważ przepisy w internecie zazwyczaj zawierają pewne powtórzenia;
- `version` — określa pomocniczy numer wersji żądanego API; podczas pisania niniejszej książki używaliśmy domyślnej w danym momencie, czyli: `2023-05-29`.

Krok 4. akcji polega na sprawdzeniu, czy wywołanie API zakończyło się sukcesem, a jeśli tak, to także wyświetleniu odpowiedzi użytkownikowi. W takim przypadku generujemy odpowiedź zaczynającą się od statycznego tekstu „Oto przepis, o który prosiłeś", do którego dołączamy odpowiedź zwróconą przez model LLM. Nie musimy pisać żadnego kodu parsującego dane JSON; specyfikacja OpenAPI mówi nam, że możemy odwołać się do wartości `generated_text`. Konfiguracja odpowiedzi akcji została przedstawiona na rysunku 2.19.



Rysunek 2.19. *Wyświetlanie wyniku z wywołania LLM użytkownikowi*

Budowanie chatbota Cake Bot było naprawdę bardzo zabawne! Listing 2.5 pokazuje przykładową rozmowę, którą przeprowadziliśmy z tym botem. Przepis jest długi, więc dla skrócenia listingu pominęliśmy jego znaczną część. Uruchom Cake Bota i wypróbuj go sam!

Listing 2.5. Skrócona przykładowa konwersacja z Cake Botem

```
Użytkownik: Potrzebuję szybkiego przepisu na ciasto z masłem orzechowym.
Bot: Oto przepis:
...
Składniki:
- 1 gotowy spód z kruchych ciasteczek
- 1 szklanka kremowego masła orzechowego
- ...

Przepis:
W dużej misce ubij masło orzechowe, cukier puder i miękki serek śmietankowy, aż masa
będzie gładka i kremowa.
...
Podawaj schłodzone i ciesz się szybkim oraz łatwym ciastem z masłem orzechowym!
Daj znać, jeśli potrzebujesz dodatkowych informacji lub wyjaśnień.
```

Ćwiczenia

1. Skorzystaj z kodu chatbota do tego rozdziału dostępnego w przykładach dołączonych do książki, które można pobrać z repozytorium w serwisie GitHub (*https://github.com/andrewrfreed/EffectiveConversationalAI*) lub z serwera FTP wydawnictwa Helion: *https://ftp.helion.pl/przyklady/chatbo.zip*. Załaduj chatbota do środowiska watsonx Assistant i postępuj zgodnie z instrukcjami, aby zintegrować go z platformą watsonx.ai. Użyj panelu podglądu (*Preview*), aby przetestować przepływ przepisów w chatbocie.
2. Alternatywnie możesz zaimplementować proces składania zamówień w bocie Cake Bot, korzystając z wybranych przez siebie platform konwersacyjnej i generatywnej sztucznej inteligencji:
   - Wykryj intencję dotyczącą przepisu.
   - Zbuduj prompt na podstawie zestawu instrukcji i danych wejściowych użytkownika.
   - Przekieruj odpowiedź modelu językowego do użytkownika.

Podsumowanie

- Boty odpowiadające na pytania (Q&A) to doskonały sposób na rozpoczęcie budowy pierwszej konwersacyjnej sztucznej inteligencji.
- Trenowanie botów Q&A na przykładach pytań pozwala dostarczać z góry zdefiniowane odpowiedzi na powiązane grupy pytań (intencje).
- Akcje rozpoczynają się od intencji i mogą mieć wiele rezultatów: udzielenie odpowiedzi na pytanie, przekierowanie użytkownika do konsultanta, zadawanie pytań uzupełniających oraz wykonywanie wywołań API.
- Agent kierujący identyfikuje intencje i przekazuje informacje ludziom — konsultantom. To doskonała metoda stopniowego dodawania możliwości do konwersacyjnej AI przy jednoczesnym wykorzystaniu ludzkich umiejętności.
- Konwersacyjna sztuczna inteligencja może wykorzystywać kombinację tradycyjnych technik opartych na regułach wraz z generatywną AI.

Skuteczne chatboty w praktyce

Konwersacyjna sztuczna inteligencja jest potężnym narzędziem. Dzięki chatbotom firmy usprawniają obsługę klientów i obniżają jej koszty, a pracownicy mogą się koncentrować na zadaniach wymagających większych kompetencji i budować wartość biznesową. Mimo że interfejsy konwersacyjne stały się powszechne, wiele wdrożeń nie spełnia oczekiwań użytkowników lub z czasem traci na jakości i przydatności.

W tej praktycznej książce opisano narzędzia, techniki i sprawdzone podejścia do budowania chatbotów wykorzystujących duże modele językowe i klasyczne komponenty systemów konwersacyjnych. Autorzy pokazują, jak projektować rozwiązania, które działają niezawodnie także w skali korporacyjnej. Omawiają metody rozpoznawania intencji użytkownika za pomocą modeli LLM, sposoby reagowania na nieoczekiwane dane wejściowe i technikę RAG, która pozwala zachować aktualność i trafność odpowiedzi. Duży nacisk położono na pętle informacji zwrotnej umożliwiające ciągłe doskonalenie jakości chatbotów, jak również na bezpieczną integrację generatywnej AI z istniejącymi architekturami.

Andrew Freed — jest wybitnym inżynierem i członkiem zespołu IBM Watson, w którym realizuje projekty oparte na sztucznej inteligencji.
Cari Jacobs — jest inżynierem kognitywnym i architektem rozwiązań, doradzała dziesiątkom firm z listy Fortune 500 i wdrażała u nich systemy konwersacyjnej AI.
Enikő Rózsa — pełni funkcję dyrektora technicznego w IBM Global AI & Analytics Practice. Jest również uznaną wynalazczynią — opublikowała liczne patenty z dziedziny przetwarzania języka naturalnego (NLP).

Helion
helion.pl
HELION S.A.
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej! ▶

ISBN 978-83-289-3600-3

Cena: 119,00 zł

MANNING

”

Bezcenne źródło wiedzy dla każdego, kto chce pomyślnie tworzyć i wdrażać konwersacyjne rozwiązania AI!

Marc Nehme
Microsoft

Gotowy plan działania dla tych, którzy chcą budować i mierzyć skuteczne systemy konwersacyjnej sztucznej inteligencji!

Corville Allen
Google

Wspaniały, kompleksowy przewodnik!

Sara Hines
pionierka innowacji AI