



**Politechnika Krakowska
im. Tadeusza Kościuszki**

Wydział Informatyki I Telekomunikacji

Iwona Poręba

Numer albumu: 145698

**Porównanie narzędzi do konstrukcji chatbotów
na przykładzie tematyki zbilansowanej diety**

**Comparison of frameworks for chatbot construction
on the example of a balanced diet**

**Praca magisterska
na kierunku INFORMATYKA
specjalność DATA SCIENCE**

Praca wykonana pod kierunkiem:
dr Radosław Kycia

Kraków, 2023

Spis treści

1.	Wstęp.....	4
1.1.	Motywacja.....	4
1.2.	Cel pracy.....	4
1.3.	Zakres pracy.....	4
1.4.	Struktura pracy.....	5
I.	Część teoretyczna.....	6
2.	Podstawy teoretyczne.....	7
2.1.	Przetwarzanie języka naturalnego.....	7
2.1.1.	Metody klasyczne.....	9
2.1.1.1.	Tokenizacja.....	9
2.1.1.2.	Wektoryzacja.....	13
2.1.1.3.	Analiza semantyczna.....	14
2.1.2.	Metody uczenia głębokiego.....	16
2.1.2.1.	Konwolucyjne sieci neuronowe.....	16
2.1.2.2.	Rekurencyjne sieci neuronowe.....	18
2.2.	Chatboty.....	20
2.2.1.	Definicja chatbota.....	20
2.2.2.	Ewolucja chatbotów na przestrzeni lat.....	21
2.2.3.	Zastosowanie chatbotów.....	22
2.3.	Klasyfikacja systemów dialogowych.....	23
2.3.1.	Chatboty oparte o reguły.....	23
2.3.2.	Chatboty oparte o korpus.....	23
2.3.3.	Modele hybrydowe.....	24
2.4.	Przegląd frameworków do tworzenia chatbotów.....	24
2.4.1.	Dialogflow.....	24
2.4.2.	Rasa.....	25

2.4.3.	ChatterBot.....	27
2.4.4.	IBM Watson	28
2.5.	Metryki oceny chatbota	29
II.	Część praktyczna	33
3.	Implementacja	34
3.1.	Wymagania.....	34
3.1.1.	Wymagania funkcjonalne	34
3.1.2.	Wymagania нефункционалне	38
3.2.	Projektowanie chatbotów	38
3.2.1.	Dialogflow	39
3.2.2.	Rasa	54
3.2.3.	ChatterBot.....	61
3.3.	Testowanie i porównanie chatbotów	68
3.3.1.	Wady i zalety zastosowanych frameworków	68
3.3.2.	Ocena satysfakcji użytkowników	74
4.	Podsumowanie i wnioski	88
	Literatura.....	89
	Załączniki.....	91
	Spis rysunków.....	97
	Spis tabel.....	101

Streszczenie

Celem pracy było wykorzystanie oraz porównanie różnych frameworków, przeznaczonych do tworzenia chatbotów, poprzez stworzenie własnego wirtualnego asystenta - FitBota, oferującego pomoc w zakresie zdrowego odżywiania. W części teoretycznej przybliżono podstawowe zagadnienia oraz metody związane z przetwarzaniem języka naturalnego. Część praktyczna obejmuje implementację FitBota przy pomocy trzech narzędzi, takich jak Dialogflow, Rasa oraz ChatterBot, a następnie ich testowanie oraz porównanie. Istotną częścią pracy było również przeprowadzenie ankiety, mającej na celu ocenę satysfakcji użytkowników ze stworzonego bota.

Abstract

The aim of the work was to use and compare various frameworks for creating chatbots by building own virtual assistant - FitBot, offering help in the field of healthy eating. The theoretical part presents the basic issues and related methods with natural language processing. The practical part covers the implementation of FitBot using three tools, such as Dialogflow, Rasa and ChatterBot, and then testing and comparing them. An important part of the work was also conducting a survey to assess user satisfaction with the created bot.

1. Wstęp

1.1. Motywacja

W dzisiejszych czasach obserwuje się bardzo duże zainteresowanie związane z chatbotami, które wynika z ich umiejętności prowadzenia efektywnych i spersonalizowanych interakcji oraz zdolności do udzielania pomocy użytkownikom w różnych dziedzinach życia. Wirtualni asystenci są obecnie dostępni w wielu różnych miejscach i najczęściej używanych platformach, takich jak strony internetowe wielu firm, aplikacje do wysyłania wiadomości oraz platformy społecznościowe typu Facebook czy Instagram. Można ich również spotkać w aplikacjach mobilnych i punktach obsługi klienta, a także występują oni w formie asystentów głosowych, do których należą bardzo znani Google Assistant oraz Amazon Alexa. Ogromna popularność chatbotów przyczyniła się do tego, że istnieje obecnie wiele narzędzi do samodzielnego ich tworzenia, co umożliwia dopasowywanie ich funkcjonalności do indywidualnych preferencji użytkownika.

1.2. Cel pracy

Celem pracy było porównanie wybranych frameworków do budowania wirtualnych asystentów, aby dowiedzieć się, które z nich są najbardziej efektywne i pozwolą na spełnienie określonych wymagań oraz osiągnięcie pożądaných rezultatów. Przy wykorzystaniu odpowiednich narzędzi stworzono chatbota, oferującego pomoc w zakresie prawidłowego odżywiania i zbilansowanej diety. Pomysł na zbudowanie wirtualnego asystenta, posiadającego wiedzę dotyczącą zdrowego stylu życia ma swoje uzasadnienie w rosnącym zapotrzebowaniu na dostępne i spersonalizowane porady zdrowotne w dzisiejszym dynamicznym świecie, w którym ciężko znaleźć czas i motywację do zadbania o swoją kondycję fizyczną oraz prawidłowe odżywianie.

1.3. Zakres pracy

W zakres pracy wchodzi analiza dostępnych na rynku narzędzi do tworzenia chatbotów, a następnie wyłonienie z nich trzech, celem ich późniejszego przetestowania oraz porównania. Wykorzystując wybrane frameworki zbudowano chatbota, przeznaczonego do pomocy w prowadzeniu zdrowego trybu życia. Powinien on spełniać założone wymagania funkcjonalne i нефункционалне oraz zapewniać pomoc w zakresie

wielu kwestii, związanych z dopasowaniem odpowiedniej diety dla użytkownika, zdolnością udzielania informacji dotyczących zasad zdrowego odżywiania, błędów żywieniowych oraz rodzajów diet, a także być w stanie zaproponować pełnowartościowe posiłki, bazując na indywidualnych preferencjach użytkowników. Istotną częścią pracy było opracowanie za pomocą narzędzia Google Forms ankiety, mającej na celu ocenę satysfakcji użytkowników z przeprowadzenia interakcji z chatbotem w każdym z wybranych frameworków.

1.4. Struktura pracy

Początkowym krokiem był przegląd dostępnych frameworków i wybranie z nich trzech różnych, posiadających odmienne funkcjonalności oraz przeznaczonych do tworzenia chatbotów o odrębnym przeznaczeniu. Narzędzia, jakie zostały wytypowane to Rasa, będąca konwersacyjną platformą typu open-source, Dialogflow – skierowany do użytku komercyjnego oraz ChatterBot, służący do budowania bardzo prostych botów, automatycznie generujących odpowiedzi.

Praca została podzielona na dwie części. W części teoretycznej znajduje się definicja oraz metody związane z przetwarzaniem języka naturalnego oraz charakterystyka i zastosowanie chatbotów. Część praktyczna obejmuje implementację wirtualnego asystenta – FitBota - przy pomocy trzech wybranych frameworków oraz jego testowanie i ocenę efektywności na podstawie wyników przeprowadzonej wśród użytkowników ankiety.

I. Część teoretyczna

2. Podstawy teoretyczne

2.1. Przetwarzanie języka naturalnego

Przetwarzanie języka naturalnego to z definicji „obszar badań w informatyce i sztucznej inteligencji (AI) zajmujący się przetwarzaniem języków naturalnych, takich jak angielski czy mandaryński. Przetwarzanie to obejmuje zazwyczaj tłumaczenie języka naturalnego na dane (liczby), które komputer może wykorzystać do uczenia się o świecie. A to zrozumienie świata jest czasami wykorzystywane do generowania tekstów w języku naturalnym, które odzwierciedlają to zrozumienie” [1].

Istnieje duża różnica pomiędzy językami naturalnymi a językami programowania. Języki, takie jak Java czy Python są używane do przeprowadzania obliczeń w celu umożliwienia wydawania poleceń maszynom. Nie są one przeznaczone do komunikacji ani do wymiany informacji między ludźmi, tak jak ma to miejsce w przypadku języków naturalnych. Wszelkie programy komputerowe, które tworzone są dzięki językom programowania, precyzyjnie określają sposób, w jaki maszyna powinna działać, natomiast w przypadku języków naturalnych ich przetworzenie nie jest łatwym zadaniem, ponieważ nie są one tak regularne, jak języki programowania. Niemniej jednak istnieje wiele sposobów na to, aby maszyna była w stanie przetwarzać język naturalny. Jeśli program komputerowy skutecznie przetransformuje taki język, to będzie w stanie pracować na otrzymanych komunikatach, a nawet reagować na nie w odpowiedni sposób. Jednak reakcja zwrotna będzie niesprecyzowana, co zatem pozwoli na większą dowolność w działaniu twórcom potoku przetwarzania języka naturalnego (z ang. *natural language pipeline*).

Zgodnie z definicją, „system przetwarzania języka naturalnego jest często określany jako pipeline, ponieważ zwykle obejmuje kilka etapów przetwarzania, w których język naturalny wpływa z jednej strony, a przetworzone dane wyjściowe wypływają z drugiej” [1]. Potok języka naturalnego składa się z serii kroków, które związane są z tworzeniem dowolnych modeli przetwarzania języka naturalnego. Budowanie i rozwijanie potoku jest uważane za punkt wyjścia dla każdego procesu rozwoju aplikacji NLP. Podstawowe elementy ogólnego potoku dla obecnego rozwoju systemu NLP to: pozyskiwanie danych (*Data Acquisition*), czyszczenie tekstu (*Text Cleaning*), przetwarzanie wstępne (*Pre-processing*), inżynieria cech (*Feature Engineering*), modelowanie (*Modeling*),

ocena (*Evaluation*), wdrożenie (*Deployment*) oraz monitorowanie i aktualizacja modelu (*Monitoring and Model Updating*) [9].

Pozyskiwanie danych, kluczowych dla danego zadania, stanowi pierwszy etap w procesie budowania dowolnego systemu NLP. Rzadko zdarza się, aby otrzymane przez nas dane były odpowiednio przygotowane i nie zawierały niepotrzebnych informacji, dlatego potrzebny jest **zabieg czyszczenia tekstu**. Po wyczyszczeniu danych, często mają one wiele odmian i muszą zostać przekonwertowane na odpowiednią formę. Dzieje się to na **etapie wstępnego przetwarzania**, a później ma miejsce **inżynieria cech**, w której wyodrębniane są wskaźniki najistotniejsze dla konkretnego zadania. Są one konwertowane do formatu zrozumiałego przez algorytmy modelowania. Później następuje **etap modelowania i oceny**, w którym tworzony jest jeden lub więcej modeli oraz porównywane są one za pomocą odpowiednich mierników oceny. Wybierany jest najlepszy model, a następnie jest on **wdrażany do produkcji**. Ostatecznym krokiem jest **regularne monitorowanie wydajności modelu i aktualizowanie** go, jeśli zajdzie taka konieczność, aby jego efektywność pozostała niezmienną [9].

Istnieje wiele istotnych powodów, dla których opracowanie oprogramowania zdolnego do przetwarzania języka naturalnego jest kluczowe. Pierwszą kwestią jest ogromna ilość słów wypowiedzianych każdego dnia przez ludzi na całym świecie. Bazując na danych z Wikipedii [2] na świecie występują obecnie tysiące języków programowania, natomiast jak podaje źródło Ethnologue [3] języków naturalnych jest ponad kilka razy więcej.

Kolejną ważną rzeczą jest fakt, że proces przekształcania języka naturalnego jest niezwykle ciężki. Nie jest to naturalne dla maszyn, kiedy muszą przetwarzać coś, co nie zostało stworzone dla nich do rozpoznania. Można uznać to za niewiarygodne, ponieważ do tej pory istniało przekonanie, że tylko ludzie mają zdolność do zrealizowania takiego zadania. Zdobycie cennych informacji za pomocą języka naturalnego może być wymagające, ponieważ wiąże się z uciążliwym prowadzeniem księgowości statystycznej oraz wieloma innymi przeszkodami technicznymi. Obecność przetwarzania języka naturalnego można zauważyć praktycznie wszędzie. Najprostszym i najbardziej znanym przykładem może być wyszukiwarka internetowa Google, która znajduje o wiele bardziej dopasowane wyniki, jeśli wykorzystuje indeksowanie stron, biorąc pod uwagę znaczenie komunikatów w języku naturalnym. NLP jest używane również w telefonach komórkowych, gdzie podczas pisania wiadomości używana jest funkcja

autouzupełniania. Asystenci dialogowi (chatboty) używają języka naturalnego w celu odpowiadania na komunikaty rozmówcy [1].

2.1.1. Metody klasyczne

Komputery nadal nie są w stanie poradzić sobie z niektórymi zagadnieniami NLP, takimi jak konwersacja czy rozumienie tekstu, jednak istnieją metody (omówione w kolejnych podrozdziałach) na tyle skuteczne, aby sprawić, że to właśnie maszyny będą mogły przewyższyć ludzi w precyzyjnym i błyskawicznie szybkim wykonaniu różnych zadań. Przykładem sytuacji, w której maszyna radzi sobie lepiej od człowieka jest zidentyfikowanie poszczególnych wiadomości z Twittera pod kątem zawartego w nich sarkazmu – jak się okazuje maszyna jest w stanie dokonać tego o wiele skuteczniej. Jednak w przypadku, kiedy brana jest pod uwagę większa ilość wymienianych komunikatów, człowiek zdecydowanie lepiej jest w stanie przetwarzać, kiedy ktoś żartuje, ponieważ posiada umiejętność przechowywania informacji związanych z daną wiadomością. Obecnie maszyny coraz lepiej radzą sobie z utrzymywaniem kontekstu wypowiedzi. Jeśli wydzieli się z języka naturalnego wszelkie dane liczbowe czy wektory, możliwe jest wykorzystanie dowolnych metod matematycznych oraz narzędzi uczenia maszynowego. Stosowane są identyczne techniki algebry liniowej, z jakich korzysta się podczas rzutowania obiektów 3D na komputery z ekranem 2D. Dzięki zastosowaniu takich algorytmów komputery są w stanie rozpoznawać oraz zachowywać zarówno ilość słów, jak i kontekst wiadomości. Dzięki analizie semantycznej można poradzić sobie z problemem związanym z posiadaniem przez słowa wielu różnych znaczeń. W przypadku języków naturalnych rozpoznawanie informacji polega głównie na odnajdywaniu statystycznych związków pomiędzy frazami, a nie jak to jest w przypadku języków programowania, gdzie bazuje się na rozbudowanym systemie logicznych zasad [1].

2.1.1.1. Tokenizacja

W przetwarzaniu języka naturalnego **tokenizacja** to jeden z rodzajów segmentacji dokumentów. Segmentacja polega na rozłożeniu tekstu na mniejsze fragmenty – w przypadku tokenizacji dokument dzielony jest na tak zwane tokeny (zwykle są to słowa). Tokenizacja stanowi podstawowy krok w potoku NLP. Tokenizator jest odpowiedzialny za rozłożenie danych na mniejsze kawałki, a liczba występowania

tokenów w danym tekście może być wykorzystana jako wektor, który będzie go reprezentował. Dzięki temu dokument zostanie przekształcony w numeryczną strukturę, którą komputer będzie w stanie użyć do wykonania odpowiednich działań i dostarczenia odpowiedzi. Tokenizacja pojedynczego zdania może być w łatwy sposób wykonana poprzez użycie białych znaków w tekście jako separatora słów, co przedstawiono na poniższym rysunku (Rysunek 1) [1].

```
sentence = """Thomas Jefferson rozpoczął budowę Monticello w wieku 26 lat"""
sentence.split()

['Thomas',
 'Jefferson',
 'rozpoczął',
 'budowę',
 'Monticello',
 'w',
 'wieku',
 '26',
 'lat']
```

Rysunek 1 Przykład zdania podzielonego na tokeny (źródło: opracowanie własne na podstawie [1], s. 34)

Po wykonaniu tokenizacji, następnym krokiem jest utworzenie odpowiedniej reprezentacji wektorowej dla każdego ze słów. Utworzone w ten sposób wektory (*one-hot-vectors*) pozwalają na pokonanie pierwszego z problemów NLP, jakim jest transformacja słów na liczby (Rysunek 2) [1].

```
import pandas as pd
pd.DataFrame(onehot_vectors, columns=vocab)
```

	26	Jefferson	Monticello	Thomas	budowę	lat	rozpoczął	w	wieku
0	0	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0
3	0	0	0	0	1	0	0	0	0
4	0	0	1	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0	1
7	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	1	0	0	0

Rysunek 2 Reprezentacja wektorowa (one-hot-vectors) dla zdania (źródło: opracowanie własne na podstawie [1], s. 36)

Dzięki transformacji zdania na wektory, komputer może przeprowadzić na nich odpowiednie operacje i wprowadzić je do potoku przetwarzania języka naturalnego. Wykorzystanie kodowania gorąco-jedynkowego (*one-hot vectors*) podczas transformacji tekstu na pojedyncze słowa sprawia, że zachowane zostają wszystkie informacje dotyczące gramatyki oraz kolejności wyrazów w zdaniu. Jednak stosowanie ich w przypadku dużych rozmiarów dokumentów może być bardzo uciążliwe. Dla każdego pliku zawierającego dane do przetworzenia konieczne jest utworzenie nowej tabeli (macierzy). Zazwyczaj podczas pracy z potokiem NLP rozmiar leksykonu tokenów jest ogromny i może osiągać wartości milionów tokenów. W takim przypadku zastosowanie kodowania gorąco-jedynkowego byłoby bardzo niepraktyczne [1].

Zdecydowanie lepszym rozwiązaniem jest zastosowanie wektora worka słów (ang. *bag-of-words vector*). Ideą tego podejścia jest rozdzielenie dokumentów na krótsze fragmenty i założenie, że większość kontekstu danego zdania może zostać wychwycona ze słów samych w sobie. Zebranie wszystkich wyrazów w tak zwaną „torbę”, gdzie nie jest brana pod uwagę ich kolejność ani poprawność gramatyczna, okazuje się być bardzo racjonalną koncepcją. Okazuje się, że nawet w przypadku dużych plików, wektor worków słów jest bardzo pomocny do przekazania istoty dokumentu. Analizując przykład zdania o Jeffersonie (Rysunek 3) widać, że nawet po zastosowaniu leksykalnego sortowania słów, nadal jest się w stanie odgadnąć, czego dotyczyła wypowiedź.

```
sentence_bow = {}
for token in sentence.split():
    sentence_bow[token] = 1
sorted(sentence_bow.items())

[('26', 1),
 ('Jefferson', 1),
 ('Monticello', 1),
 ('Thomas', 1),
 ('budowę', 1),
 ('lat', 1),
 ('rozpoczął', 1),
 ('w', 1),
 ('wieku', 1)]
```

Rysunek 3 Zapisanie zdania w postaci bag-of-word vector (źródło: opracowanie własne na podstawie [1], s. 40)

W taki sam sposób, jak człowiek nie ma problemu z odczytaniem sensu tego zdania, tak samo nie miałyby z tym kłopotu maszyna. Wykorzystanie idei wektora worka słów jest zatem bardzo przydatne do przetworzenia informacji w strukturę danych łatwą do późniejszej pracy [1].

Efektywność potoku przetwarzania języka naturalnego bezpośrednio wiąże się z wielkością słownictwa, nad którym pracujemy. Jedną z technik redukcji słów jest ich normalizacja. Stosuje się ją w celu transformacji wyrazów o zbliżonym znaczeniu do pojedynczej i znormalizowanej postaci. W taki sposób zredukowana zostaje ilość tokenów koniecznych do przechowywania, jak również zwiększa się skuteczność poprawnego powiązania kontekstów między odmiennymi pisowniami dla danego tokena. Jednym ze sposobów na przeprowadzenie normalizacji słów jest technika składania wielkości liter (z ang. *case folding*). Polega ona na łączeniu wielu postaci danego słowa, różniących się jedynie wielkością liter, z których się składa. Składanie wielkości liter polega na normalizacji wszystkich słów, które zostały napisane dużą literą – zarówno tych, które rozpoczynają dane zdanie w tekście, jak i tych składających się w całości z wielkich liter. Przeprowadzenie takiego zabiegu znacznie zmniejsza rozmiar leksykonu i upraszcza potok NLP, jak również ułatwia połączenie słów posiadających to samo znaczenie, przy użyciu tylko jednego tokena. Kolejną bardzo znaną metodą normalizacji słów jest *stemming*. Technika ta ma na celu wyeliminowanie niewielkich różnic w znaczeniu wyrazów, wynikających z występowania liczby mnogiej słowa lub odmiennej formy czasowników. Stemming normalizuje słowa poprzez znalezienie wspólnej części pośród wielu odmian danego wyrazu. Przykładowo, słowa *cooking* i *cooks* mają wspólny rdzeń (*stem*), jakim jest *cook*. Stemming eliminuje wszelkie przyrostki z wyrazów i stara się połączyć ze sobą słowa o zbliżonym znaczeniu pod jednym rdzeniem. Nie jest wymagane, aby wspólny rdzeń był poprawnym gramatycznie wyrazem. Wystarczy, aby był tokenem, który reprezentuje różne odmiany pisowni dla danego wyrażenia. Istnieje jeszcze inna technika normalizacji, nazywana lematyzacją, której działanie polega na szerszym zagłębieniu w dane słowo, aż do jego semantycznego korzenia - tak zwanego lematu. Dzięki zabiegowi lematyzacji możliwe jest połączenie ze sobą słów o odmiennej pisowni, jeśli znane są informacje na temat zależności pomiędzy ich znaczeniami. Technika ta jest uważana za dużo bardziej precyzyjną w porównaniu do stemmingu czy normalizacji wielkości liter, ponieważ uwzględnia znaczenie wyrazu. Poprzez wykorzystanie bazy wiedzy na temat wyrazów bliskoznacznych oraz końcówek słów

można zagwarantować, że tylko wyrażenia o podobnym kontekście zostaną połączone jednym tokenem [1].

2.1.1.2. Wektoryzacja

Do najbardziej skutecznych i najczęściej wykorzystywanych sposobów reprezentowania słów należą worki słów (*bags of words*), worki n-gramów (*bags of n-grams*) oraz wektory TF-IDF (*TF-IDF vectors*). Istnieją również metody oparte na sieciach neuronowych – wektory słów - word2vec. Worki słów to wektory liczby lub częstotliwości wyrazów, charakteryzujące się wysoką wydajnością i prostotą. [4]. Worki n-gramów zliczają liczbę par słów (bigramów), trójek (trigramów) itd. i pomimo faktu, że ich wykorzystanie jest bardzo łatwe, okazują się być znakomite, a często dużo lepsze od innych rozwiązań w niektórych zadaniach NLP [5].

Wektory TF-IDF służą do oceny słów, która lepiej odzwierciedla ich znaczenie. TF-IDF jest skróconą wersją wyrażenia *term frequency times inverse document frequency*. Term frequency (częstotliwość terminów) to zliczenie każdego wyrazu w dokumencie, natomiast *inverse document frequency* (odwrotna częstotliwość dokumentów) dotyczy faktu, że każda z tych liczb słów zostanie podzielona przez ilość dokumentów, w jakich konkretny wyraz się znajduje. Wszystkie z wymienionych metod: worki słów, worki n-gramów oraz wektory TF-IDF, mogą być wykorzystywane zarówno samodzielnie, jak również jako jedna z części potoku przetwarzania języka naturalnego. Stanowią one modele statystyczne ze względu na ich działanie związane z częstotliwością. Liczba wystąpień słowa w danym dokumencie nazywana jest częstotliwością wyrażenia (z ang. *term frequency* - *TF*). Analizując tylko ilość powtórzeń konkretnego wyrazu, można naprawdę wiele się o nim dowiedzieć. Istnieje jednak sposób na bardziej eleganckie zaprezentowanie zliczonych słów w celach matematycznych, jakim jest stworzenie wektora słów. Występuje on powszechnie jako uporządkowana kolekcja lub tablica i pozwala na wykonywanie operacji obliczeniowych bezpośrednio na nim. Dysponowanie tylko jednym wektorem dla pojedynczego dokumentu nie jest zadowalające. Istnieje opcja, aby wykorzystać kilka innych źródeł i utworzyć dla nich wektory, jednak wartości w każdym z nich muszą być zależne i spójne z pozostałymi. Wektory umożliwiają użycie metod algebry liniowej do ich analizy. W programie reprezentowane są zwykle w postaci uporządkowanej listy liczb lub współrzędnych. Przestrzeń wektorowa stanowi zbiór wszystkich wektorów. Zgodnie z tym, wektor

posiadający dwie składowe będzie położony w przestrzeni wektorowej 2D, natomiast wektor z trzema składowymi w przestrzeni 3D, itd. W przypadku przestrzeni wektorowej dla dokumentu języka naturalnego, wymiarowość określa się poprzez zliczenie liczby różnych słów pojawiających się w całym korpusie. Ważną kwestią jest oszacowanie podobieństwa dokumentu w celu odnalezienia wykorzystania jednakowych słów szacunkowo tyle samo razy w zbliżonych proporcjach. Wynik takiej ewaluacji pozwala na stwierdzenie, że źródła, które reprezentują z dużym prawdopodobieństwem dotyczą tych samych rzeczy [1].

Podobieństwo cosinusowe to cosinus kąta między dwoma wektorami (θ). Poniżej przedstawiono wzór na jego wyznaczenie z euklidesowego iloczynu punktowego.

$$A * B = |A||B| * \cos\theta \quad (1.1)$$

Wartości dla podobieństwa cosinusowego dla większości zagadnień uczenia maszynowego znajdują się w zakresie od 0 do 1. Jeśli wartość prawdopodobieństwa jest równa 1, oznacza to, że znormalizowane wektory są identyczne i wskazują w tym samym kierunku. W przypadku wektorów dla dokumentów języka naturalnego, których podobieństwo cosinusowe jest wartością bliską 1, wiadomo, że źródła wykorzystują pokrewne wyrazy z zachowaniem podobnych proporcji. Innymi słowy, dokumenty, których wektory znajdują się w niedalekiej odległości od siebie, z dużym prawdopodobieństwem dotyczą tej samej rzeczy. Wartość podobieństwa cosinusowego wynosząca 0 wskazuje na wektory nie mające ze sobą żadnych wspólnych składowych. W przypadku wektorów TF NLP może się to zdarzyć jedynie w wypadku, kiedy dwa źródła nie mają żadnych wspólnych wyrazów, a co za tym idzie dotyczą zupełnie innych kwestii. Warto zauważyć, że taka sytuacja nie oznacza, że dokumenty posiadają odmienne znaczenia, ale wskazuje, że wykorzystują całkiem inne słowa [1].

2.1.1.3. Analiza semantyczna

Analiza semantyczna (z ang. *latent semantic analysis*) to algorytm pozwalający na odkrycie semantyki/znaczenia słów oraz wyznaczenie wektorów, które będą je reprezentować. Wyróżnia się dwa rodzaje takich wektorów: semantyczne oraz tematyczne (*semantic and topic vectors*). Wektory semantyczne mają zastosowanie podczas znajdowania wyrazów, które są najbardziej związane z treścią danej wypowiedzi

czy dokumentu. Dzięki nim, możliwe jest wskazanie słów o największym znaczeniu dla danego źródła. Wektory tematyczne są bardzo pomocne w przypadku odnajdywania dokumentów w oparciu o ich znaczenie, co nazywane jest wyszukiwaniem semantycznym. Wyniki takiego poszukiwania są w znacznej większości przypadków bardzo precyzyjne i często dostarczają użytkownikowi dokładnie tego, czego poszukiwał, nawet jeżeli nie potrafił on wybrać pasujących słów w zapytaniu. Wektory tematów LSA mogą składać się z tylko jednego wymiaru, ale mogą też być dużo bardziej rozbudowane i wielowymiarowe. Oszacowanie odległości pomiędzy takimi wektorami może być bardzo użyteczne do grupowania dokumentów względem ich znaczenia, a następnie wyszukiwania odpowiednich źródeł bazując na ich semantyce. Po wykonaniu takiego zabiegu otrzymuje się pojedynczy wektor dokument-temat (*document-topic*) dla każdego dokumentu w korpusie [1].

Opracowanie reprezentacji semantyki słów w postaci numerycznej jest ciężkim zadaniem. Problem pojawia się na przykład dla języka angielskiego, który posiada dużo różnych dialektów oraz znaczeń dla jednakowych wyrazów. **Polisemia**, czyli takie występowanie słów, które posiadają więcej niż jedno znaczenie, może mieć duży wpływ na semantykę wyrazów. Występuje ona pod różnymi postaciami, a do najbardziej znanych należą:

- **homonimy**: słowa o jednakowej pisowni, ale posiadające różne znaczenia
 - **zeugma**: wykorzystanie dwóch kontekstów wyrazu równoległe w jednym zdaniu
 - **homografy**: wyrazy o takiej samej pisowni, ale odmiennej wymowie i znaczeniu
 - **homofony**: wyrazy o takiej samej wymowie, ale odmiennej pisowni i znaczeniu
- [1].

Pomimo wielu komplikacji związanych z polisemią, analiza semantyczna jest w stanie bez problemu sobie z nimi poradzić. LSA to ogólna nazwa na algorytmy analizy semantycznej. [1].

Jednym z przykładów LSA jest metoda PCA, czyli analiza głównych składowych. Jest ona stosowana do pracy na wielowymiarowych danych w uczeniu maszynowym. Okazuje się, że opiera się na dokładnie takiej samej matematyce, co LSA, a różnicą jest jedynie to, że PCA stosowane jest do zmniejszania wymiarowości obrazów, podczas gdy analiza semantyczna redukuje wektory worków słów lub TF-IDF. Analiza semantyczna

bazuje na jednej z najstarszych i najbardziej popularnych metod redukcji wymiarów, jaką jest dekompozycja wartości pojedynczych (SVD). Technika ta polega na rozłożeniu macierzy na trzy kwadratowe macierze, spośród których jedna jest diagonalna. Jest metodą matematyczną polegającą na poszukiwaniu najlepszego rozwiązania dla liniowego przekształcenia wektorów, takich jak bag-of-words czy wektory TF-IDF. W wielu zastosowaniach optymalnym sposobem jest ułożenie wymiarów w nowo utworzonych wektorach z największą możliwą rozpiętością lub wariancją w częstotliwościach słów. Poprzez wybranie takiej metody, w nowej przestrzeni wektorowej nie będą się już znajdowały wymiary, które nie mają dużego wpływu na wariancję wektorów z dokumentu na dokument [1].

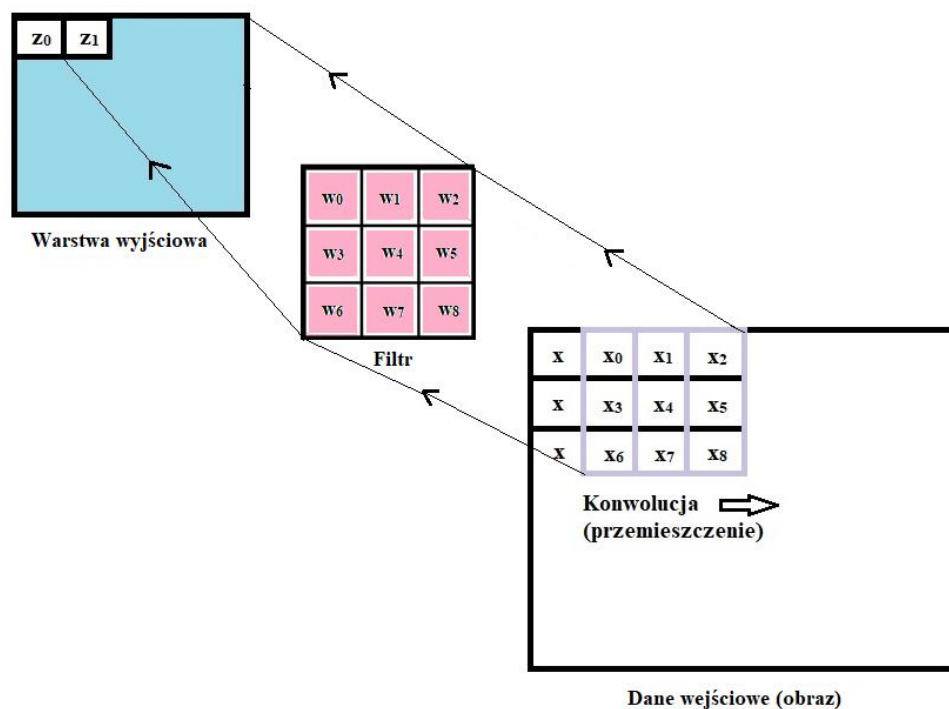
2.1.2. Metody uczenia głębokiego

Wymienione w poprzednim podrozdziale klasyczne metody przetwarzania języka naturalnego badały jedynie liniowe relacje pomiędzy wyrazami. Bardzo często wymagały one ludzkiej interwencji i oceny w celu stworzenia odpowiedniego ekstraktora cech oraz dopasowania parametrów dla modelu. Rozważane w tym podrozdziale metody uczenia głębokiego, jakimi są sieci neuronowe, znacznie ułatwiają pracę i są w stanie w większości samodzielnie przeprowadzić ekstrakcję cech. Są przy tym o wiele bardziej precyzyjne i skuteczne od ręcznie przygotowywanych ekstraktorów, tworzonych przy zastosowaniu metod klasycznych. Zastosowanie sieci neuronowych w uczeniu maszynowym określa się jako „*deep learning*”, czyli głębokie uczenie, ponieważ nowoczesne sieci są często bardzo głębokie, jeśli chodzi o dużą liczbę posiadanych przez nich warstw [6]. Deep learning pozwala na modelowanie tekstu przy zastosowaniu rozległego i nieliniowego podejścia. Na skutek wszechstronnie dostępnych narzędzi open source, sieci neuronowe stały się niezwykle potężne w odnajdywaniu wzorców pośród dużych zestawów danych i dokonały ogromnej transformacji w dziedzinie NLP. W dalszej części podrozdziału przedstawione zostaną coraz bardziej efektywne i prędko rozwijające się warianty sieci neuronowych: konwolucyjne oraz rekurencyjne [1].

2.1.2.1. Konwolucyjne sieci neuronowe

Konwolucyjne sieci neuronowe (*convolutional neural nets* – CNN) są bardzo efektywne w rozwiązywaniu problemów dotyczących rozpoznawania obrazów [8]. Zawdzięczają swoją nazwę idei przemieszczania się małego okna na zestawie danych,

gdzie piksele w tym oknie są łączone (konwolucja) w jeden piksel. Jednym z początkowych zastosowań CNN było przetwarzanie oraz identyfikacja obrazów. Konwolucyjne sieci neuronowe potrafią wychwycić, jakie obiekty znajdują się na danym rysunku, dzięki ich zdolności do znajdowania przestrzennych relacji między punktami danych we wszystkich próbkach. Działanie CNN jest całkiem inne niż w przypadku standardowej sieci zasilającej, ponieważ w miejsce określania wagi dla elementów obrazu, opracowuje się odpowiedni zestaw filtrów realizowanych przez macierze (tak zwanych jąder konwolucyjnych), które przemieszczają się po obrazie. Wszystkie migawki dla danej próbki danych mogą być wyliczane rozdzielnie, bez względu na pozostałe. Określenie rozmiaru okna filtra należy do osoby konstruującej model i ściśle zależy od struktury danych. W niektórych jednak przypadkach, na przykład dla zbiorów danych bazujących na obrazach, występuje uniwersalny punkt wyjścia i rozmiar okna ustala się jako 3x3 piksele. Filtry złożone są z dwóch części, takich jak zestaw wag oraz funkcja aktywacji. Działanie filtra polega na przemieszczaniu się po obrazie i zatrzymywaniu krok po kroku w celu wykonania zdjęć dla pikseli, które w danym momencie obejmuje. Wszystkie zebrane wartości dla pikseli są kolejno przemnażane przez wagi odpowiadające konkretnemu położeniu w filtrze, a zsumowane iloczyny są ostatecznie dostarczane do funkcji aktywacji.



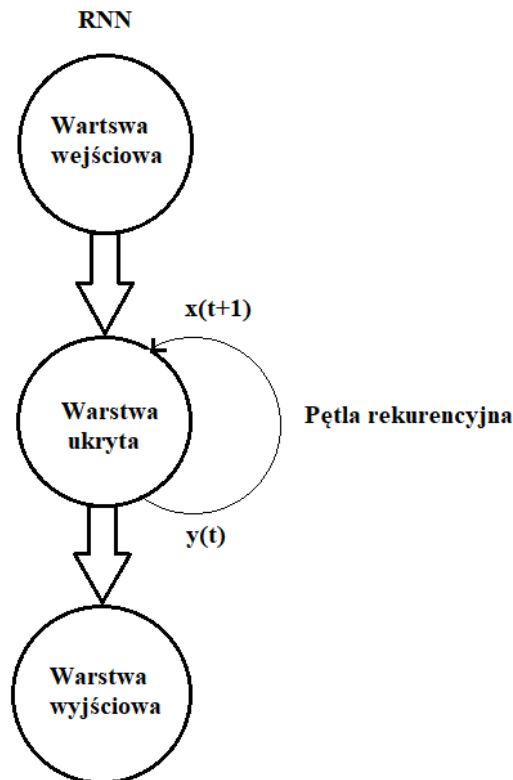
Rysunek 4 Etap konwolucyjnej sieci neuronowej
(źródło: opracowanie własne na podstawie [1], s. 226)

Na powyższym rysunku (Rysunek 4) x_i oznacza wartość piksela w pozycji i , a z_0 jest wyjściem funkcji aktywacji, które jest odnotowane jako wartość pozycyjna w obrazie wyjściowym.

Do przetwarzania języka naturalnego konwolucyjne sieci neuronowe mogą wykorzystywać jako dane wejściowe wektory słów. Wszystkie ważne informacje można zdobyć bazując na horyzontalnych pozycjach pomiędzy wyrazami. Relacje pionowe między słowami byłyby uzależnione od szerokości strony, zatem nie dałoby się wydobyć żadnych istotnych treści ze wzorców, które mogłyby w nich występować [1].

2.1.2.2. Rekurencyjne sieci neuronowe

Rekurencyjne sieci neuronowe (RNN) są bardzo potężnym narzędziem, ponieważ łączą w sobie dwie właściwości: rozproszony stan ukryty, który pozwala w efektywny sposób na przechowywanie dużej ilości informacji o przeszłości oraz nieliniową dynamikę, która umożliwia im aktualizację ich ukrytego stanu na różne sposoby [8]. RNN zapewniają sieciom neuronowym możliwość zachowania w pamięci wszystkich poprzednio występujących wyrazów w danym zdaniu. Na poniższym rysunku przedstawiono sposób ich działania (Rysunek 5) [1].



Rysunek 5 Rekurencyjna sieć neuronowa (źródło: opracowanie własne na podstawie [1]) s. 250

Neuron rekurencyjny znajdujący się w warstwie ukrytej (*Hidden layer*) dodaje rekurencyjną pętlę, aby umożliwić ponowne wprowadzenie do obiegu wyjścia z warstwy ukrytej w określonym czasie t . Wyjście w czasie t jest przyłączane do kolejnego wejścia w czasie $t+1$. Sieć analizuje wynik wyjściowy w kroku t oraz nowy token w kroku $t+1$ celem obliczenia wyniku dla warstwy ukrytej w czasie $t+1$, które następnie zostaje jeszcze raz przekształcane na wejście w kroku czasowym $t+2$ i tak dalej. Po podzieleniu wszystkich próbek danych na tokeny, następuje umieszczenie każdego z tokenów w sieci typu *feedforward*, czyli takiej, gdzie informacja porusza się tylko w jednym kierunku - do przodu - od węzłów wejściowych, przez węzły ukryte i do węzłów wyjściowych. Następnie wyjście z każdego kroku czasowego jest dostarczane do wejścia identycznej warstwy sąsiadującej z wejściową, z kolejnego kroku czasowego. Wyjście pochodzące z ostatniego kroku czasowego jest pobierane i oceniane względem zgodności z etykietą. Następuje propagacja wsteczna błędu poprzez całość grafu, tak, aby dotrzeć ostatecznie do początkowego wejścia z krokiem czasowym 0. Wykorzystanie sieci rekurencyjnej może okazać się bardzo kosztowne. Jest to widoczne szczególnie w przypadku, kiedy

wystąpi potrzeba wyuczenia długiej sekwencji, na przykład takiej posiadającej 10 tokenów. Większa ilość tokenów wiąże się z tym, że algorytm propagacji wstecznej będzie wykonywany przez większą liczbę kroków. Wiąże się to z koniecznością wyznaczenia większej ilości pochodnych. Rekurencyjne sieci neuronowe mogą więc być wymagające obliczeniowo, ale za to nadal bardzo efektywne [1].

Warto też wspomnieć o istnieniu dwukierunkowych rekurencyjnych sieci neuronowych. Są one bardzo przydatne, szczególnie w przypadkach, kiedy analizując daną informację istnieje potrzeba powrócenia do wcześniejszych części tekstu. Można to wytłumaczyć na podstawie prostego zdania:

„Dzieci chciały pogłaskać psa, którego futro było brązowe”.

Po przeczytaniu fragmentu tej wypowiedzi i zatrzymaniu się przy słowie „futro”, wiadomo już coś na temat psa. Jednak czytając dalej okazuje się, że oprócz tego pies ten posiada futro, które dodatkowo jest brązowe. Te nowe informacje są istotne dla wcześniejszej czynności pogłaskania i faktu, że to „dzieci” chciały tego dokonać. Ludzie są w stanie podczas czytania poradzić sobie z komentarzami, które mogą być ustawione w dowolnej kolejności i je ze sobą połączyć. Dwukierunkowa sieć rekurencyjna została stworzona właśnie w celu umożliwienia modelowi cofania się poprzez dane wejściowe i dzięki temu lepszemu zanalizowaniu dokumentów. [1]

2.2. Chatboty

2.2.1. Definicja chatbota

Chatbot [9] to interaktywny system umożliwiający użytkownikom komunikację z nim za pomocą języka naturalnego. Wchodzenie w interakcję z chatbotem obejmuje zarówno przeprowadzanie rozmów tekstowych, jak również głosowych. Pierwsze chatboty pojawiły się na początku 2016 roku i zrewolucjonizowały świat, wprowadzając do społeczeństwa nową erę technologii: erę interfejsu konwersacyjnego. W niedługim czasie rozmowy przeprowadzane z chatbotami będą praktycznie nierozróżnialne z tymi prowadzonymi z rodziną czy przyjaciółmi. Do najbardziej znanych platform chatbotowych należą: Dialogflow [11], IBM Watson Assistant [17], Microsoft Bot Framework [23], Google Assistant [24] oraz Amazon Alexa [25]. Można spotkać ich jednak znacznie więcej, szczególnie, że występuje obecnie wiele dostępnych narzędzi do

samodzielnego tworzenia chatbotów, dopasowanych do konkretnych potrzeb. Przetwarzanie języka naturalnego ma ogromny wpływ na to, jak wirtualni asystenci są w stanie zrozumieć wypowiedzi użytkowników oraz odpowiadać w racjonalny i inteligentny sposób. Do utrzymywania w pamięci wszystkich wypowiedzianych zdań i odpowiedzi, chatbot potrzebuje czterech typów przetwarzania oraz bazy danych. Etapy przetwarzania są następujące:

- Parsowanie – wydzielenie cech i uporządkowanych danych liczbowych z dokumentu napisanego w języku naturalnym.
- Analizowanie – tworzenie oraz zestawianie ze sobą cech, bazując na ocenie wypowiedzi pod względem znaczenia, semantyki i poprawności gramatycznej.
- Generowanie – konstruowanie potencjalnych odpowiedzi poprzez wykorzystywanie szablonów, przeszukiwanie lub używanie modeli językowych.
- Wykonywanie - projektowanie wypowiedzi na podstawie historii rozmowy i dopasowanie kolejnej odpowiedzi [9].

Każdy z wymienionych etapów może wykorzystywać do przetwarzania jeden lub więcej algorytmów pracujących jednocześnie albo szeregowo. Zdecydowana większość chatbotów posiada wszystkie z podanych czterech etapów przetwarzania, razem z bazą danych. Istnieją jednak aplikacje, w których wystarczy zastosować proste algorytmy dla wielu spośród tych elementów. Nie wszystkie chatboty będą tak samo dobre w identycznych przypadkach – jedne lepiej będą radzić sobie odpowiadając na przemyślane pytania, podczas gdy pozostałe świetnie sprawdzą się w tworzeniu długich i skomplikowanych, oraz praktycznie nierozróżnialnych od ludzkich, wypowiedzi. Dla każdej możliwości potrzebne jest zastosowanie odmiennego podejścia [9].

2.2.2. Ewolucja chatbotów na przestrzeni lat

Koncepcja chatbotów została podjęta w latach 50. i 60. XX wieku przez informatyków: Alana Turinga oraz Josepha Weizenbauma, którzy zastanawiali się nad stworzeniem maszyn komunikujących się w taki sam sposób, jak ludzie. W 1966 r. Weizenbaum za pomocą zaledwie 200 linijek kodu stworzył pierwszego w historii chatterbota o imieniu Eliza [9]. Bot wzorował się na języku psychoterapeuty, wykorzystując wyrażenia regularne oraz reguły. Użytkownicy mieli świadomość, że wchodząc w interakcję z Elizą, komunikują się z maszyną, ale pomimo tego, jej

realistycznie emocjonalne reakcje sprawiały, że bardzo łatwo nabierali do niej sympatii. W późniejszych latach, kiedy zaczęły powstawać narzędzia służące do przetwarzania sygnałów, naukowcy postanowili skoncentrować się na tworzeniu narzędzi dialogu mówionego, mając na celu ulepszenie doświadczeń użytkowników. W latach 1980-2000 powstało bardzo dużo tego typu systemów i służyły one początkowo do celów wojskowych, aby udoskonalić zautomatyzowaną komunikację z żołnierzami. Systemy dialogu mówionego były przeznaczone do przekazywania instrukcji, które następnie były tłumaczone chatbotom, aby pomagać użytkownikom w otrzymaniu odpowiedzi na różne pytania dotyczące konkretnych usług. Chatboty były cały czas konstruowane w sposób ręczny, a tworzone przez nie odpowiedzi były takie same i boty nie były w stanie poradzić sobie z wychwyceniem kontekstu wypowiedzi [9].

W ciągu ostatnich lat boty bardzo ewoluowały i stały się bardzo przydatne z powodu rozpowszechnienia smartfonów oraz rozwoju w dziedzinie uczenia maszynowego. Poza interfejsami API do konstruowania chatbotów na znanych platformach do komunikacji, takich jak Facebook Messenger, istnieje dużo więcej różnych narzędzi do tworzenia sztucznej inteligencji, które są rozwijane każdego dnia. Dzięki temu można w łatwy sposób budować własne chatboty, bez potrzeby posiadania bogatego zaplecza AI [9].

2.2.3. Zastosowanie chatbotów

Chatboty mają zastosowanie w bardzo wielu różnych dziedzinach, takich jak na przykład: zakupy i handel elektroniczny, wiadomości, obsługa klienta, medycyna, prawo. W przypadku branży e-commerce chatboty służą pomocą przede wszystkim podczas składania zamówień oraz płatności. Istnieje też wiele botów polecających różne towary i coraz częściej skupia się uwagę na tworzeniu konwersacyjnych systemów do rekomendowania, w celu ulepszenia doświadczeń użytkowników. Chatboty są bardzo często używane do wyszukiwania informacji. Użytkownicy podają kryteria oraz cele poszukiwań za pomocą konwersacji, a zadaniem maszyn jest przekazanie odpowiednich wyników, zgodnych z podanymi wymaganiami. Inną dziedziną, w której chatboty są intensywnie używane jest obsługa klienta. Boty pomagają w analizowaniu skarg oraz dostarczają odpowiedzi na wszelkie pytania FAQ. W branży medycznej chatboty są niezwykle przydatne, jeśli chodzi o wsparcie pacjentów w jak najszybszym odnalezieniu potrzebnych informacji o ich sprawności fizycznej, bazując na towarzyszącym im

objawom. Ostatnimi czasy, bardzo popularne stały się również, szczególnie wśród starszych osób, chatboty, które starają się dowiedzieć o ich stanie zdrowia poprzez zadawanie im precyzyjnych i odpowiednio skonstruowanych pytań. Boty mogą być też bardzo pomocne w dziedzinie prawnej przy obsłudze najczęściej zadawanych pytań przez użytkowników. Mogą też świetnie sprawdzić się w sytuacjach, kiedy dana osoba szuka artykułu prawnego dotyczącego konkretnej sprawy, a bot zadaje wtedy odpowiednie pytania uzupełniające, aby być w stanie odszukać najbardziej pasującą odpowiedź [9].

2.3. Klasyfikacja systemów dialogowych

2.3.1. Chatboty oparte o reguły

Chatboty oparte o reguły (*Rule-based chatbots*) [10], s. 9 - nazywane też często botami decyzyjnymi, posługują się zestawem zdefiniowanych reguł do wytwarzania odpowiedzi. Mogą wykorzystywać zarówno bardzo proste, jak również bardziej skomplikowane reguły. Nie są jednak w stanie dostarczyć odpowiedzi na żadne z pytań, które nie zostały wcześniej odpowiednio zdefiniowane. Chatboty oparte o reguły nie potrafią uczyć się na podstawie interakcji. Posiadają zdolność do wykonywania oraz pracowania tylko z takimi schematami, jakie zostały im przedstawione i które wcześniej opanowały. Stosowanie tego typu chatbotów może być o wiele bardziej czasochłonne od innych ze względu na konieczność tworzenia wielu ręcznie zapisanych reguł. Pomimo tego, są one jednak w stanie objąć dużo bardziej rozległy zakres tematów [10].

2.3.2. Chatboty oparte o korpus

Chatboty oparte o korpus (*Corpus-based chatbots*) [10], s. 9 - to rodzaj botów bazujących na regułach, które są w stanie zrozumieć język naturalny, a dodatkowo działać zgodnie z wyznaczonym schematem tak, aby dostarczyć najbardziej trafną odpowiedź na dany problem. Chatboty oparte o korpus posiadają zdolność do zapamiętywania kontekstu rozmowy oraz koniecznych do spełnienia wymagań. Potrafią z łatwością zmieniać tematy konwersacji w zależności od potrzeb wynikających z obecnej rozmowy i są w stanie odpowiedzieć na każde zapytanie użytkownika w dowolnym momencie. Można podzielić te chatboty na dwie kategorie:

- **Modele oparte na wyszukiwaniu** [10], s. 10 – wymagają posiadania bazy danych, zawierającej dopuszczalne odpowiedzi. Stosując to podejście, wyłaniane są z bazy danych wyniki, które najbardziej pasują do danego zdania, a następnie spośród nich wybierana jest jedna najlepsza odpowiedź do dostarczenia.
- **Modele generatywne** [10], s. 10 – używają metod uczenia maszynowego do tworzenia odpowiedzi w czasie rzeczywistym. Wykorzystywane są do generowania odpowiedzi za pomocą tłumaczenia danych początkowych na odpowiedzi i trenowania na zestawie danych realnych rozmów.

2.3.3. Modele hybrydowe

Modele hybrydowe stanowią połączenie chatbotów opartych na regułach oraz bazujących na głębokim uczeniu. Są one konstruowane przy pomocy zestawu predefiniowanych zasad oraz algorytmów uczenia maszynowego [10], s. 12.

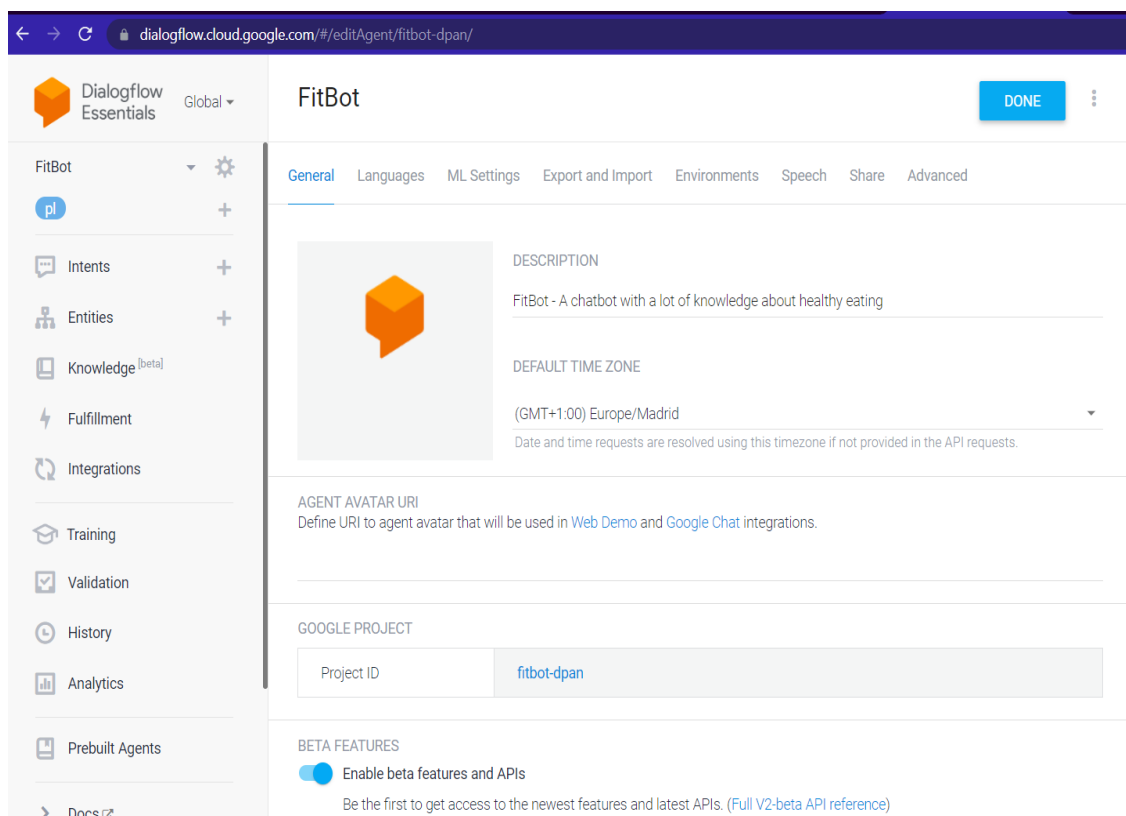
2.4. Przegląd frameworków do tworzenia chatbotów

2.4.1. Dialogflow

Dialogflow [11] to platforma należąca do Google'a, pozwalająca na budowanie aplikacji rozumiejących język naturalny i dająca użytkownikom możliwość prowadzenia konwersacji, zarówno w formie tekstowej, jak i głosowej. Dialogflow posiada dwie możliwe usługi do tworzenia wirtualnych agentów: Dialogflow CX, który najlepiej sprawdzi się do konstruowania dużych i złożonych chatbotów oraz Dialogflow ES, zapewniający standardowy typ agenta, najlepszy dla mniejszych i prostych chatbotów (Rysunek 6). Dialogflow umożliwia obsługę aż 32 języków i dialektów, w tym języka polskiego. Posiada konektory, które pozwalają na integrację tworzonych chatbotów z różnymi popularnymi komunikatorami oraz platformami, takimi jak na przykład: Asystent Google, Facebook Messenger, Slack, Skype, Amazon Alexa, Viber, Twilio, Telegram [12]. Koncepcje, które są wykorzystywane w zrozumieniu języka naturalnego przez Dialogflow to przede wszystkim:

- Odgadywanie w wypowiedziach użytkownika jego intencji, czyli w zależności od sytuacji konkretnej potrzeby, prośby, pytania (z ang. *intent matching*).
- Rozpoznawanie encji oraz parametrów, które są konieczne do prawidłowego odczytania wypowiedzi użytkownika (z ang. *entity extraction*).

- Monitorowanie kontekstu rozmowy w trakcie następujących po sobie wiadomości (z ang. *context persistence*).
- Powiązanie interfejsu konwersacyjnego z odpowiednimi systemami informatycznymi oraz logiką biznesową (z ang. *fulfillment*).
- Ocena sentymentu, czyli analiza wiadomości użytkownika pod kątem emocjonalnym (z ang. *sentyment analysis*).
- Analiza mowy, czyli transformacja mowy na tekst lub tekstu na mowę (z ang. *speech to text* – STT lub *text to speech* – TTS) [12].



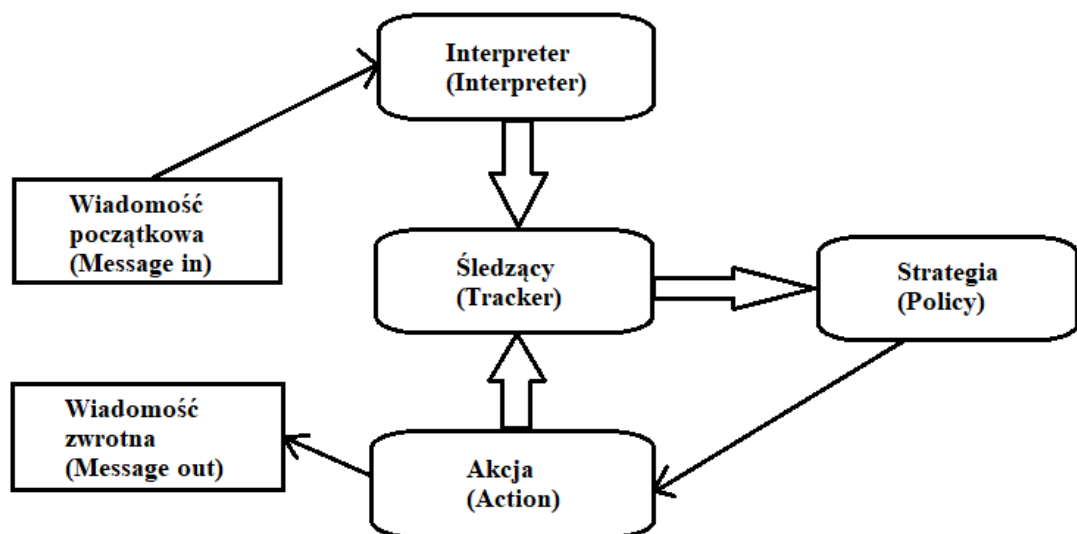
Rysunek 6 Panel do zarządzania chatbotem za pośrednictwem narzędzia Dialogflow (źródło: opracowanie własne)

2.4.2. Rasa

Rasa Open Source [13] to najpopularniejszy framework do tworzenia czatów oraz głosowych asystentów AI. Umożliwia komunikację z wieloma platformami komunikacyjnymi, takimi jak na przykład: Facebook, WhatsApp czy Telegram oraz zapewnia bloki konstrukcyjne do budowania wirtualnych chatbotów [13]. Rasa składa się z dwóch podstawowych modułów:

- Rasa NLU – w tym miejscu odbywa się próba zrozumienia wypowiedzi użytkowników, odczytanie intencji oraz wydobycie encji zawartych w wiadomości.
- Rasa Core – po otrzymaniu danych wyjściowych z Rasa NLU, moduł odpowiedzialny jest za poprawne odczytanie kontekstów, wynikających z przepływu wiadomości oraz wybranie najlepszej odpowiedzi i przesłanie użytkownikowi [14].

Na poniższym rysunku (Rysunek 7) przedstawiono schemat architektury Rasy. W momencie, kiedy użytkownik wysyła wiadomość chatbotowi (*Message in*), jest ona odbierana oraz przekazywana do interpretera (*Interpreter*). Jego zadaniem jest rozpoznanie intencji tekstu oraz wydobycie z niego potrzebnych informacji. Kolejnym elementem architektury Rasa jest tzw. *Tracker*, którego obowiązkiem jest kontrolowanie przepływu wiadomości pomiędzy użytkownikiem a chatbotem. *Policy* odpowiada za podejmowanie decyzji, jaką akcję powinien wykonać bot w odpowiedzi na wiadomość użytkownika. Podjęta reakcja jest nadal obserwowana przez trackera i ostatecznie przekazywany jest użytkownikowi komunikat zwrotny [15].



Rysunek 7 Schemat architektury Rasy (źródło: opracowanie własne na podstawie [15])

2.4.3. ChatterBot

ChatterBot [16] jest biblioteką języka Python, która wykorzystując algorytmy uczenia maszynowego umożliwia tworzenie automatycznych odpowiedzi i konstruowanie chatbotów. ChatterBot pozwala na stworzenie wirtualnych agentów w dowolnym języku. Dzięki zastosowaniu uczenia maszynowego, możliwe jest udoskonalanie chatbota w oparciu o zdobytą dotychczas wiedzę o dopuszczalnych odpowiedziach, nauczonych podczas interakcji z użytkownikami oraz bazując na innych źródłach informacji. Niewytrenowany ChatterBot po każdej przeprowadzonej z użytkownikiem konwersacji, zapamiętuje, co było początkową wypowiedzią i jaki był jego komunikat zwrotny. Im więcej rozmów zostanie podjętych przez bota, tym bardziej zwiększy się jego wiedza na temat dopuszczalnych wiadomości zwrotnych, dzięki czemu po pewnym czasie będzie w stanie dawać odpowiedzi najbardziej pasujące do zadanego pytania. ChatterBot posiada moduły adapterów, które służą do kontrolowania zachowania dla konkretnych rodzajów zadań. Adaptery pamięci masowej (*storage adapters*) dostarczają interfejs do łączenia się z systemami zarządzania bazą danych, takimi jak np. MongoDB, natomiast adaptery logiczne (*logic adapters*) odpowiadają za logikę, którą ChatterBot wykorzystuje do odpowiadania na otrzymane dane wejściowe. Duża ilość danych treningowych jest kluczem do tego, aby chatbot był bardziej efektywny i dokładny. ChatterBot posiada zestaw narzędzi ułatwiających trenowanie instancji bota. Szkolenie chatbota polega na dostarczaniu dopuszczalnych dialogów do jego bazy danych. Dzięki temu zestawy znanych wypowiedzi oraz odpowiedzi są stale rozbudowywane. Wirtualny asystent posiada kilka wbudowanych klas służących do trenowania. Jedną z nich umożliwia aktualizowanie wiedzy bazy danych w oparciu o listę wypowiedzi, które reprezentują dialogi, co pokazano na poniższych rysunkach.

```
trainer = ListTrainer(chatbot)

trainer.train([
    "Hej!",
    "Cześć",
])

trainer.train([
    "Witaj!",
    "Cześć",
])
```

Rysunek 8 Sposób przekazania osobnych, krótkich dialogów do trenowania chatbota (źródło: opracowanie własne na podstawie [16])

```
trainer.train([
    "Jak się czujesz? Czy wszystko w porządku?",
    "Tak, jest w porządku.",
    "To super, cieszę się.",
    "Dziękuję za troskę.",
    "Nie ma za co.",
])
```

Rysunek 9 Sposób przekazania rozbudowanego dialogu do uczenia chatbota (źródło: opracowanie własne na podstawie [16])

Należy przekazać listę zdań, których kolejność jest zależna od ich umiejscowienia w danej rozmowie. Można podawać wiele krótkich dialogów (Rysunek 8) lub dostarczyć dłuższą listę, w której każde kolejne zdanie będzie możliwą odpowiedzią na poprzednie (Rysunek 9).

Inną opcją może być narzędzie, które pozwala na szkolenie chatbota poprzez dostarczenie korpusu danych treningowych. ChatterBot dostarcza moduły danych korpusowych i użytkowych, które pomagają w trenowaniu chatbotów w komunikacji z użytkownikiem. Na poniższym rysunku (Rysunek 10) pokazano przykład wybrania modułu danych korpusu dla języka angielskiego.

```
corpus_trainer = ChatterBotCorpusTrainer(chatbot)
corpus_trainer.train("chatterbot.corpus.english")
```

Rysunek 10 Przekazanie chatbotowi korpusu językowego do wytrenowania (źródło: opracowanie własne na podstawie [16])

Możliwe jest też samodzielne zbudowanie własnej klasy szkoleniowej. Jest to szczególnie przydatne w przypadku, kiedy istnieje konieczność trenowania bota przy pomocy danych zapisanych w formatach nieobsługiwanych przez domyślne narzędzia [16].

2.4.4. IBM Watson

IBM Watson [17] to superkomputer oparty na sztucznej inteligencji, posiadający wysokorozwinięte oprogramowanie analityczne, dzięki któremu może być używany jako efektywna maszyna do odpowiadania na pytania. Dzięki wykorzystaniu 90 serwerów oraz magazynu danych z ogromną ilością informacji analizowanych przez blisko sześć milionów reguł logicznych, jest w stanie odpowiadać na pytania lepiej niż człowiek. IBM udostępnia użytkownikom wiele interfejsów API, pozwalających na konstruowanie

aplikacji do przetwarzania języka naturalnego. Jednym z przykładowych narzędzi jest Asystent IBM Watson. Narzędzie to zostało zbudowane na podstawie modeli głębokiego uczenia, uczenia maszynowego oraz NLP. Posiada zdolność do rozpoznawania pytań oraz znajdowania najlepszych odpowiedzi do przekazania użytkownikom. Watson potrafi odczytywać intencje wiadomości oraz rozpoznawać intencje, dzięki czemu jest w stanie zrozumieć kontekst wypowiedzi. Istnieje kilka możliwych kanałów do interakcji z asystentem:

- Czat internetowy - umieszczony na stronie internetowej danej firmy, przekazuje zapytania do pracownika działu obsługi klienta.
- Platformy komunikacyjne – platformy służące do prowadzenia konwersacji w mediach społecznościowych, przykładowo: Slack, Facebook Messenger, WhatsApp.
- Rozmowa przez telefon lub wymiana wiadomości tekstowych.
- Aplikacje mobilne lub roboty posiadające interfejs głosowy.

W przypadku budowania asystentów dla konkretnych firm, Watson jest w stanie połączyć się z platformą danych wszystkich klientów oraz różnymi cyfrowymi narzędziami i innymi dodatkowymi aplikacjami, aby dostarczyć obsługę na jak najwyższym poziomie. Asystent posiada rozbudowaną analitykę, dzięki której może zrozumieć oraz udoskonalić efektywność poprzez integrację z platformami, przechowującymi dane o klientach (np. Segment) oraz przekazywanie swoich uwag innym aplikacjom [17].

2.5. Metryki oceny chatbota

Ocena chatbota jest trudnym zadaniem, ponieważ nie istnieje jeden ogólny schemat określający dokładnie, jakie cechy oraz wymagania powinien on spełniać. Ciężko jest zdecydować, jak efektywny jest dany system oraz porównać go z innymi dostępnymi, ponieważ większość z istniejących analiz dotyczących działania chatbotów w perspektywie technologicznej nie bierze pod uwagę punktu widzenia biznesu oraz użytkownika. Analizując wiele artykułów oraz prac badawczych dotyczących metryk oceny chatbota wybrano kilka, które wydają się być najbardziej trafne. PARAdigm for Dialogue System Evaluation (PARADISE) [18] jest jednym z najstarszych frameworków przeznaczonych do analizy oraz oceny chatbotów. Rozdziela on wymagania związane z zadaniami od zachowań dialogowych bota, pozwala

na porównywanie strategii dialogowych, kalkuluje efektywność w dialogach, ocenia, jaki wpływ na wydajność mają konkretne czynniki oraz umożliwia zestawienie ze sobą różnych chatbotów, realizujących odmienne zadania poprzez zastosowanie normalizacji dla złożoności zadania. Do wyliczenia wpływu poszczególnych parametrów na efektywność chatbota, PARADISE używa do tego teorii decyzji. Wydajność jest wyznaczana jako ważona funkcja bazującej na danym zadaniu miary sukcesu oraz miary kosztów związanych z danym dialogiem, z wagami oszacowanymi na podstawie korelacji zadowolenia użytkownika z efektywnością. Do innych rozwiązań stosowanych przez framework należy wykorzystanie współczynnika Kappa [19] do operacjonalizacji powodzenia wykonania zadania oraz zastosowanie regresji liniowej w celu oceny ilościowej wpływu poszczególnych czynników sukcesu na satysfakcję użytkowników. W pracy badawczej pod tytułem: „Commercial Chatbot: Performance Evaluation, Usability Metrics and Quality Standards of Embodied Conversational Agents” autorstwa Karoliny Kuligowskiej [20] przedstawionych zostało jedenaście parametrów, będących podstawą do oceny wydajności komercyjnych chatbotów. Wykorzystano tam standardowe narzędzie pomiarowe ze skalą ocen od 1 do 5, gdzie: 1 oznacza bardzo słabe, 2 – słabe, 3 – zadowalające, 4 – dobre i 5 – bardzo dobre. Autorka wymieniła następujące parametry oceny chatbotów:

1. **Wygląd wizualny:** to, w jaki sposób chatbot się prezentuje jest istotnym czynnikiem, mającym wpływ na jakość jego wdrożenia.
2. **Forma implementacji na stronie internetowej:** oceniana jest widoczność bota znajdującego się na danej witrynie.
3. **Moduł tekst-na-mowę:** jak twierdzi Kuligowska, aspekt zamiany tekstu pisanego na mowę syntetyczną może pozytywnie wpłynąć i wzmocnić zaufanie użytkowników. Jest to jeden z parametrów jakościowych.
4. **Baza wiedzy (wiedza podstawowa):** ocena wirtualnego asystenta pod względem umiejętności udzielania odpowiedzi na proste pytania, takie jak np. jego nazwa, aktualna data, itp.
5. **Baza wiedzy (wiedza specjalistyczna):** parametr odpowiedzialny za analizę posiadanej przez chatbota specjalistycznej wiedzy, takiej jak oferowane produkty i usługi, dane kontaktowe oraz zaawansowana wiedza o firmie.

6. **Prezentacja dodatkowej wiedzy i funkcjonalności:** ten komponent jakościowy rozpatruje dodatkowe funkcjonalności posiadane przez chatboty, mające na celu wsparcie użytkownika podczas nawigacji na stronie (np. przycisk "Pomoc" lub przewijanie historii czatu).

7. **Umiejętności konwersacyjne i wrażliwość na kontekst:** ten aspekt oblicza umiejętność chatbota do przeprowadzenia spójnej konwersacji, radzenia sobie ze skomplikowanymi danymi, wpisywanymi przez użytkowników oraz zdolność do kontrolowania rozmowy.

8. **Cechy osobowości:** wirtualni asystenci powinni posiadać umiejętność do wyrażania osobowości. Konieczne jest wzbogacenie bazy wiedzy chatbota o warstwę psychologiczną, dodając informacje o cechach osobowości, szczegółach biograficznych i wyrażanych emocjach.

9. **Opcje personalizacji:** mają istotny wpływ na ocenę jakości interakcji z botem przez użytkowników, którzy określają bota jako dużo bardziej godnego zaufania i pożytecznego, jeśli istnieje opcja zmiany jego cech i wyglądu w celu dopasowania ich do potrzeb konsumentów.

10. **Reakcje w nieoczekiwanych sytuacjach:** boty powinny potrafić radzić sobie w nieprzewidywalnych sytuacjach, na przykład, kiedy użytkownik wprowadzi niezrozumiałą wypowiedź lub nie posiada wystarczającej ilości informacji dotyczących danego pytania, aby na nie odpowiedzieć. Wirtualni asystenci powinni posiadać umiejętność rozwiązywania sytuacji, kiedy użytkownicy używają obelg lub brzydkich słów, być w stanie rozpoznawać różne języki oraz wyłapywać literówki, błędy ortograficzne i kolokwializmy zawarte w wypowiedziach.

11. **Możliwość klasyfikowania chatbota i strony internetowej przez użytkownika:** każdy chatbot powinien umożliwiać konsumentom ocenę ich satysfakcji z przeprowadzonej z nim interakcji. Informacja zwrotna na temat oceny danego chatbota jest bardzo istotna dla jego właściciela.

Jadeja i Varia, autorzy pracy naukowej pod tytułem: „Perspectives for Evaluating Conversational AI” [21], skupili się na czterech głównych aspektach, które należy wziąć pod uwagę podczas oceny działania chatbotów. Są to:

- **Perspektywa użytkownika:** ma na celu analizę satysfakcji użytkownika i efektywności. Według autorów, rozpoznanie wymagań konsumentów, zapewnienie bezpieczeństwa, gdy wymagane są poufne dane użytkownika oraz rozpoznanie strategii konsumentów może poprawić te kryteria. Największą wadą tego aspektu jest to, że wymaga bardzo dużo czasu oraz pracy.
- **Perspektywa pozyskiwania informacji:** dotyczy oceny precyzyjności informacji otrzymywanych od chatbota oraz czasu jego reakcji na otrzymane od użytkownika dane wejściowe.
- **Perspektywa lingwistyczna:** obejmuje pomiar czterech cech związanych z językowymi umiejętnościami wirtualnego asystenta, takimi jak:
 - jakość: oceniana na podstawie tego, jak precyzyjne są wypowiedzi bota
 - ilość informacji: wystawia ocenę na podstawie tego, jak dużo informacji chatbot przekazuje
 - analiza odpowiedzi pod kątem ich powiązania z danym tematem
 - ocena rozmowy: analiza, jak bardzo bezpośrednia oraz prosta jest dana rozmowa.
- **Perspektywa AI:** celem jest ocena umiejętności bota do prowadzenia interakcji z ludźmi, która może być zrealizowana przez wykonanie testu Turinga. Aspekt ten ulepsza zdolności radzenia sobie z problemami oraz wywierania wpływu przez wirtualnego asystenta.

Można zauważyć, że z wymienionych powyżej sposobów oceny chatbotów nie da się stworzyć ogólnego schematu, który można by było wykorzystać do wszystkich rodzajów wirtualnych asystentów. Pomimo że w każdej z przedstawionych metryk koncentrowano się na innych podejściach do oceny efektywności botów, to jednak we wszystkich bardzo ważnym aspektem była perspektywa oceny użytkowników. Można by uznać, że jest to najważniejszy parametr oceny, ponieważ głównym celem chatbotów jest wchodzenie w interakcję z ludźmi, dlatego to ich satysfakcja powinna mieć największe znaczenie.

II. Część praktyczna

3. Implementacja

FitBot to wirtualny asystent, który został zaimplementowany przy użyciu trzech różnych frameworków, takich jak: Dialogflow, Rasa oraz ChatterBot. Jest to chatbot przeznaczony do odpowiadania na pytania użytkowników dotyczące diety, prawidłowego odżywiania oraz prowadzenia zdrowego stylu życia. Oferuje pomoc w dopasowaniu odpowiedniej do potrzeb diety oraz stworzeniu zdrowych posiłków. Posiada szeroką wiedzę na temat zasad odżywiania, składu produktów, rodzajów diet czy też błędów żywieniowych.

3.1. Wymagania

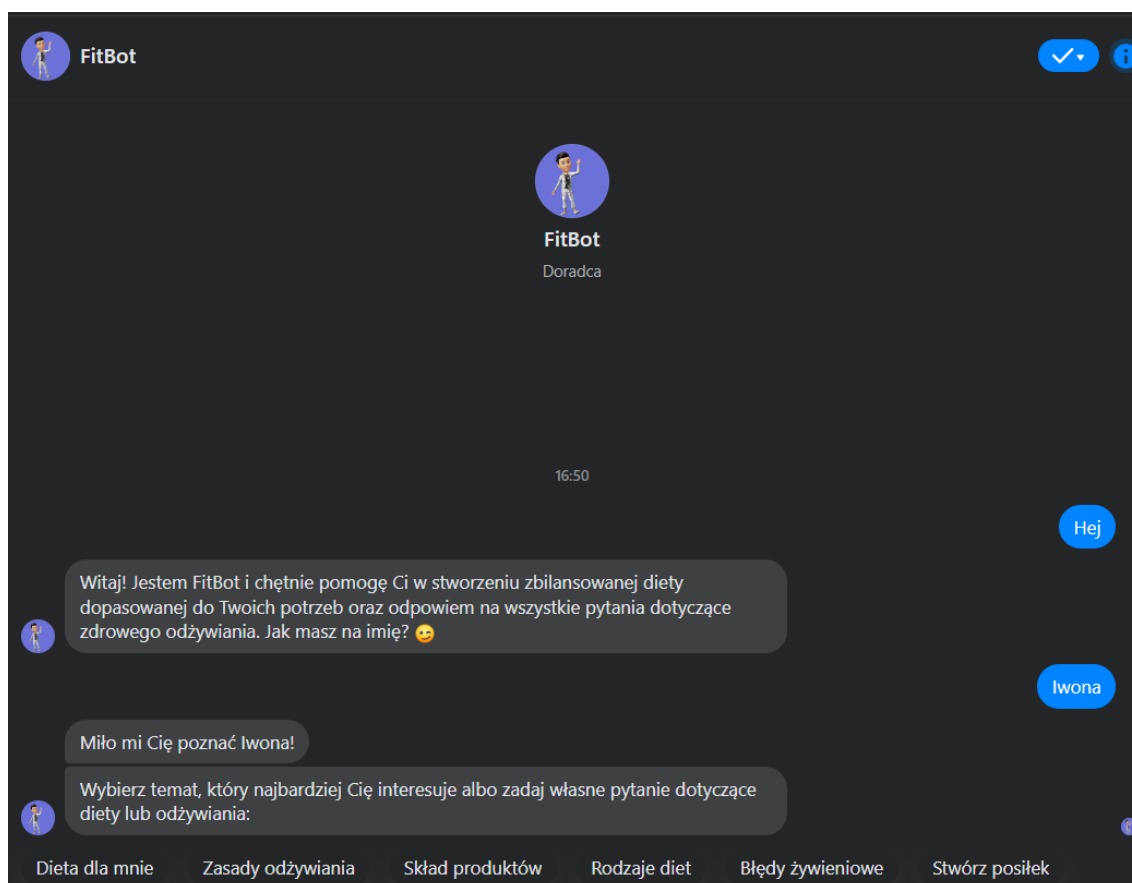
3.1.1. Wymagania funkcjonalne

Wymagania funkcjonalne obejmują wszystkie aspekty mające bezpośredni wpływ na umiejętności chatbota, takie jak rozumienie wypowiedzi użytkowników, tworzenie informacji zwrotnych za pomocą języka naturalnego oraz precyzyjność w zwracanych odpowiedziach.

Do tych wymagań należą:

- **Szybkość:** podstawowym celem chatbota jest błyskawiczne udzielanie odpowiedzi użytkownikom. Powinno zająć mu to znacznie mniej czasu niż człowiekowi. Zamiast przeglądania wielu różnych stron dotyczących zdrowego odżywiania, typów diety czy też samodzielnej próby dopasowywania do siebie jadłospisu, użytkownik powinien dostać szybką odpowiedź na jakiegokolwiek pytanie związane z wymienionymi kwestiami. Dzięki temu znacznie zaoszczędzi swój czas, a dodatkowo zapobiegnie to jego dezorientacji wynikającej z posiadania zbyt wielu informacji pochodzących z różnych źródeł.
- **Zorientowanie na cel:** chatbot został stworzony, aby dostarczyć użytkownikom informacji na temat diety, prawidłowego odżywiania oraz zdrowego stylu życia, dlatego przebieg konwersacji powinien być cały czas zorientowany na ten cel. Jeśli konsument będzie próbował prowadzić rozmowę na całkiem inne tematy, może zakończyć się to rozczarowaniem, ponieważ nie będzie on w stanie zapewnić mu odpowiedzi, jakich oczekuje. Ważną kwestią jest zatem, aby bot jasno określił, jakie jest jego zadanie, aby zapewnić użytkownika na jakie pytania

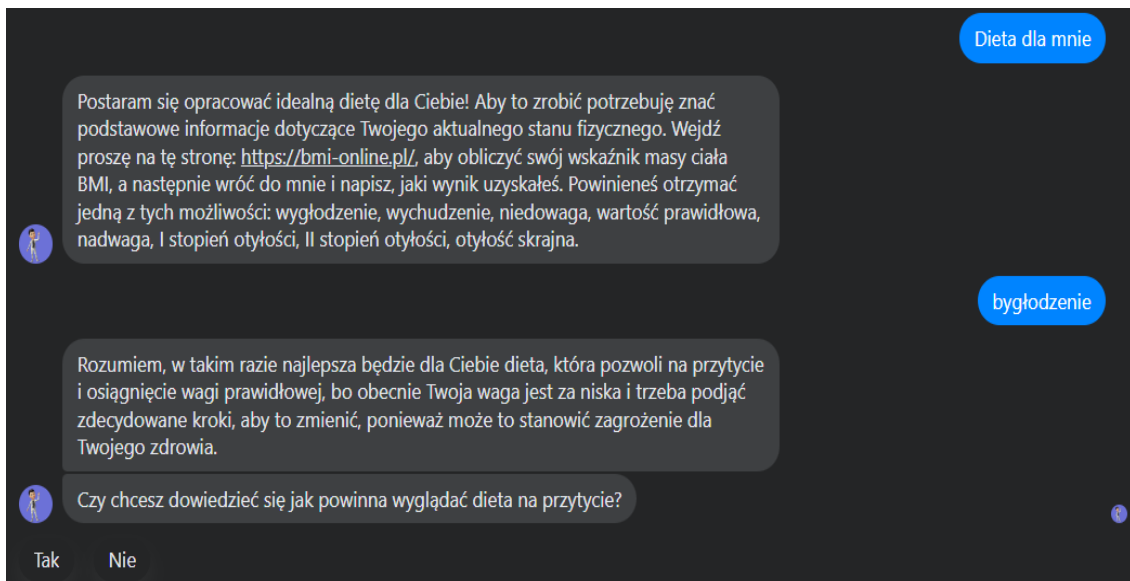
jest w stanie najlepiej mu odpowiedzieć. Na poniższym obrazku (Rysunek 11) pokazano ekran powitalny FitBota, stworzonego w Dialogflow, zintegrowanego z Messengerem. Podczas rozpoczynania dialogu z wirtualnym asystentem od razu zwraca on odpowiednią informację dotyczącą tego, w jakim celu został stworzony, a przyciski sugerują możliwy zakres tematów do wyboru.



Rysunek 11 Ekran powitalny FitBota w narzędziu Dialogflow (źródło: opracowanie własne)

- **Zdolność rozumienia:** chatbot musi posiadać zdolność rozumienia języka naturalnego w obrębie danej dziedziny i umieć określić temat wpisywanych wypowiedzi, niezależnie od tego, w jaki sposób zdanie zostało zbudowane. Dobre umiejętności rozumienia przez chatbota powinny pozwolić użytkownikom na płynną interakcję z nim. Jeśli użytkownik popełni jakiś błąd ortograficzny lub inny podczas pisania, wirtualny asystent powinien uruchomić opcję auto-korekty, aby mimo tego być w stanie zrozumieć jego wypowiedź. Przykładowo, jeśli użytkownik będzie chciał dowiedzieć się czegoś na temat diety dla siebie i zamiast wprowadzenia słowa „wygłodzenie” popełni literówkę i wpisze wyraz:

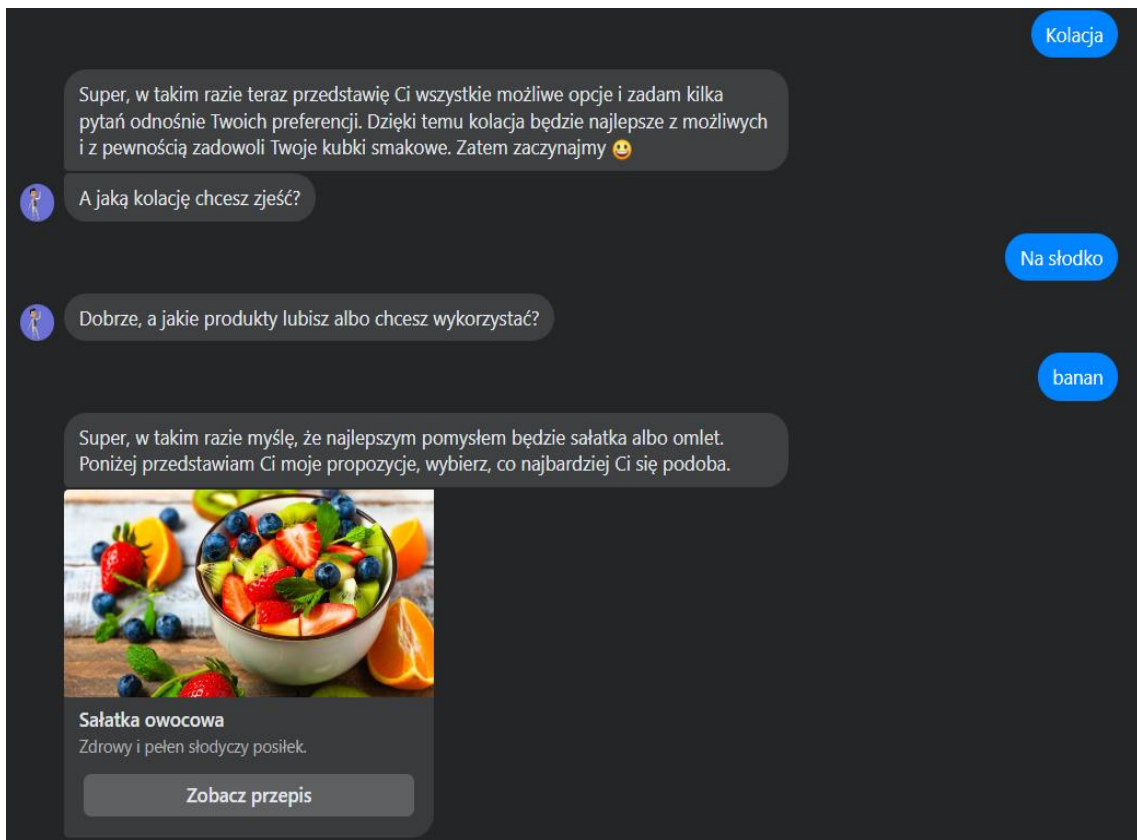
„bygłódzenie”, to FitBot nadal będzie w stanie odpowiednio zareagować i udzielić właściwej odpowiedzi, jak pokazano na poniższym rysunku (Rysunek 12).



Rysunek 12 Przykład radzenia sobie FitBota z rozpoznaniem danych od użytkownika, pomimo błędów występujących w pisowni (źródło: opracowanie własne)

- **Możliwości udzielania odpowiedzi:** po zrozumieniu przez chatbota, czego oczekuje użytkownik, powinien on być w stanie znaleźć najlepszą odpowiedź, bazując na podanych przez niego informacjach oraz kontekście wynikającym z rozmowy. Zwrotne wypowiedzi chatbota powinny być jak najbardziej spójne i zgodne z początkowym kontekstem zdania. Jedynym przypadkiem, kiedy możliwe jest zaakceptowanie braku spójności odpowiedzi z tematem dialogu jest podanie nieznanego botowi wyrażenia. Chatbot musi wtedy rozszerzyć swoją bazę wiedzy o nowe informacje, aby był w stanie prawidłowo je zinterpretować i odpowiedzieć, jeśli taka sytuacja powtórzy się jeszcze w przyszłości.
- **Zaangażowanie użytkownika:** wirtualny asystent powinien posiadać umiejętność podejmowania dialogu z konsumentami i płynnej interakcji. Jego zadaniem jest zdolność do zrównoważonego wykorzystywania tekstu, co oznacza, że powinien stosować kombinację wiadomości, używając zarówno samego tekstu, jak i dodając do tego przykładowo emotikony, tak aby wzbudzić sympatię użytkownika i zachęcić go do rozmowy. FitBot bardzo często używa emotikonów, wzbogaca opisy o zdjęcia oraz proponuje różne opcje do wyboru,

aby zapewnić użytkownikowi łatwiejszą komunikację i zachęcić do zadawania pytań (Rysunek 13).



Rysunek 13 Przykład odpowiedzi chatbota, która wzbogacona jest o emotikony i zdjęcia (źródło: opracowanie własne)

- **Zdolności odwracania odpowiedzi:** w sytuacjach, kiedy bot nie jest w stanie zrozumieć wypowiedzi użytkownika powinien być w stanie odpowiedzieć w sprytny sposób, tak, aby pokazać, że jest inteligentną maszyną i nie ma pytania, z którym by sobie nie poradził.
- **Styl językowy i słownictwo:** chatbot powinien posługiwać się takim samym językiem jak użytkownik. FitBot komunikuje się tylko w języku polskim i powinien odpowiadać, używając odpowiedniej dla tego języka składni, korzystając ze słownictwa związanego z tematyką zbilansowanej diety.
- **Spójność:** podczas interakcji z botem powinien on udzielać odpowiedzi, które są jasne i łatwe do zrozumienia. Nie powinny one zawierać żadnych nieistotnych, zbędnych informacji, które mogłyby utrudnić użytkownikowi zrozumienie wiadomości zwrotnej.

3.1.2. Wymagania niefunkcjonalne

Wymagania niefunkcjonalne obejmują wymagania związane z cechami oraz jakością chatbota, które nie dotyczą jego funkcjonalności lub nie dotyczą bezpośrednio generowania informacji lub interakcji z użytkownikiem. Wymagania niefunkcjonalne, które powinien spełniać FitBot są następujące:

- **Skuteczność:** powinien szybko i płynnie odpowiadać na wszystkie pytania zadawane przez użytkowników.
- **Dostępność:** powinien być dostępny przez całą dobę, 7 dni w tygodniu, aby umożliwić użytkownikom uzyskanie pomocy w dowolnym momencie.
- **Personalizacja:** powinien być w stanie dopasowywać swoje odpowiedzi do indywidualnych preferencji i wymagań konsumentów.
- **Integracja:** powinien być łatwy do zintegrowania z innymi systemami i narzędziami.
- **Użyteczność:** powinien być łatwy w obsłudze, intuicyjny i przyjazny dla użytkowników, aby zapewnić im szybkie i efektywne uzyskanie pomocy.

3.2. Projektowanie chatbotów

We wszystkich z zastosowanych technologii FitBot jest w stanie odpowiadać na pytania dotyczące zbilansowanej diety, w tym szczególnie skupia się na zagadnieniach związanych z tematami, takimi jak:

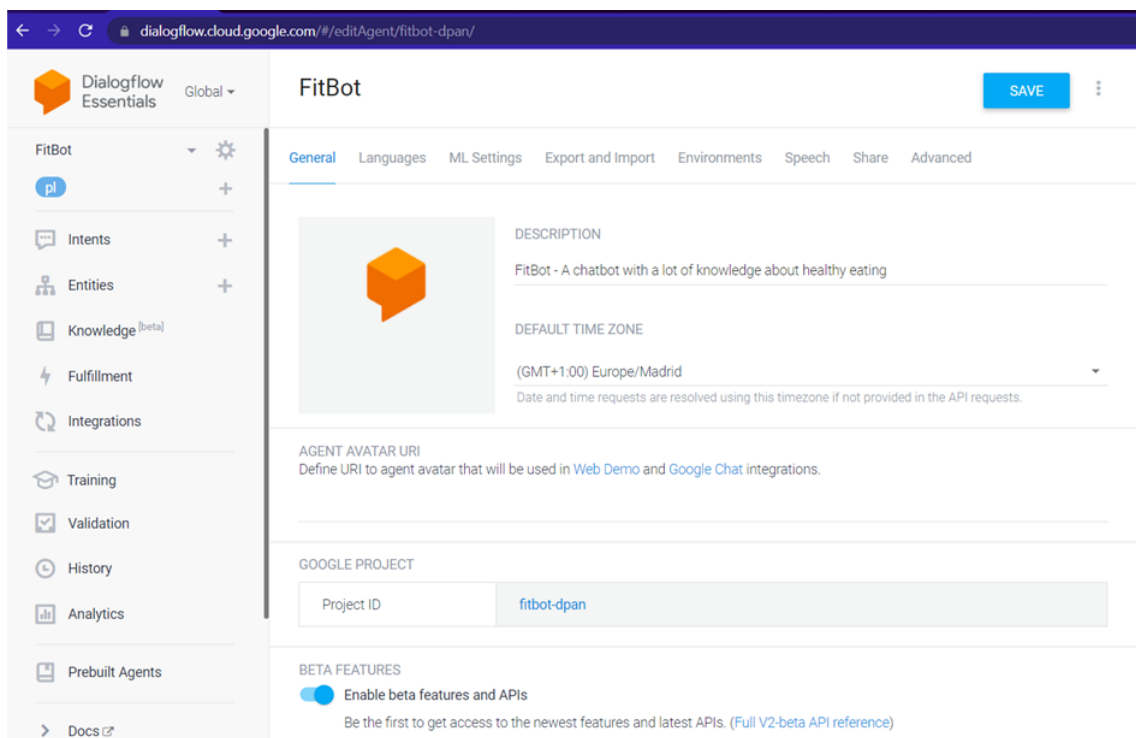
- dopasowanie diety dla użytkownika
- przedstawienie zasad zdrowego odżywiania
- opisanie wartości odżywczych dla konkretnych produktów
- zaprezentowanie rodzajów możliwych diet
- dostarczenie informacji na temat błędów podczas żywienia
- opcja stworzenia dowolnego posiłku.

Wykorzystane frameworki posiadają inne rozwiązania, opcje oraz ograniczenia, dlatego powyższe aspekty zostały zaimplementowane na różne sposoby, w oparciu o dostępne możliwości oferowane przez każde z narzędzi. W przypadku RASY oraz ChatterBota w celu zapewnienia użytkownikom przyjaznego środowiska do prowadzenia konwersacji, jakim jest czat dostępny na stronie internetowej, wykorzystano Flaska [22]. Jest to mikro-framework języka Python, którego używa się do tworzenia małych aplikacji

internetowych. Jego działanie oparte jest na znanym silniku szablonowania o nazwie Jinja2, który łączy w sobie system szablonów internetowych wraz ze źródłami danych do dynamicznych witryn internetowych.

3.2.1. Dialogflow

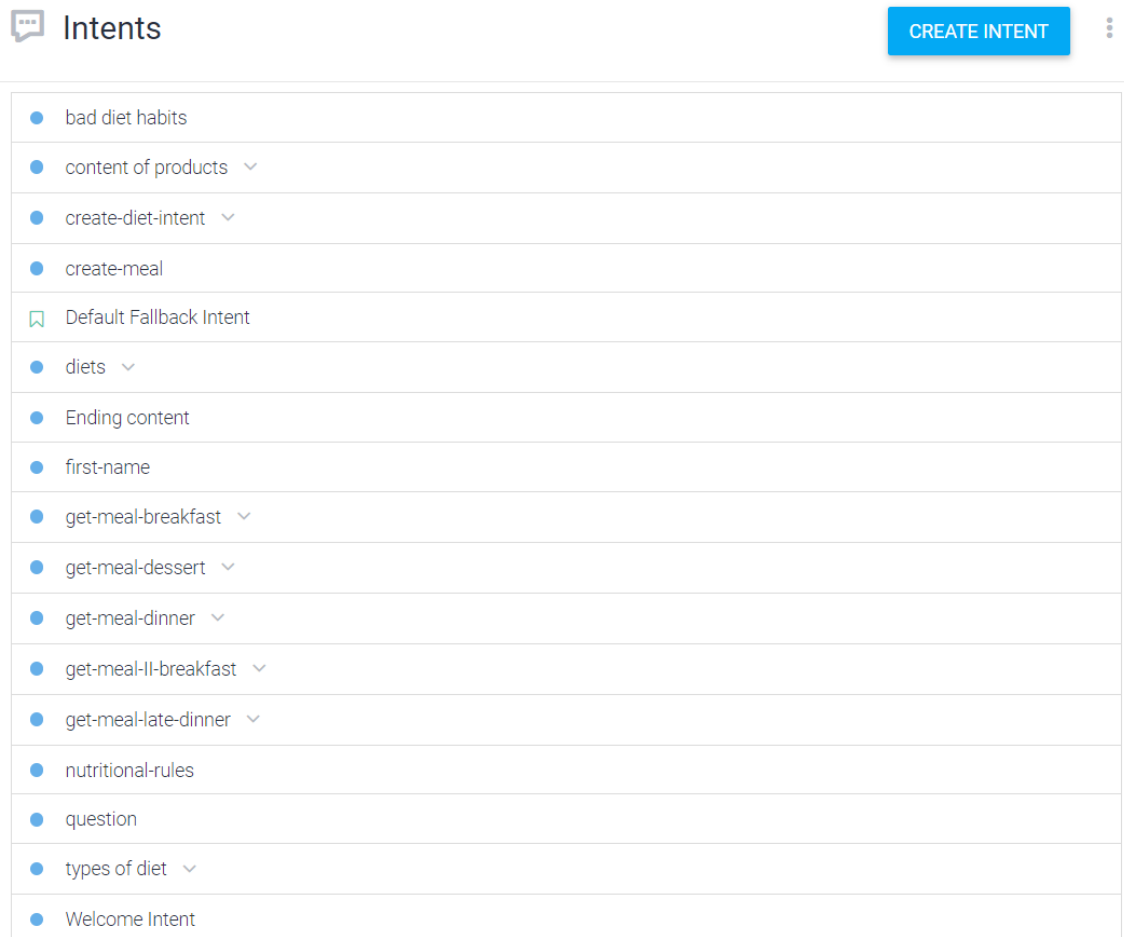
Stworzenie bota za pomocą Dialogflow było bardzo prostym zadaniem. Wystarczyło zalogować się na stronie <https://dialogflow.com/> (Rysunek 14) i wybrać odpowiednie ustawienia. W Dialogflow chatbot określany jest jako Agent.



Rysunek 14 Panel do tworzenia i obsługi chatbota w narzędziu Dialogflow (źródło: opracowanie własne)

Najważniejsze podczas konstruowania Agentu w Dialogflow jest zdefiniowanie odpowiednich intencji (*Ententy*) oraz encji (*Entities*). Intencje reprezentują to, jak bot interpretuje wypowiedzi użytkowników. Wiele wyrażen może oznaczać dokładnie to samo, pomimo że są inaczej zapisane, zawierają synonimy poszczególnych słów lub szyk wyrazów jest odmienny. Znaczenie danego wyrażenia jest opisywane jako intencja, czyli to, co użytkownik chce przekazać Agentowi. Dzięki wykorzystaniu intencji można w łatwy sposób pogrupować wiele wyrażen w jedno. Do stworzenia FitBota konieczne było zdefiniowanie wielu różnych intencji, dopasowanych do poszczególnych sytuacji

i możliwych wariantów pytań, zadawanych przez użytkowników. Stworzenie różnych intencji powoduje, że nie trzeba uczyć chatbota, jak ma reagować na każde jedno wyrażenie. Zamiast tego, można skategoryzować wyrażenia w intencje, z którymi bot będzie w stanie poradzić sobie bez problemu (Rysunek 15).



Rysunek 15 Przykładowe intencje stworzone do wytrenowania FitBota (źródło: opracowanie własne)

Na poniższym obrazku przedstawiono przykładową intencję dotyczącą przypadku, kiedy użytkownik chciałby znaleźć dopasowaną do siebie dietę (Rysunek 16).

• create-diet-intent

SAVE

exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available [system](#) or [custom](#) entity types.

” Add user expression
” chcę zmienić dietę
” chcę ułożyć dietę
” potrzebuję diety
” jaka dieta dla mnie
” chcę znaleźć dietę dla siebie
” przygotuj mi dietę
” potrzebuję diety dla mnie
” znajdź dietę dla mnie
” dieta dla mnie
” zrób mi dietę

Rysunek 16 Przykładowa konstrukcja intencji w Dialogflow (źródło: opracowanie własne)

Wypisane zostały różne możliwości zdań, jakie mogłyby zostać wypowiedziane w takiej sytuacji. Dzięki temu FitBot będzie w stanie dużo łatwiej nauczyć się i zinterpretować o co chodzi użytkownikowi i jak powinien na to zareagować. Encje (Rysunek 17) przeznaczone są do identyfikowania istotnych danych wejściowych w języku naturalnym. Pozwalają one na wydobycie konkretnych informacji od użytkowników. Przykładowo, w sytuacji, kiedy dana osoba będzie chciała stworzyć dla siebie posiłek i zostanie poproszona o podanie owoców, które lubi jeść, chatbot będzie mógł skorzystać ze zdefiniowanych encji, aby być w stanie odpowiednio zinterpretować odpowiedź i dopasować do możliwych opcji.

fruit

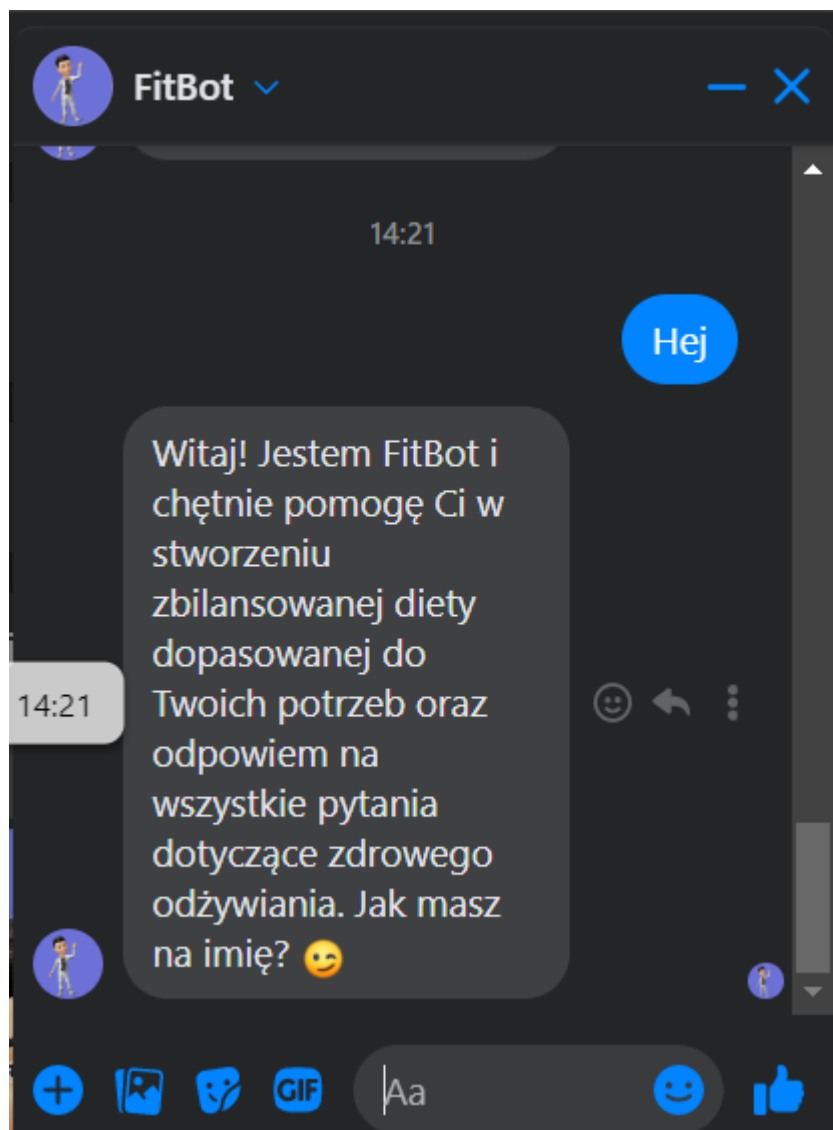
SAVE

Define synonyms ⓘ Regexp entity ⓘ Allow automated expansion Fuzzy matching ⓘ

jabłko	jabłko, jablko
truskawka	truskawka
malina	malina
banan	banan
pomarańcza	pomarańcza
gruszka	gruszka
arbuz	arbuz
kiwi	kiwi
winogrona	winogrona
mango	mango
ananas	ananas
brzoskwinia	brzoskwinia

Rysunek 17 Przykładowe zdefiniowanie encji dla owoców (fruit) (źródło: opracowanie własne)

Domyślnie agent odpowiada na dopasowaną intencję statyczną odpowiedzią. Możliwe jest zapewnienie bardziej dynamicznej odpowiedzi poprzez użycie mechanizmu Fulfillment, dzięki któremu można samodzielnie tworzyć odpowiednie reakcje bota, używając do tego odpowiedniego języka programowania i różnych usług. Niestety w wersji darmowej Dialogflow Essential, która była używana do stworzenia FitBota taka opcja nie jest dostępna. Dialogflow oferuje wiele platform, z którymi można powiązać chatbota. FitBot został zintegrowany z jednym z najbardziej popularnych i wykorzystywanych obecnie komunikatorów, jakim jest Facebook Messenger. Dzięki takiemu rozwiązaniu każdy użytkownik Facebooka może łatwo wejść w interakcję z FitBotem w dowolnym momencie. Wystarczy rozpocząć z nim konwersację na czacie i zadać dowolne pytanie (Rysunek 18).



Rysunek 18 Przykład konwersacji z FitBotem za pośrednictwem Messengera
(źródło: opracowanie własne)

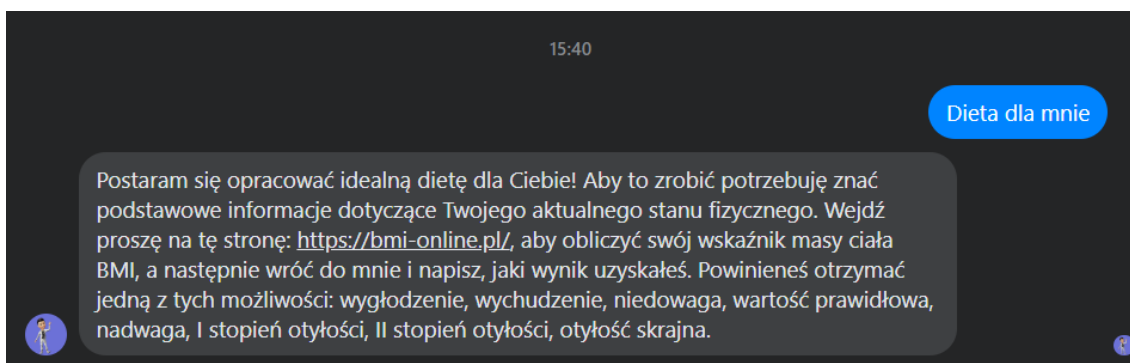
Użytkownik może zadać FitBotowi dowolne pytanie związane z dietą oraz zdrowym stylem życia albo wybrać odpowiednią kategorię z sześciu zaproponowanych przez agenta. Są one następujące:

- Dieta dla mnie
- Zasady odżywiania
- Skład produktów
- Rodzaje diet
- Błędy żywieniowe
- Stwórz posiłek

Poniżej przedstawione zostanie działanie każdej z opcji.

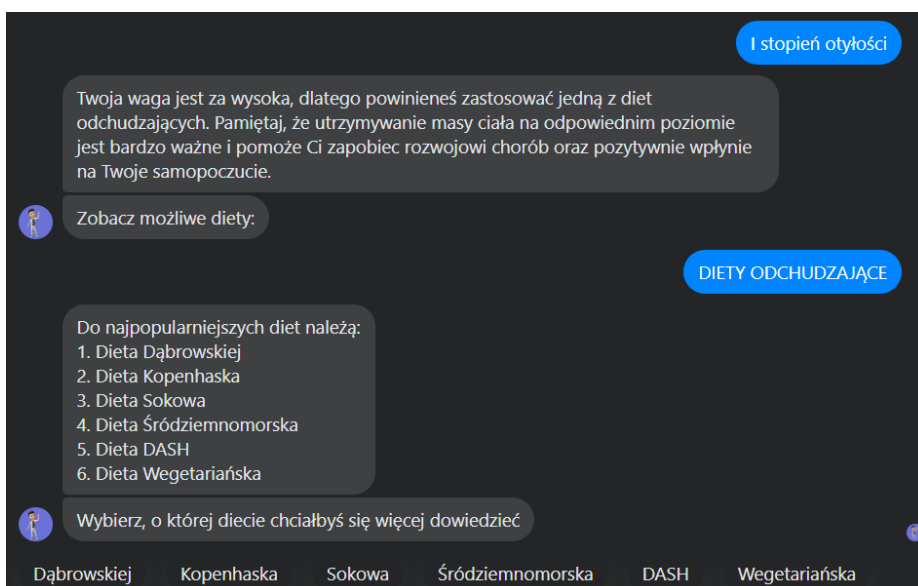
Dieta dla mnie

Po wybraniu przez użytkownika tej opcji, jest on w stanie dowiedzieć się więcej na temat tego, jaka dieta jest dla niego odpowiednia. Chatbot musi zdobyć informacje dotyczące stanu zdrowia danej osoby, aby mógł stwierdzić, co będzie dla niego najlepsze (Rysunek 19).



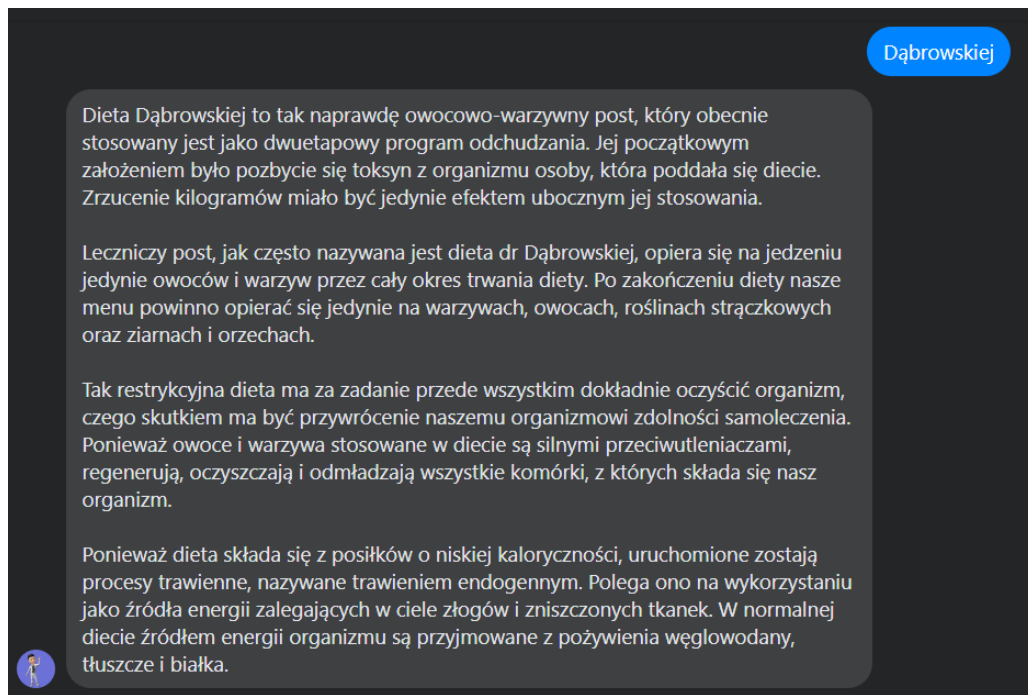
Rysunek 19 Przykład odpowiedzi chatbota związanej z dietą dla użytkownika (źródło: opracowanie własne)

Po zadaniu istotnych pytań FitBot jest w stanie doradzić, jaka dieta powinna być stosowana oraz wskaże produkty, które warto spożywać, podejmując każdą z nich. Przykładowo, poniżej pokazano, co chatbot doradził osobie posiadającej I stopień otyłości (Rysunek 20).

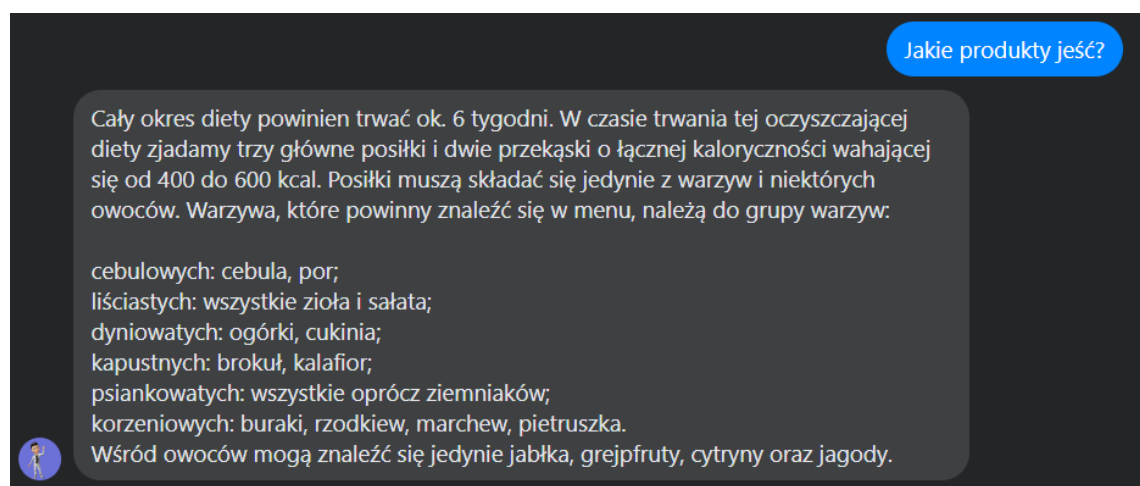


Rysunek 20 Odpowiedź chatbota, jaka dieta będzie najlepsza dla użytkownika (źródło: opracowanie własne)

Chatbot wymienił kilka z najpopularniejszych diet odchudzających i użytkownik może dowiedzieć się więcej o każdej z nich, aby następnie wybrać, która będzie dla niego najbardziej odpowiednia. Przykładowo, po wybraniu diety Dąbrowskiej (Rysunek 21, Rysunek 22):



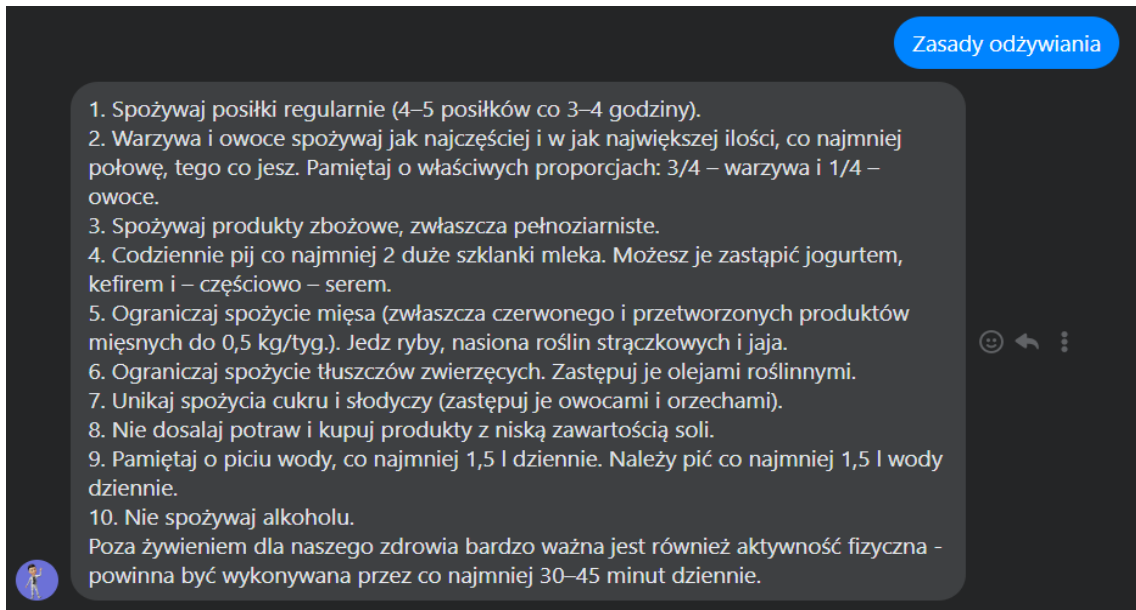
Rysunek 21 Odpowiedź chatbota dotycząca informacji o diecie Dąbrowskiej (źródło: opracowanie własne)



Rysunek 22 Odpowiedź chatbota związana ze szczegółowym pytaniem użytkownika o dietę Dąbrowskiej (źródło: opracowanie własne)

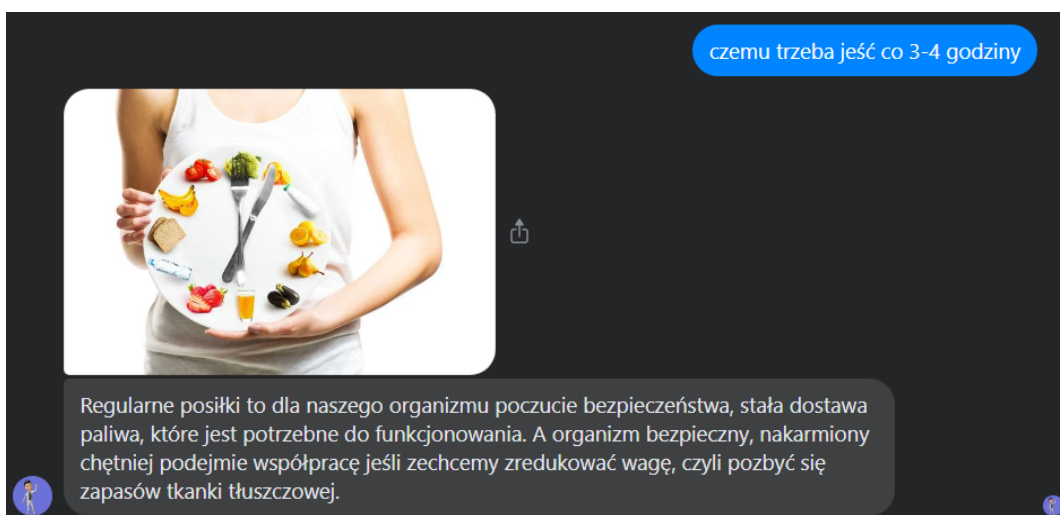
Zasady odżywiania oraz Błędy żywieniowe

Po wybraniu tej opcji użytkownik dostaje spis najważniejszych wskazówek dotyczących zasad odżywiania i błędów popełnianych podczas jedzenia (Rysunek 23, Rysunek 26). Może następnie zadawać chatbotowi konkretne pytania dotyczące otrzymanych informacji, aby je lepiej zrozumieć lub zapytać o inne zasady lub błędy, które nie zostały wspomniane przez bota (Rysunek 24, Rysunek 25).



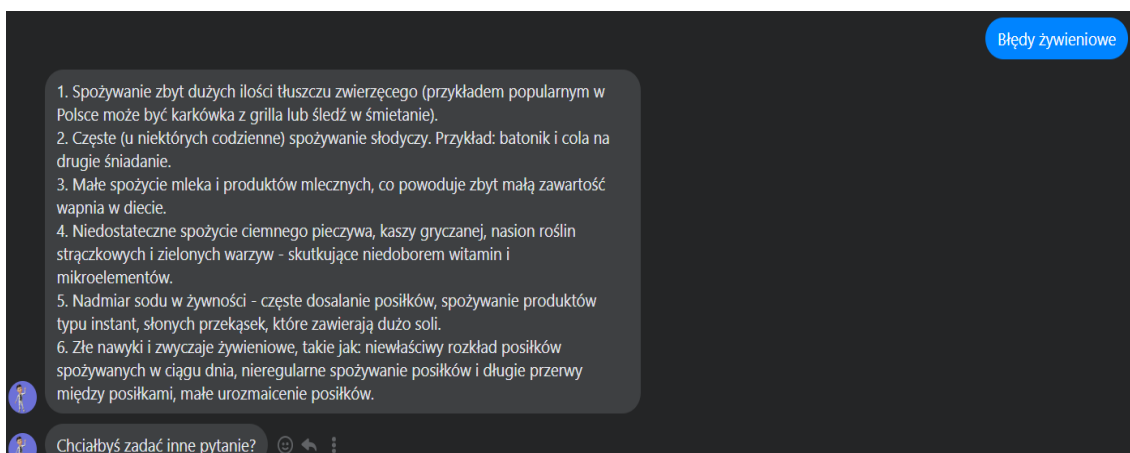
Rysunek 23 Wskazówki udzielane przez bota, związane ze zdrowym odżywianiem

(źródło: opracowanie własne)

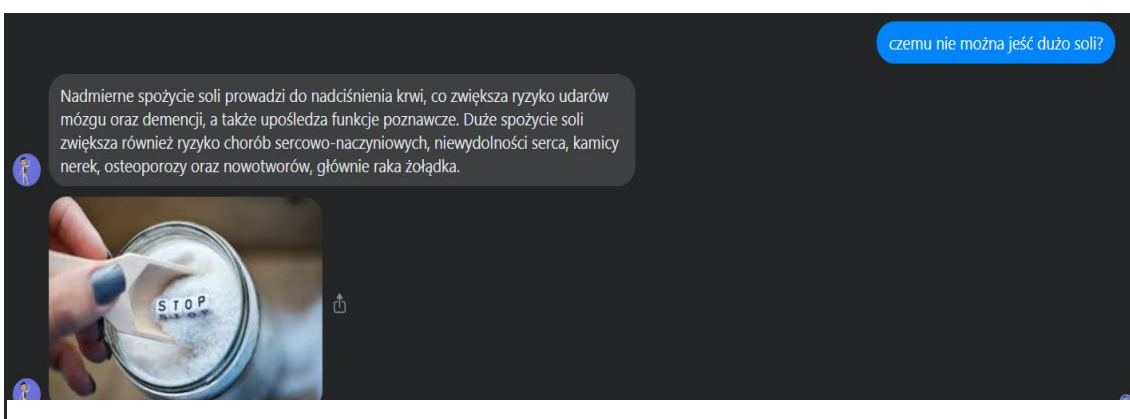


Rysunek 24 Odpowiedź bota na pytanie dotyczące jednej z zasad zdrowego odżywiania

(źródło: opracowanie własne)



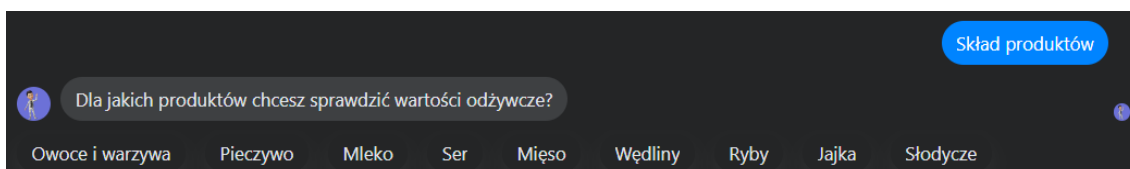
Rysunek 26 Informacje udzielone przez bota odnośnie do błędów żywieniowych (źródło: opracowanie własne)



Rysunek 25 Odpowiedź bota na szczegółowe pytanie dotyczące jednego z błędów w żywieniu (źródło: opracowanie własne)

Skład produktów

Ta opcja umożliwia sprawdzenie wartości odżywczych dla produktów, należących do różnych kategorii (Rysunek 27).



Rysunek 27 Sugerowane przez bota możliwości produktów, dla jakich można sprawdzić wartości odżywcze (źródło: opracowanie własne)

Chatbot wyświetla użytkownikowi tabele zawierające informacje na temat wartości odżywczych produktów (kalorie, białka, tłuszcze, węglowodany, wapń, fosfor, żelazo) oraz witamin.

Przykładowa odpowiedź bota dotycząca wartości odżywczych mięsa znajduje się poniżej (Rysunek 28, Rysunek 29).

wartość odżywcza w 100g produktu (część jadalna)	fragment tuszki mięsa indyczego								
	tuszka	pierś		podudzie		udziec		skrzydło	
		ze skórą	bez skóry	ze skórą	bez skóry	ze skórą	bez skóry	ze skórą	bez skóry
energia [kcal]	129	100	84	142	100	131	104	168	115
białko [g]	17	18,7	19,2	15,7	16,6	18,4	19,4	16,5	18
tłuszcz [g]	6,8	2,7	0,7	8,9	3,7	6,4	2,8	11,4	4,7
SFA [g]	2,22	0,64	0,22	2,54	1,12	1,83	0,87	3,23	1,42
MUFA [g]	1,67	1,01	0,33	4,08	1,65	2,92	1,29	5,21	1,29
PUFA [g]	2,14	0,47	0,15	1,89	0,77	1,34	0,61	2,43	0,98
sód [g]	63	48	47	92	48	76	77	63	62
potas [g]	300	441	460	324	441	384	414	308	359
cynek [mg]	2	0,8	0,63	2,53	2,83	2,46	2,64	1,98	2,27
witamina B ₂	0,2	0,15	0,15	0,2	0,21	0,26	0,27	0,14	0,15
witamina B ₆	0,4	0,6	0,59	0,28	0,3	0,28	0,3	0,19	0,2
witamina B ₁₂	0,4	0,7	0,7	1,7	1,7	1,7	1,7	0,7	0,6
witamina D	1,4	0,3	0	0,4	0,1	0,4	0,2	0,5	0,7

Węglowodany [g]	0	0	0	0	0	0	0	0
Żelazo [mg]	1,2	0,6	0,7	0	1	0,8	4,5	
Wit. B12 [ug]	0,3	0,4	0,4	0,7	0,4	0,4	0,5	
Wit. B1 [mg]	0,1	0,05	0,1	0,04	0,1	0,06	0,3	
Wit. B3 [mg]	6,8	6	11,2	4,9	5,8	4	6,9	

Mięso – 100 g	Kalorie	Białko	Tłuszcz
Wieprzowina mielona, gotowana	297 kcal	25.7 g	20.8 g
Wołowina mielona, duszona	217 kcal	26.1 g	11.8 g
Kurczak gotowana pierś	165 kcal	31 g	3.6 g
Jagnięcina pieczony udziec	258 kcal	25.6 g	16.5 g
Dzik gotowana połówka	160 kcal	28.3 g	4.4 g

Rysunek 28 Przekazane przez bota informacje dotyczące wartości odżywczych dla mięsa (źródło: opracowanie własne)

Rodzaj mięsa	Energia (kcal)	Białko (g)	Tłuszcz (g)	Cholesterol (mg)
Dzik	114-125	17,1-24,5	0,4-17,7	60-90
Jeleń	108-120	19,9-24,7	0,9-3,3	44
Sarna	98-113	19,2-24,6	0,3-3,7	40
Łoś	145	22,9	1,8	67
Daniel	95-110	21,0-23,1	0,5-1,4	110
Królik	100-144	21,3-22,7	2,6-7,2	65
Zając	107	23,5-24,8	1,1-2,3	49
Bażant	140	21,8-25,4	0,43-1,9	101
Kuropatwa	121	22,7	3,0	150
Kaczka dzika	197	17,1	12,0	140
Kurczak	158	10,7-21,5	1,3-9,3	75-84
Indyk	129	15,7-19,4	2,7-11,4	49-81
Wołowina	113-217	20,6-22,0	3,6-15,7	56-75
Wieprzowina	174-510	18,3-22,0	10,0-67,8	66-205

Chcesz zobaczyć wartości odżywcze innych produktów czy zadać inne pytanie?

Inne produkty Pytanie

Rysunek 29 Przekazane przez bota dane dotyczące wartości odżywczych dla mięsa – c.d. (źródło: opracowanie własne)

wartości odżywcze chleba razowego

Poniżej znajdziesz tabele z wartościami odżywczymi dla wybranego pieczywa:

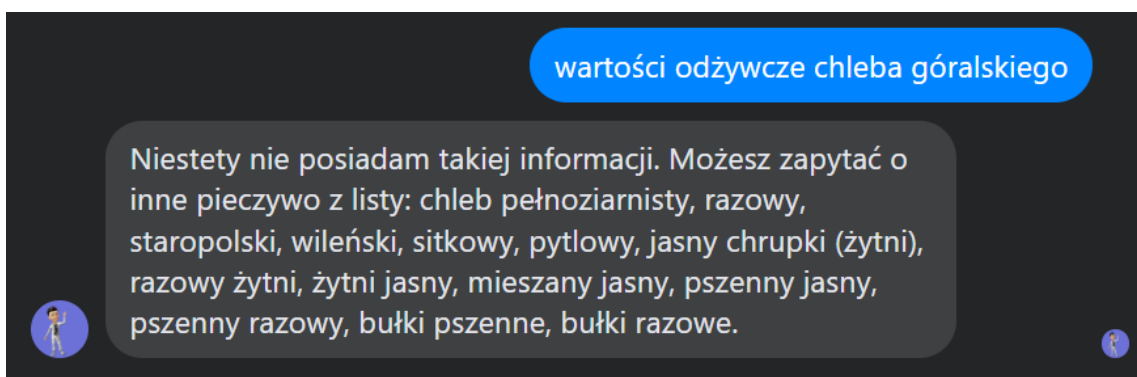
Wartość odżywcza pieczywa żytniego (w przeliczeniu na 100 g chleba)

Składnik odżywczy	Rodzaj chleba						
	pełnoziarnisty	razowy	staropolski	wileński	sitkowy	pyłtowy	jaśny
białko [g]	6,7	5,6	5	4,4	5,2	4,4	4,0
tłuszcz [g]	1,8	1,7	1,7	1,5	1,5	1,5	1,3
węglowodany ogółem [g]	53,9	51,5	50,5	52,3	53,2	56,2	57,5
skrobia [g]	34,4	35,0	39,3	39,8	42,0	46,9	49,9
błonnik [g]	6,1	5,9	5,3	5,2	5,3	4,9	4,7
potas [mg]	356	279	232	188	217	147	115
wapń [mg]	66	25	22	19	23	16	14
fosfor [mg]	245	204	166	125	159	92	68
magnez [mg]	71	64	52	37	49	28	19
żelazo [mg]	2,5	2,3	2,0	1,4	1,8	1,1	0,80
cynek [mg]	2,86	2,54	2,34	1,70	2,12	1,15	0,77
mangan [mg]	3,01	2,74	2,54	1,86	2,36	0,97	0,87
witamina E [mg]	0,89	0,86	0,75	0,39	0,69	0,97	0,52
witamina B ₁ [mg]	0,212	0,184	0,160	0,132	0,155	0,147	0,128
witamina B ₂ [mg]	0,172	0,103	0,099	0,076	0,064	0,070	0,050
witamina PP [mg]	0,82	0,71	0,66	0,67	0,66	0,56	0,47

Rysunek 30 Odpowiedź bota związana z wartościami odżywczymi chleba razowego (źródło: opracowanie własne)

FitBot jest w stanie sam sklasyfikować, w jakiej kategorii znajduje się produkt, o który użytkownik zapyta. Na przykładzie poniżej pokazano, jak zapytano go o wartości odżywcze chleba razowego (Rysunek 30).

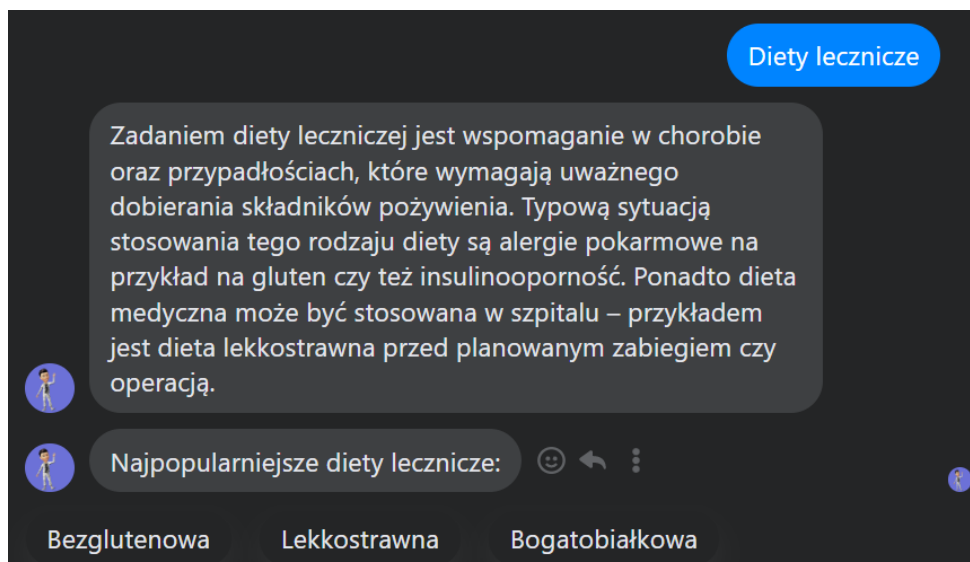
Baza wiedzy chatbota jest ograniczona, nie posiada on informacji na temat wszystkich dostępnych rodzajów produktów. Jeśli użytkownik zapyta przykładowo o typ chleba, o którym bot nie posiada danych, chatbot zwróci mu odpowiedni komunikat, informując go o wszystkich dostępnych typach pieczywa, o których jest w stanie mu coś opowiedzieć (Rysunek 31).



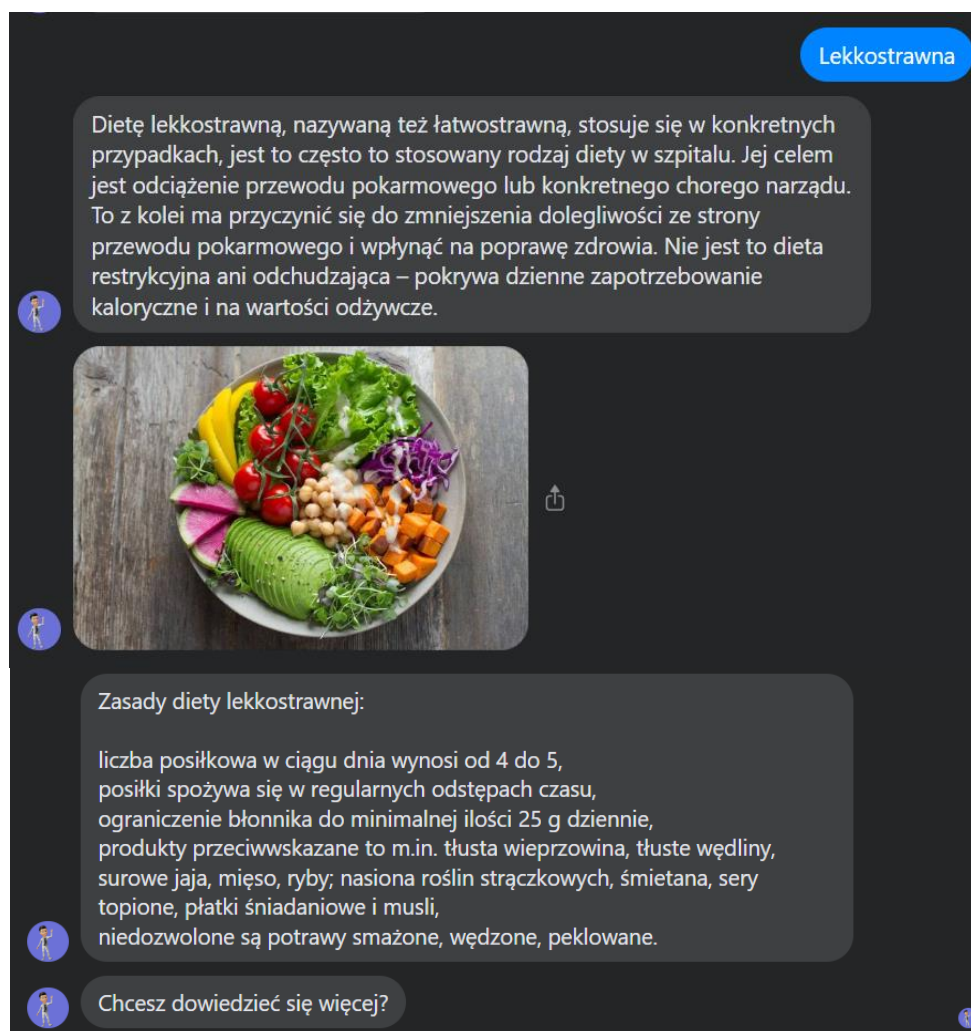
Rysunek 31 Odpowiedź bota w przypadku, kiedy nie rozpozna produktu, o jaki pyta użytkownik (źródło: opracowanie własne)

Rodzaje diet

Po wybraniu tej opcji użytkownik może dowiedzieć się czegoś na temat możliwych typów diet – ich szczegółowe opisy, cel oraz zastosowanie. Chatbot posiada wiedzę na temat najbardziej podstawowych i popularnie stosowanych typów diet. Poniżej przedstawiono odpowiedź chatbota, gdy użytkownik chce dowiedzieć się więcej na temat diety leczniczej (Rysunek 32, Rysunek 33).



Rysunek 32 Informacje przekazane przez bota dotyczące diet leczniczych (źródło: opracowanie własne)



Rysunek 33 Odpowiedź dotycząca diety leczniczej – ciąg dalszy (źródło: opracowanie własne)

Stwórz posiłek

Jest to najbardziej rozbudowana opcja FitBota. Pozwala na znalezienie oraz zaproponowanie użytkownikowi najbardziej odpowiedniego posiłku, w oparciu o jego preferencje i wymagania. Na poniższych obrazkach pokazano, jak chatbot zasugerował użytkownikowi kilka opcji zdrowego śniadania po otrzymaniu informacji dotyczących jego oczekiwań (Rysunek 34).

Śniadanie

Super, w takim razie teraz przedstawię Ci wszystkie możliwe opcje i zadam kilka pytań odnośnie Twoich preferencji. Dzięki temu śniadanie będzie najlepsze z możliwych i z pewnością zadowoli Twoje kubki smakowe. Zatem zaczynamy 😊



Wolisz śniadanie na słodko czy na słono?

SŁODKO



Dobrze, a jakie produkty lubisz albo chcesz wykorzystać?

banan, truskawka

Super, w takim razie myślę, że dobrym pomysłem byłaby sałatka, omlet lub owsianka. Zobacz moje poniższe propozycje i wybierz, co Ci się najbardziej podoba.



Sałatka owocowa

Pyszna, kolorowa i zdrowa sałatka owocowa.

ZOBACZ PRZEPIS



Puszysty omlet z owocami

Smakuje jak najłżejszy biszkopt upieczony w piekarniku!



Owsianka z owocami

Pyszna owsianka z owocami sezonowymi szykowana na jogurcie naturalnym jest nie t...

ZOBACZ PRZEPIS



Chciałbyś jeszcze się czegoś dowiedzieć?



Pytanie

Rysunek 34 Konwersacja z botem dotycząca znalezienia użytkownikowi posiłku, którego potrzebuje (źródło: opracowanie własne)

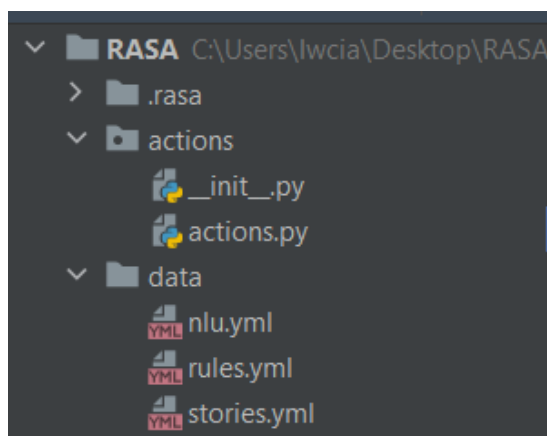
Bazując na preferencjach, chatbot jest w stanie znaleźć odpowiedni scenariusz i najbardziej pasujące produkty. Może się jednak zdarzyć, że użytkownik wpisze produkt, który lubi albo chce wykorzystać, a nie będzie on pasował do żadnego ze zdefiniowanych botowi schematów. W takiej sytuacji FitBot zaproponuje rozwiązanie, które będzie wydawać się mu najbardziej zbliżone do oczekiwanego.

3.2.2. Rasa

FitBot stworzony za pomocą Rasy jest w stanie dostarczyć użytkownikowi odpowiedzi na wszelkie pytania dotyczące zdrowego odżywiania i oferuje pomoc dotyczącą wszystkich z kategorii wymienionych na początku rozdziału, które zostaną szczegółowo opisane poniżej. W celu zapewnienia przyjemnego i wygodnego prowadzenia konwersacji z botem, wykorzystano dodatkowo framework Flask, dzięki czemu użytkownik jest w stanie prowadzić rozmowę poprzez czat na stronie internetowej.

Stworzenie chatbota za pomocą Rasy wymagało stworzenia odpowiednich plików. Do najważniejszych z nich należą `nlu.yml`, `rules.yml`, `stories.yml` oraz `actions.py` (Rysunek 35).

- **nlu.yml** – plik ten zawiera dane szkoleniowe, które składają się z przykładowych wypowiedzi użytkowników, które są pogrupowane ze względu na konkretne intencje. Zawiera też encje, czyli fragmenty informacji, które można wyciągnąć z wypowiedzi użytkownika.



Rysunek 35 Struktura najważniejszych plików, potrzebnych do stworzenia bota za pomocą Rasy (źródło: opracowanie własne)


```

- intent: stwórz_posiłek
examples: |
  - co mogę zjeść na [śniadanie](posiłek)
  - co zjeść na [śniadanie](posiłek)
  - co zjeść na [obiad](posiłek)
  - co najlepiej zjeść na [obiad](posiłek)
  - co zjeść na [podwieczorek](posiłek)
  - co zjeść na [kolację](posiłek)
  - posiłek na [kolację](posiłek)
  - co zjeść na [drugie śniadanie](posiłek)
  - co zjeść na [II śniadanie](posiłek)
  - co mogę zjeść na [obiad](posiłek)
  - pomysł na [śniadanie](posiłek)
  - co zjeść na [deser](posiłek)

```

Rysunek 36 Przykładowy intent dotyczący tworzenia posiłku (źródło: opracowanie własne)

Na powyższym rysunku (Rysunek 36) przedstawiono przykładowy intent: stwórz posiłek. Dotyczy on przypadku, kiedy użytkownik chce dowiedzieć się, co mógłby zjeść, jaki posiłek byłby najlepszy na daną porę. Zapisane w nawiasach kwadratowych nazwy posiłków: śniadanie, obiad itp. oznaczają encje, które są wydobywane ze zdania przez chatbota, dzięki czemu mogą być potem przez niego wykorzystane podczas dalszej rozmowy.

- **rules.yml** – w tym pliku przedstawione są reguły działania, jakie chatbot powinien realizować w zależności od intencji. Reguły opisują małe fragmenty konwersacji, które powinny zawsze realizować tą samą ścieżkę (Rysunek 37).

```

- rule: Powiedz, czy użytkownik chce o coś zapytać, jeśli użyje słów okej, dobrze itp.
  steps:
    - intent: potwierdzenie
    - action: action_potwierdzenie

- rule: Powiedz, jakie są zasady zdrowego odżywiania
  steps:
    - intent: zasady_zdrowego_odzywiania
    - action: action_podaj_zasady

```

Rysunek 37 Fragment pliku rules.yml, w którym definiowane są reguły działania chatbota (źródło: opracowanie własne)

- **stories.yml** – plik ten, tak samo jak **rules.yml** jest reprezentacją dialogów pomiędzy użytkownikiem a botem. Historie używane są do szkolenia modelu uczenia maszynowego, aby zidentyfikować wzorce w wypowiedziach i zbudować uogólnienia na niewidziane etapy konwersacji.

```
- story: Co zjeść na wybrany posiłek
  steps:
    - intent: stwórz_posiłek
    - action: action_stwórz_posiłek
    - intent: słono_słodko
    - action: action_zatwierdź_typ_posiłku
    - action: action_jakie_produkty_wykorzystać
    - intent: produkty
    - action: action_zatwierdź_produkty
    - action: action_podaj_propozycje
```

Rysunek 38 Zdefiniowanie dialogu za pomocą intencji oraz akcji w pliku **stories.yml**
(źródło: opracowanie własne)

Jak widać na powyższym rysunku (Rysunek 38), historie są dużo bardziej rozbudowane, składają się z wielu intencji i akcji. Rasa oferuje możliwość tworzenia własnych akcji, które opisują zachowanie chatbota w zależności od danej sytuacji. Dzięki zastosowaniu akcji możliwe jest wydobywanie informacji od użytkownika i następnie wykorzystanie ich w dalszej części rozmowy. Dzięki temu konwersacja jest dużo bardziej rozbudowana i chatbot jest w stanie odpowiadać znacznie efektywniej, bazując na indywidualnych potrzebach danej osoby. Akcje definiowane są w pliku **actions.py**, przykładowa akcja została przedstawiona na poniższym rysunku (Rysunek 39).

```

class ActionWybranyPosiłek(Action):
    def name(self) -> Text:
        return "action_wybrany_posiłek"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
        current_food = next(tracker.get_latest_entity_values("posiłek"), None)
        next_step = tracker.get_slot("posiłek")
        if current_food == "kolacja":
            current_food == "kolację"
            msg = f"Zaraz spróbuję znaleźć dla Ciebie coś pysznego. Wolisz zjeść {current_food} na słońco czy na słodko?"

            dispatcher.utter_message(text=msg)

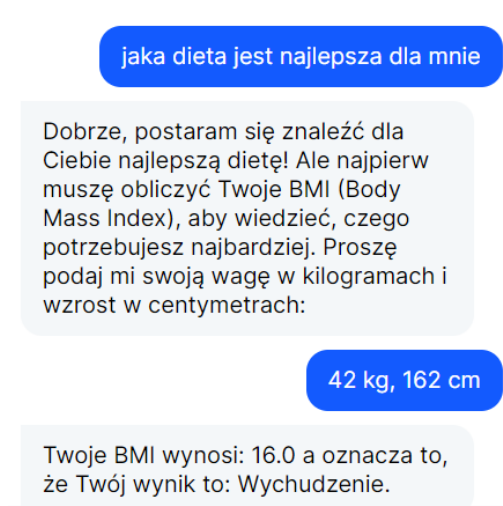
        return []

```

Rysunek 39 Przykład akcji, zdefiniowanej w pliku actions.py (źródło: opracowanie własne)

Opis realizacji głównych tematów, związanych z prowadzeniem zdrowej diety:

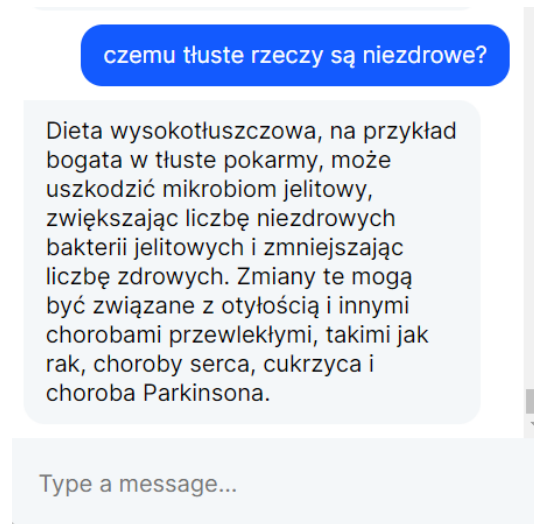
- Dopasowanie odpowiedniej diety użytkownikowi – chatbot stworzony za pomocą Rasy po zdobyciu od użytkownika informacji związanych z jego wagą oraz wzrostem jest w stanie samodzielnie obliczyć jego BMI, aby następnie opowiedzieć jaka dieta będzie dla niego najlepsza (Rysunek 40).



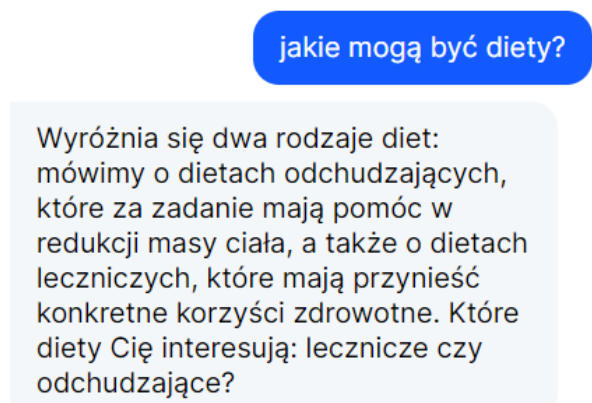
Rysunek 40 Odpowiedź bota stworzonego za pomocą Rasy dotycząca diety dla użytkownika (źródło: opracowanie własne)

- Wskazówki dotyczące zdrowego odżywiania i błędów podczas jedzenia oraz rodzaje diety

FitBot posiada dużą wiedzę dotyczącą tego, jak powinno się odżywiać na co dzień i czego unikać. Użytkownik może zadawać wiele pytań z tym związanych oraz dopytywać o konkretne zasady lub błędy, aby lepiej je zrozumieć (Rysunek 41, Rysunek 42).



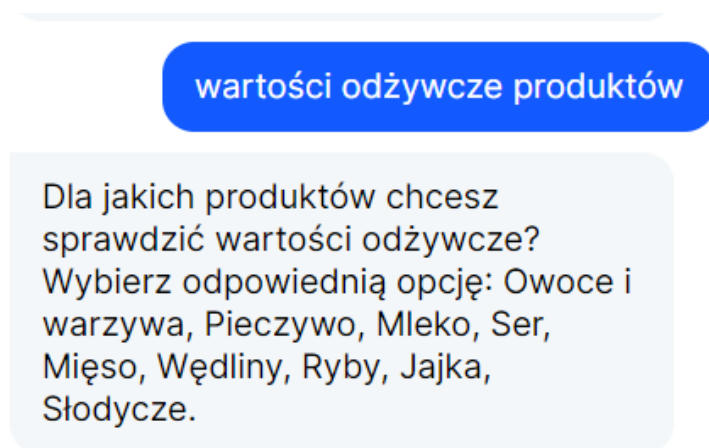
Rysunek 41 Odpowiedź bota na szczegółowe pytania dotyczące zdrowego odżywiania (źródło: opracowanie własne)



Rysunek 42 Odpowiedź chatbota na pytanie związane z dietami (źródło: opracowanie własne)

- Informacje dotyczące wartości odżywczych produktów

FitBot posiada bogatą wiedzę na temat wartości odżywczych wielu popularnie stosowanych produktów. Wyświetla użytkownikowi odpowiedź w formie tabel i zdjęć, co ułatwia ich odczytanie i jest dużo przyjemniejsze niż gdyby było to podane w formie surowego tekstu (Rysunek 43, Rysunek 44, Rysunek 45).



Rysunek 43 Odpowiedź FitBota dotycząca wartości odżywczych produktów (źródło: opracowanie własne)

Ser

Rodzaj sera	Energia (kcal)	Białko (g)	Tłuszcz (g)	Wapń (mg)	Witamina A (µg)	Witamina B2 (mg)
Brie	330	20,7	27,7	600	285	0,52
Camembert	300	19,8	24,3	388	235	0,49
Mozzarella	281	19,4	21,6	517	258	0,24
Roquefort	369	21,5	30,6	560	520	0,59
Gorgonzola	357	25,0	28,6	714	350	0,43
Ementaler	383	28,8	29,7	835	297	0,38
Gruyere	413	29,8	32,3	1011	244	0,28
Edam	357	25,0	27,8	731	289	0,37
Gouda	356	24,9	27,4	700	276	0,33
Parmezan	392	35,7	25,8	1280	253	0,33

Rysunek 44 Wyświetlenie przez FitBota wartości odżywczych dotyczących sera w formie tabeli (źródło: opracowanie własne)

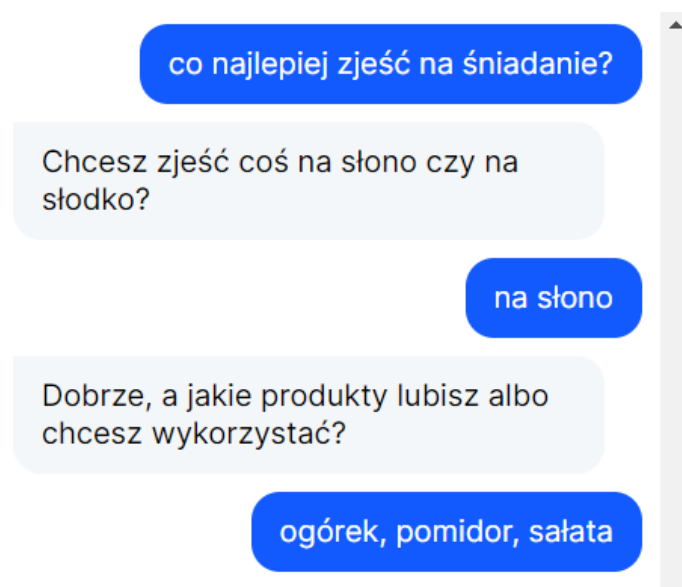
Słodczyce



Rysunek 45 Fragment odpowiedzi FitBota dotycząca wartości odżywczych słodczych (źródło: opracowanie własne)

- Pomoc w skomponowaniu zdrowego posiłku

FitBot jest w stanie zaproponować użytkownikowi najlepsze możliwe posiłki, w zależności od jego preferencji (Rysunek 46, Rysunek 47).



Rysunek 46 Konwersacja użytkownika z FitBotem dotycząca znalezienia posiłku na śniadanie (źródło: opracowanie własne)

Super, w takim razie myślę, że dobrym pomysłem byłaby sałatka, omlet lub owsianka. Zobacz moje poniższe propozycje i wybierz, co Ci się najbardziej podoba.



Rysunek 47 Ciąg dalszy konwersacji dotyczącej znalezienia posiłku użytkownikowi (źródło: opracowanie własne)

3.2.3. ChatterBot

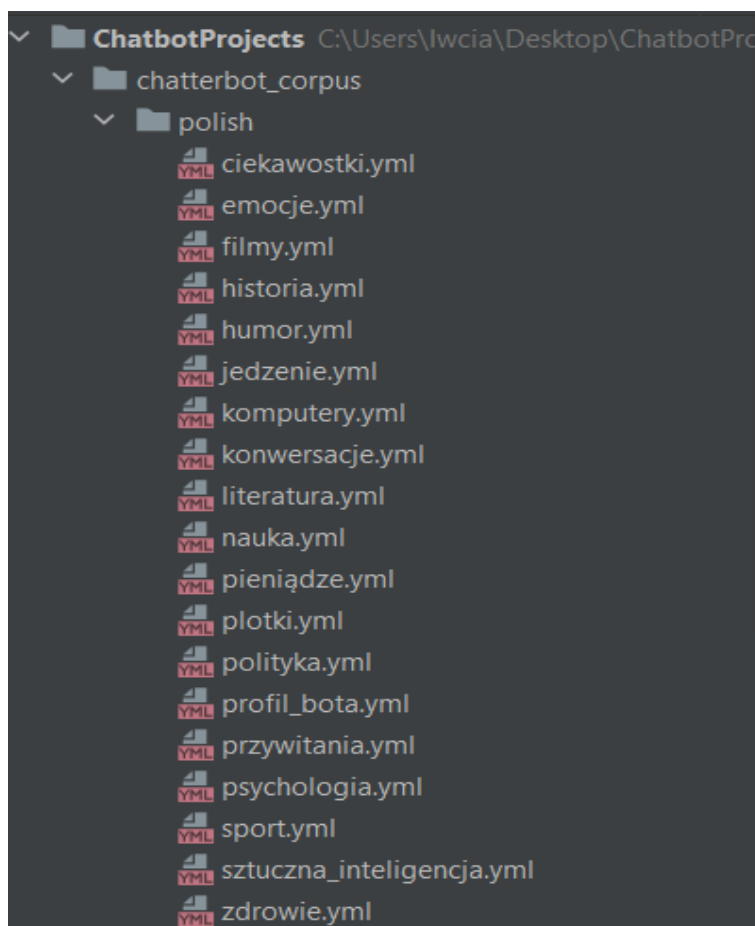
FitBot stworzony za pomocą ChatterBota, czyli biblioteki języka Python jest w stanie odpowiadać na wiele pytań dotyczących diety i zdrowego odżywiania. Tak jak w przypadku Dialogflow i Rasy, oferuje pomoc dotyczącą kwestii takich, jak:

- dopasowanie diety dla użytkownika
- dostarczenie informacji związanych z zasadami odżywiania oraz błędami podczas jedzenia
- znajomość rodzajów diet
- wiedza na temat wartości odżywczych produktów
- pomoc w stworzeniu zdrowego posiłku.

Wytrenowanie FitBota wymagało dostarczenia mu bardzo dużej ilości danych. Było to czasochłonne, ponieważ każda forma możliwego pytania musiała zostać zdefiniowana oddzielnie, zatem potrzeba było ogromnej ilości stworzonych dialogów, aby był on w stanie prawidłowo odpowiadać na pytania. Zadaniem chatbota jest generowanie automatycznych odpowiedzi na dane wpisywane przez użytkowników, na podstawie dostarczonych mu schematów konwersacji. Nie posiada on możliwości rozpoznawania kontekstu wypowiedzi i prowadzenia rozmowy na

podstawie otrzymanych informacji. ChatterBot posiada korpus danych dialogowych dla wielu języków. Nie zostały stworzone dialogi dla języka polskiego, dlatego samodzielnie stworzono odpowiedni zestaw konwersacji dla tego języka, aby na podstawie różnych wypowiedzi bot mógł się uczyć. W celu zapewnienia łatwej i przyjaznej interakcji użytkownika z botem, poprzez korzystanie z czatu na stronie internetowej, wykorzystano framework Flask.

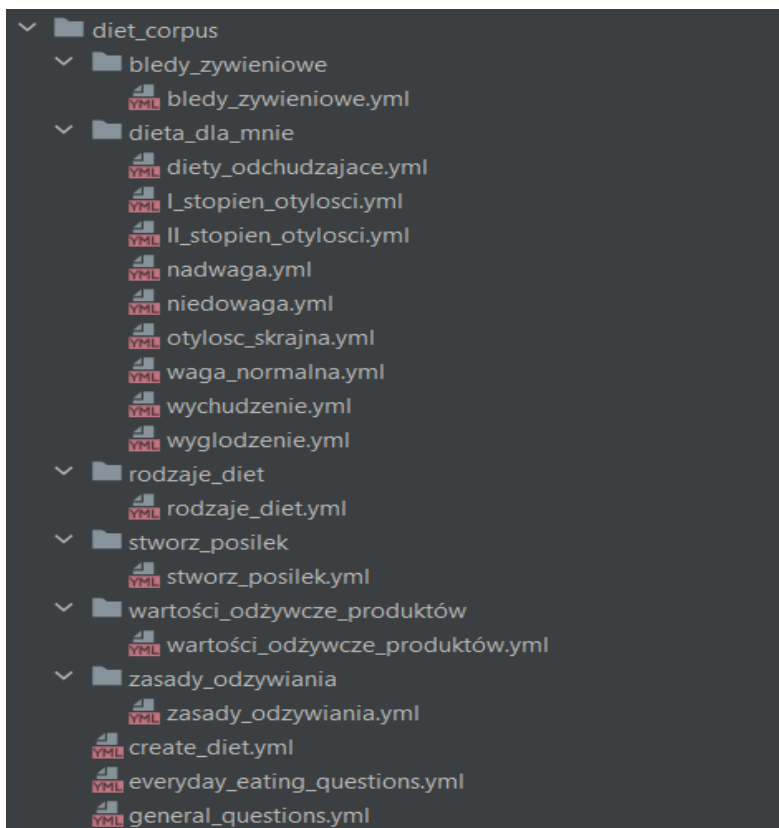
W folderze **chatterbot_corpus** znajduje się kilkanaście plików .yaml (Rysunek 48, Rysunek 49), w których zdefiniowano dialogi związane z podstawowymi kategoriami konwersacji, jakie często podejmowane są przez ludzi w życiu codziennym, np. dotyczące jedzenia, muzyki, emocji. Dzięki temu chatbot będzie w stanie odpowiedzieć nie tylko na pytania dotyczące zdrowego odżywiania i diety, ale też będzie potrafił podjąć rozmowę na inne tematy, co na pewno pozytywnie wpłynie na jego odbiór przez użytkowników.



Rysunek 48 Struktura plików, potrzebnych do stworzenia chatbota za pomocą narzędzia ChatterBot (źródło: opracowanie własne)


```
przywitania.yml x
1 categories:
2   - przywitania
3 conversations:
4   - Witaj
5   - Cześć!
6   - Cześć
7   - Witam!
8   - Hejka!
9   - Cześć!
10  - Cześć
11  - Hej! Jak tam u Ciebie?
12  - Dobrze a u Ciebie?
13  - To się cieszę. Chcesz zadać mi jakieś pytanie związane ze zdrowym odżywianiem?
14  - Hej
15  - Cześć. Jak się czujesz?
16  - Dobrze
17  - To super, cieszę się.
18  - Hej
19  - Cześć!
20  - Hej
21  - Witaj!
22  - Hej
23  - Hejka!
24  - Cześć, jak leci?
25  - Dobrze, a u Ciebie?
26  - Cześć, jak leci?
27  - Super, a u Ciebie?
```

Rysunek 49 Przykładowe dialogi, zdefiniowane dla kategorii przywitania (źródło: opracowanie własne)



Rysunek 50 Struktura folderu diet_corpus, w którym zdefiniowane zostały dialogi dla poszczególnych kwestii dotyczących diety (źródło: opracowanie własne)

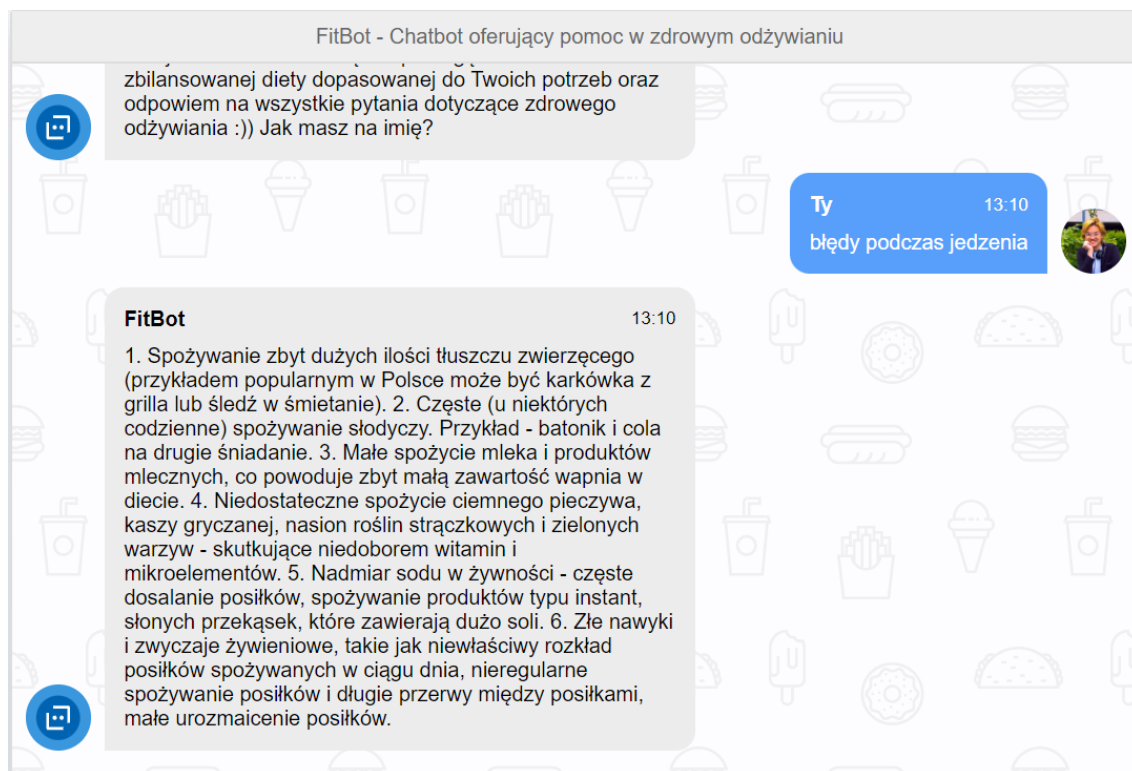
Folder **diet_corpus** został podzielony na kilka podfolderów, z których każdy zawiera pliki ze zdefiniowanymi dialogami, związanymi z poszczególnymi kwestiami dotyczącymi diety, w których chatbot jest w stanie pomóc użytkownikowi. Konieczne było stworzenie ogromnej ilości różnych przypadków tego, w jaki sposób dana osoba zada pytanie i jaka może być możliwa odpowiedź (Rysunek 50).

- Dopasowanie diety dla użytkownika

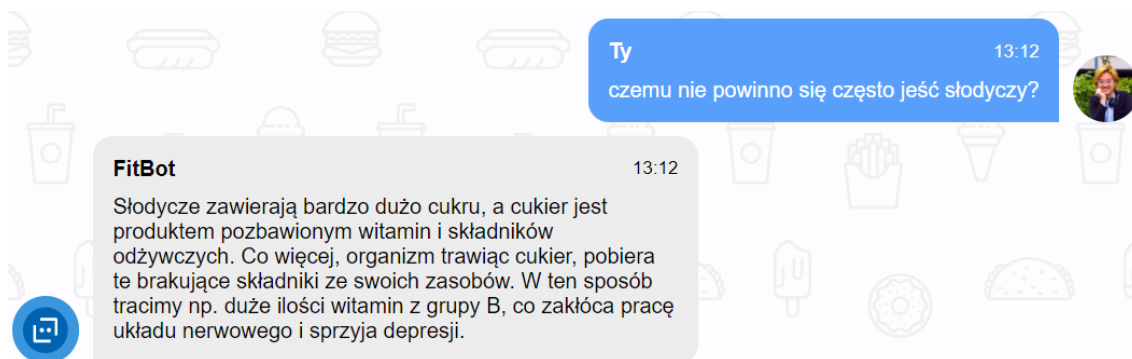
Ta funkcjonalność została zrealizowana bardzo podobnie, jak było to w przypadku Dialogflow. Wszystko powinno przebiec prawidłowo, dopóki użytkownik nie będzie popełniał żadnych błędów i literówek. Jeśli tak się zdarzy, bot może mieć problem z odpowiedzią, ponieważ nie posiada funkcji autokorekty i nie zrozumie wypowiedzi użytkownika. Jest on w stanie odpowiadać tylko na takie pytania, które zostały mu zdefiniowane, a w żadnych z nich nie pojawiły się błędy w pisowni, dlatego nie będzie w stanie ich rozpoznać.

- Dostarczenie informacji związanych z zasadami odżywiania oraz błędami podczas jedzenia

FitBot posiada bogatą wiedzę dotyczącą zasad zdrowego odżywiania oraz błędów żywieniowych i bez problemu odpowie na większość z nich. Użytkownik może zadawać wiele dodatkowych pytań związanych z konkretnymi zasadami czy błędami i również powinien być w stanie uzyskać odpowiedzi. Podczas definiowania dialogów związanych z tym tematem, korzystano ze źródeł zawierających zestaw najbardziej popularnych i najczęściej zadawanych pytań, dotyczących zdrowego odżywiania. Istnieje zatem duże prawdopodobieństwo, że użytkownicy będą chcieli otrzymać odpowiedzi spośród właśnie tych najczęściej podejmowanych i szukanych zagadnień (Rysunek 51, Rysunek 52).



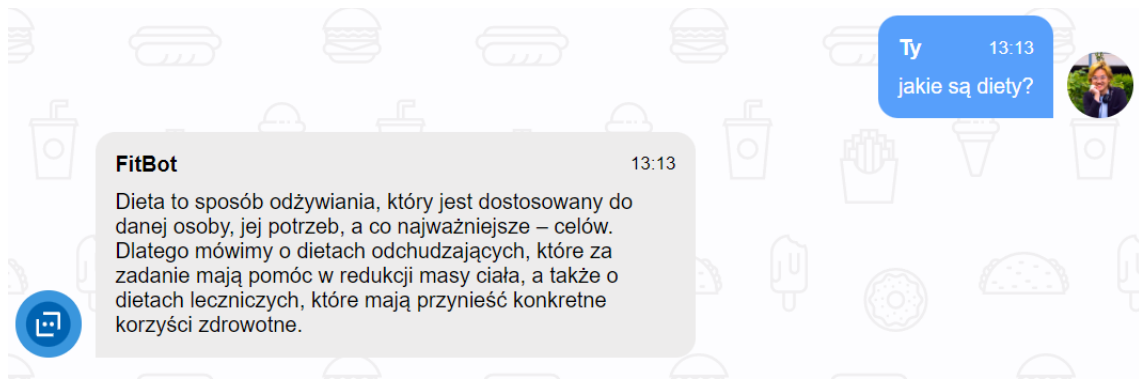
Rysunek 51 Odpowiedź FitBota dotycząca błędów podczas jedzenia (źródło: opracowanie własne)



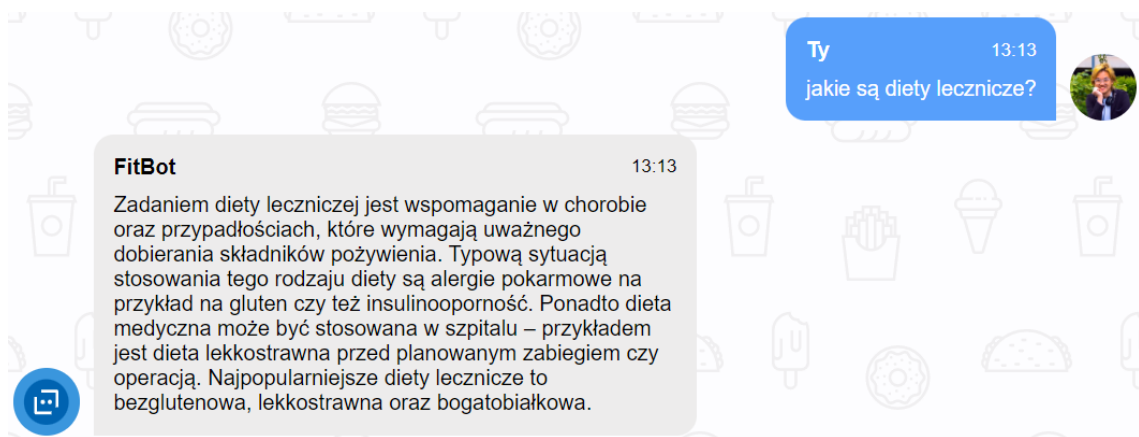
Rysunek 52 Odpowiedź FitBota na bardziej szczegółowe pytania związane z błędami podczas jedzenia (źródło: opracowanie własne)

- Rodzaje diet

Chatbot jest w stanie dostarczyć informacji na temat najbardziej popularnych i stosowanych obecnie diet (Rysunek 53, Rysunek 54).



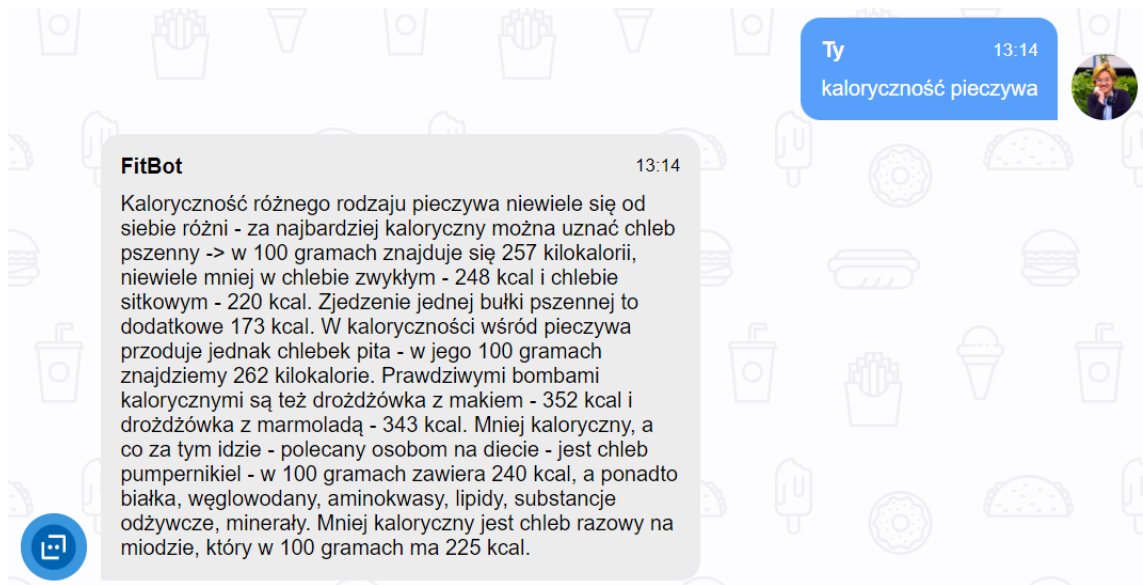
Rysunek 53 Podanie przez FitBota informacji dotyczących diet (źródło: opracowanie własne)



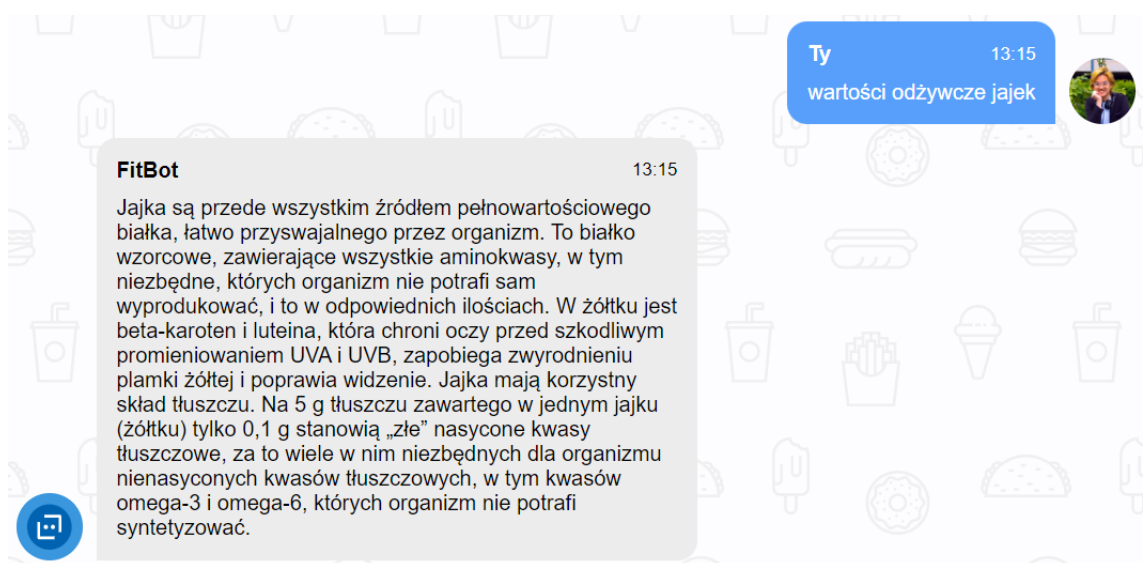
Rysunek 54 Dostarczenie przez bota informacji na temat diet leczniczych (źródło: opracowanie własne)

- Wiedza na temat wartości odżywczych produktów

FitBot posiada bogatą wiedzę dotyczącą wartości odżywczych poszczególnych produktów, szczególnie tych najbardziej znanych i wykorzystywanych, jak pieczywo, mięso, sery, kiełbasy, jajka, mleko, ryby, słodycze, owoce czy warzywa. Nie posiada jednak informacji na temat każdego możliwego wyrobu, ponieważ skonstruowanie tak dużej ilości możliwych dialogów do każdego istniejącego produktu byłoby niezwykle czasochłonne i raczej niewykonalne (Rysunek 55, Rysunek 56).



Rysunek 55 Odpowiedź FitBota dotycząca kaloryczności pieczywa (źródło: opracowanie własne)

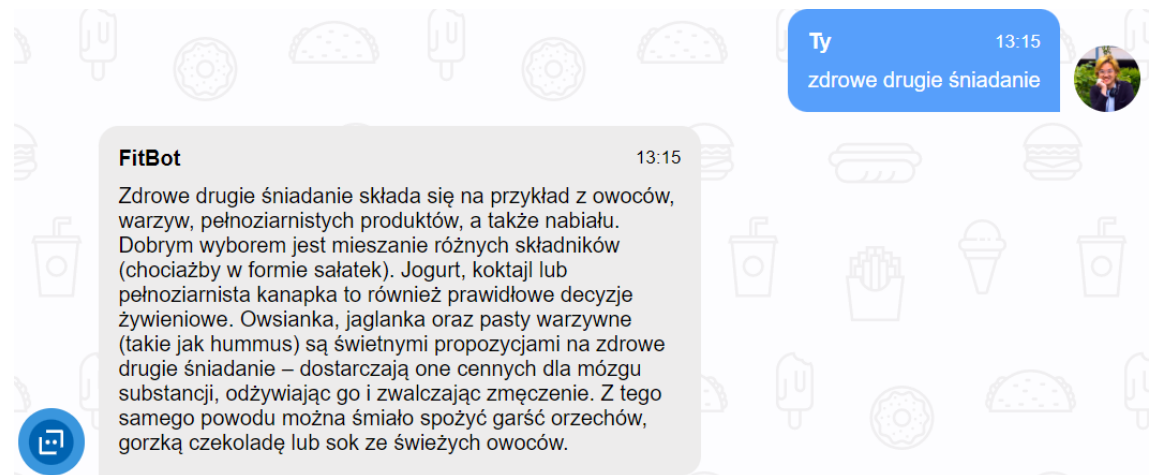


Rysunek 56 Podanie przez bota danych dotyczących wartości odżywczych jajek (źródło: opracowanie własne)

- Pomoc w stworzeniu zdrowego posiłku

FitBot jest w stanie zaproponować użytkownikowi, co mógłby zjeść na dany posiłek - na przykład na śniadanie czy kolację. Nie potrafi jednak zrobić tego tak efektywnie, jak było to w przypadku zastosowania frameworku Dialogflow, ponieważ

nie posiada umiejętności wydobywania od użytkownika jego preferencji i odpowiadania przez dopasowanie najlepszych możliwych opcji do nich. Bot potrafi jedynie podpowiedzieć, co mogłoby być dobrym pomysłem na pełnowartościowy posiłek, ale w sposób ogólny i nie bazując na indywidualnych wymaganiach użytkownika (Rysunek 57).



Rysunek 57 Pomoc FitBota w dobraniu posiłku na drugie śniadanie (źródło: opracowanie własne)

3.3. Testowanie i porównanie chatbotów

3.3.1. Wady i zalety zastosowanych frameworków

W poniższych tabelach scharakteryzowano główne wady i zalety każdego z zastosowanych frameworków oraz porównano je pod względem odpowiednich funkcjonalności (Tabela 1, Tabela 2, Tabela 3, Tabela 4).

Dialogflow – zalety
<p>Bardzo prosta obsługa i przyjazny interfejs: nie jest wymagana znajomość żadnego języka programowania, posiadanie wiedzy informatycznej/związanej ze sztuczną inteligencją, czy też wcześniejsze doświadczenie w tworzeniu chatbotów. Jest to zatem idealne rozwiązanie dla osób rozpoczynających swoją przygodę w świecie tworzenia wirtualnych asystentów.</p>

<p>Łatwość w budowaniu i wdrażaniu bota: trenowanie modelu jest automatyczne - Dialogflow posiada wbudowane funkcje przetwarzania języka naturalnego (NLP), dzięki którym jest w stanie rozumieć dane wprowadzane przez użytkowników i na nie reagować, poprzez odpowiednie zarządzanie kontekstem rozmowy.</p>
<p>Posiadanie wstępnie zdefiniowanych dialogów: na początku pracy z Dialogflow można tam znaleźć gotowe dialogi: powitalne, pożegnalne, dotyczące popularnych krótkich rozmów. Nie ma potrzeby ich szkolenia, są to wbudowane funkcjonalności. Osoba tworząca bota może je zmodyfikować, jeśli tego potrzebuje.</p>
<p>Możliwość tworzenia interfejsu konwersacyjnego dla wielu platform: za pomocą Dialogflow można zintegrować chatbota z wieloma aplikacjami, takimi jak np. Google Assistant, Cortana, Facebook Messenger.</p>
<p>Wielojęzyczność: Dialogflow obsługuje ponad 20 języków, w tym język polski.</p>
<p>Dialogflow - wady</p>
<p>Brak elastyczności podczas tworzenia scenariuszy rozmów: w przypadku, kiedy programista zdecyduje, że chciałby umieścić odpowiedź chatbota pod inną intencją niż obecnie, nie jest w stanie zwyczajnie jej skopiować lub przeciągnąć w inne miejsce. Musi usunąć wcześniejszą intencję i na nowo stworzyć ją w zmienionej lokalizacji. To powoduje, że programista musi nieraz poświęcić bardzo dużo czasu, aby odpowiednio przypisać intencje i zmusza go to do ciągłego myślenia w przód o prawidłowym przebiegu dialogu, aby potem nie musiał wszystkiego zmieniać i przepisywać na nowo.</p>
<p>Dużo niedającej się zautomatyzować pracy poświęconej na wytrenowanie chatbota: aby możliwe było odpowiednie nauczenie bota jak największej ilości scenariuszy odpowiedzi, konieczne jest wprowadzenie bardzo dużej liczby dialogów, w tym intencji oraz encji. Zajmuje to wiele czasu i wymaga ogromnej cierpliwości.</p>
<p>Brak dostępu do zaawansowanych opcji w darmowej wersji: nie ma możliwości korzystania z opcji Fulfillment, która pozwala na definiowanie swoich własnych akcji przy użyciu języka programowania.</p>

Ograniczona skalowalność i zależność od usług chmurowych: Dialogflow nie jest przeznaczony do użycia dla dużej liczby użytkowników, co może ograniczyć jego przydatność dla aplikacji posiadających dużą bazę konsumentów. Dostępność i efektywność chatbota są zależne od infrastruktury Google. Jeśli pojawią się jakieś problemy na platformie chmurowej, może to wpłynąć na jego funkcjonalność.

Tabela 1 Zalety i wady narzędzia Dialogflow

Rasa - zalety
<p>Rozumienie języka naturalnego (NLU): framework umożliwia wyodrębnianie intencji oraz parametrów z danych wejściowych, wpisywanych przez użytkowników. W swoim działaniu Rasa wspiera podejścia oparte na uczeniu maszynowym: nadzorowane uczenie maszynowe oraz systemy oparte na regułach, dzięki czemu umożliwia precyzyjne rozumienie zapytań użytkowników.</p>
<p>Łatwe dostosowanie i elastyczność: programista ma dostęp do całego potoku przetwarzania, dzięki czemu może dowolnie definiować przebieg własnych, niestandardowych akcji. Pozwala to dostosować zachowanie chatbota do konkretnych przypadków użycia i odpowiednich wymagań.</p>
<p>Dostęp do bogatej dokumentacji i posiadanie dużej bazy społeczności: Rasa dysponuje żywą i wspierającą się społecznością deweloperów. Współtworzą oni framework poprzez dostarczanie wielu cennych wskazówek oraz dzielenie się swoimi zasobami. Rasa posiada także szczegółową dokumentację oraz wiele przykładowych projektów, które są bardzo pomocne dla osób rozpoczynających pracę z chatbotami.</p>
<p>Możliwość korzystania z narzędzia w trybie offline: do tworzenia chatbota nie jest konieczne połączenie z Internetem, a to oznacza, że Rasa może być wykorzystywana w systemach offline, na przykład w urządzeniach IoT.</p>

<p>Oprogramowanie on-premise: w przeciwieństwie do rozwiązań w chmurze, Rasa jest wykorzystywana lokalnie. Dzięki temu można mieć pełną kontrolę nad danymi generowanymi przez chatbota, ale trzeba też zapewnić do tego wydajny serwer.</p>
<p>Rasa - wady</p>
<p>Trudny proces tworzenia bota: fazy instalacji, konfiguracji i trenowania bota wymagają wiedzy z zakresu uczenia maszynowego (przynajmniej na poziomie podstawowym).</p>
<p>Duże wymagania związane z trenowaniem chatbotów: efektywność Rasy zależy przede wszystkim od dostępności oraz jakości danych szkoleniowych. Proces uczenia bota może zajmować wiele czasu i wymaga posiadania specjalistycznej wiedzy. Dopasowanie odpowiednich danych do wytrenowania chatbota jest istotnym czynnikiem do tego, aby był on w stanie precyzyjnie rozpoznawać intencje oraz wydobywać z nich odpowiednie encje.</p>
<p>Brak automatycznej opcji rozpoznawania kontekstu: bot nie posiada wbudowanej opcji do rozpoznania i utrzymania kontekstu, wymagane jest samodzielne wytrenowanie go za pomocą odpowiednich dialogów i akcji.</p>
<p>Brak wbudowanej integracji z platformami komunikacyjnymi: Rasa posiada możliwość integracji z różnymi usługami, ale nie jest to proste i wymaga od programisty dodatkowej pracy.</p>

Tabela 2 Zalety i wady frameworku Rasa

<p>ChatterBot - zalety</p>
<p>Łatwość użycia: pomimo tego, że wymaga posiadania podstawowej wiedzy programistycznej, jest stosunkowo łatwy do zrozumienia i użycia, co czyni go odpowiednim dla osób rozpoczynających swoją przygodę związaną z budowaniem chatbotów w Pythonie.</p>
<p>Posiadanie wstępnie wytrenowanych modeli: ChatterBot dostarcza wstępnie wytrenowane modele językowe, np. dla języka angielskiego, które mogą być dobrym punktem wyjścia do</p>

<p>rozwoju chatbota. Modele te zostały już wytrenowane na dużym korpusie danych konwersacyjnych, dzięki czemu zmniejsza to ilość wymaganych danych szkoleniowych.</p>
<p>Bezproblemowe dostosowywanie: zapewnia łatwe trenowanie chatbota zgodnie z wymaganiami programisty. Możliwe jest definiowanie własnych danych treningowych tak, aby bot był w stanie rozumieć i odpowiadać na pytania z wielu dziedzin.</p>
<p>Wsparcie dla wielu języków: ChatterBot obsługuje wiele języków i umożliwia tworzenie botów w językach innych niż angielski. Jest to bardzo pomocne przy budowaniu chatbotów, które zaspokajają różnorodną bazę użytkowników.</p>
<p>Łatwa integracja z innymi platformami: narzędzie pozwala na bezproblemową integrację z różnymi frameworkami Pythona, takimi jak na przykład Flask i Django, co pozwala na wygodne wdrażanie aplikacji chatbotowych w sieci.</p>
<p>ChatterBot - wady</p>
<p>Ograniczona złożoność: jest przeznaczony do przeprowadzania łatwych konwersacji i może mieć problem z bardziej rozbudowanymi dialogami. Podstawowe algorytmy, używane domyślnie w ChatterBocie, takie jak TF-IDF nie potrafią radzić sobie dobrze z zaawansowanymi pytaniami lub kontekstem tak, jak inne, bardziej złożone modele NLP.</p>
<p>Brak świadomości kontekstu: narzędzie nie posiada automatycznej zdolności wychwytywania kontekstu z rozmowy. Powoduje to, że często nie jest w stanie utrzymać spójnego dialogu i udzielać odpowiedzi, adekwatnych do zadanego pytania.</p>
<p>Czasochłonność: osiągnięcie optymalnej wydajności wymaga wytrenowania bota przy użyciu ogromnej ilości scenariuszy rozmów. Zebranie i przygotowanie odpowiedniej ilości danych może zająć bardzo dużo czasu i wymagać dużych zasobów.</p>

Trudność w utrzymaniu i aktualizacji: ChatterBot to biblioteka open-source, która bazuje na wkładach społeczności. Wiąże się to z faktem, iż utrzymanie oraz aktualizacja biblioteki bezpośrednio zależą od aktywności i dostępności użytkowników korzystających z niej.

Tabela 3 Zalety i wady narzędzia ChatterBot

Rasa	Dialogflow	ChatterBot
Wymaga instalacji wielu komponentów.	Brak potrzeby instalacji narzędzi, można zacząć pracę z botem natychmiast po utworzeniu konta na stronie.	Wymaga instalacji wielu komponentów.
Wymaga specjalistycznej wiedzy.	Łatwy w użyciu, może być wykorzystywany przez osoby nieposiadające żadnej wiedzy technicznej.	Wymaga specjalistycznej wiedzy.
Narzędzie typu open-source, kod dostępny na GitHubie.	Zamknięty system.	Narzędzie typu open-source, kod dostępny na GitHubie.
Brak interfejsu, zapisywanie plików w formacie JSON.	Posiada webowy interfejs do tworzenia botów.	Brak interfejsu, zapisywanie plików w formacie YAML.
Brak hostingu (w darmowej wersji), hosting na własnym serwerze.	Dane są przechowywane w chmurze, nie może być hostowany na własnym serwerze.	Brak hostingu.
Brak wbudowanej integracji z platformami komunikacyjnymi, wymaga podjęcia dodatkowych działań przez programistę w celu ich powiązania.	Posiada możliwość natychmiastowej integracji z wieloma platformami, takimi jak np. Google Assistant, Slack czy Messenger.	Brak wbudowanej integracji z platformami komunikacyjnymi.

Tabela 4 Porównanie funkcjonalności wszystkich trzech frameworków – Rasy, Dialogflow i ChatterBota

3.3.2. Ocena satysfakcji użytkowników

Ocena satysfakcji użytkowników jest jednym z najważniejszych aspektów, które należy wziąć pod uwagę, aby zweryfikować efektywność stworzonych chatbotów. W tym celu na grupie 10 osób przeprowadzono badanie, polegające na interakcji z FitBotem. Przed rozpoczęciem pracy z wirtualnym asystentem, uczestnikom zadano pytania dotyczące ich ogólnej wiedzy oraz posiadanego doświadczenia w przeprowadzaniu rozmów z tego typu maszynami. Następnie, każda z osób podjęła konwersacje z FitBotem, używając wszystkich trzech frameworków (Dialogflow, Rasa oraz ChatterBot), po czym odpowiedziała na zestaw pytań dotyczących doświadczeń i satysfakcji z odbytych rozmów. Formularze z pytaniami zostały stworzone za pomocą narzędzia Google Forms i dostarczone uczestnikom w formie linków.

Opracowano 5 scenariuszów rozmów, obejmujących pięć podstawowych tematów związanych z dietą i zdrowym odżywianiem, w jakich specjalizuje się FitBot.

Scenariusz 1

Uczestnik badania zobowiązany jest do przeprowadzenia z botem rozmowy na temat diety, która będzie dla niego najbardziej odpowiednia. Powinien dowiedzieć się, czy sposób, w jaki obecnie się odżywia jest prawidłowy, czy też powinien wprowadzić zmiany i spróbować zmienić swoją obecną dietę.

Scenariusz 2

Uczestnik powinien zadać FitBotowi jak najwięcej pytań dotyczących zasad zdrowego odżywiania, błędów popełnianych podczas jedzenia i innych, związanych z prowadzeniem zdrowego stylu życia.

Scenariusz 3

Uczestnik zobowiązany jest do pozyskania informacji związanych z rodzajami diet, ich celów, wykorzystywanych produktów, zastosowań.

Scenariusz 4

Osoba biorąca udział w badaniu powinna otrzymać dane dotyczące wartości odżywczych dla konkretnego wybranego produktu.

Scenariusz 5

Wymagane jest od uczestnika uzyskanie od chatbota pomocy przy wybraniu/dopasowaniu zdrowego posiłku na daną porę (np. na śniadanie, obiad).

Wyniki ankiety

Poniżej znajdują się wyniki ankiety, dotyczące doświadczenia użytkowników w przeprowadzaniu rozmów z chatbotami.

Czy zdarzyło Ci się już przeprowadzać rozmowy z chatbotami przed tym eksperymentem? Jeśli tak, to napisz proszę, jakie to były chatboty. Jeśli nie, to pozostaw puste pole.

9 odpowiedzi

- nie
- Tak, używałam chatbotów na przykład za pośrednictwem Facebooka - tam niektóre strony mają swoje boty, które odpowiadają automatycznie, na przykład te z usługami kosmetycznymi. Oprócz tego korzystałam z pomocy wirtualnego asystenta, na przykład na stronach zakupowych- np. sklep H&M posiada takiego chatbota.
- tak, np. ChatGPT
- Tak, ChatGPT, chatboty messengerowe dla różnych stron np. AVON
- tak, z chatem GPT
- tak, chatGPT
- tak, np. chatgpt albo booking.com
- tak, chatGPT, różne chaty ze stron na Facebooku - np. Peugeot albo Komputronik

Rysunek 58 Wyniki ankiety dotyczące ich doświadczenia w przeprowadzaniu rozmów (źródło: opracowanie własne)

Jak widać na powyższym rysunku (Rysunek 58), zdecydowana większość osób biorących udział w badaniu korzystała już wcześniej z jakiegoś chatbota. Pojawiło się wiele odpowiedzi o chacie GPT, czyli jednym z najbardziej popularnych obecnie botów, który jest w stanie dostarczyć informacji z wielu różnych dziedzin życia. Uczestnicy przeprowadzali też konwersacje z wirtualnymi asystentami za pośrednictwem różnych stron na Facebooku, np. związanych z usługami kosmetycznymi czy też samochodowymi. Fakt korzystania przez respondentów z chatbotów może posłużyć jako

wskaźnik tego, że z dużym prawdopodobieństwem chcieliby też używać w przyszłości innych, np. właśnie FitBota.

Poniżej przedstawiono, jak większość uczestników oceniła chatbota w poszczególnych frameworkach (Tabela 5).

Oceniana funkcjonalność	Dialogflow	Rasa	ChatterBot
Łatwość przeprowadzenia rozmowy	10	8	6
Szybkość odpowiadania na pytania	10	10	10
Dokładność i precyzyjność odpowiedzi	9	8	7
Atrakcyjność wizualna	10	8	9
Satysfakcja z rozmowy	9	8	7

Tabela 5 Porównanie niektórych funkcjonalności pomiędzy trzema frameworkami

Analizując powyższe wyniki można zauważyć, że najlepiej został oceniony Dialogflow. Przyznano mu najwięcej punktów w każdej z kategorii. Jeśli chodzi o szybkość odpowiadania na pytania, wszystkie z chatbotów otrzymały najwyższą notę. Najsłabsze wyniki otrzymał ChatterBot, szczególnie w kwestiach związanych z łatwością przeprowadzania rozmowy, jak i dokładności odpowiedzi. Jeśli chodzi o wygląd wizualny, w każdym z frameworków ocena była bardzo wysoka.

W poniższej tabeli (Tabela 6) zebrano wyniki dotyczące odpowiedzi na pozostałe z pytań.

Sprawdzana funkcjonalność	Dialogflow	Rasa	Chatterbot
Skutecznie przeprowadzone scenariusze rozmów	8/10	6/10	8/10
Rozumienie pytań	5/10	6/10	2/10

Tabela 6 Porównanie wybranych funkcjonalności między trzema narzędziami

Obserwując powyższe wyniki można zauważyć, że we wszystkich frameworkach skuteczność przeprowadzenia wszystkich scenariuszy rozmów jest wysoka. Najlepiej udało się zrealizowanie każdego ze schematów za pomocą Dialogflow oraz ChatterBota. Pomimo że Rasa ma trochę niższą skuteczność od pozostałych frameworków, to jednak bardzo dobrze radzi sobie z rozumieniem pytań. ChatterBot jest zdecydowanie najsłabszy w tej kwestii, o czym świadczy fakt, że aż osiem spośród wszystkich dziesięciu badanych osób zaznaczyło, że nie poradził sobie z odpowiadaniem na ich wypowiedzi.

Poniżej znajdują się stwierdzenia uczestników dotyczące tego, czy udało im się przeprowadzić wszystkie scenariusze rozmów i jeśli nie, to jakie były komplikacje (Rysunek 59, Rysunek 60, Rysunek 61).

Dialogflow:

nie

nie, nie doradził dobrze, jak szukałem propozycji na obiad - nie zrozumiał produktu, który chciałem wykorzystać

Rysunek 59 Odpowiedź użytkowników na pytanie, czy udało się przeprowadzić wszystkie scenariusze rozmów w Dialogflow (źródło: opracowanie własne)

Rasa:

Nie udało się przy komponowaniu posiłku - bot nie potrafił zaproponować posiłku, bo podałem produkty, których nie potrafił rozpoznać i miał z tym kłopot

nie, bo na przykład przy zadawaniu niektórych pytań związanych z zasadami zdrowego odżywiania nie potrafił odpowiedzieć i dawał odpowiedzi niezwiązane z tematem

nie, nie rozpoznał produktu, którego wartość odżywczą chciałem sprawdzić

nie

Rysunek 60 Odpowiedź użytkowników na pytanie, czy udało się przeprowadzić wszystkie scenariusze rozmów w Rasie (źródło: opracowanie własne)

ChatterBot:

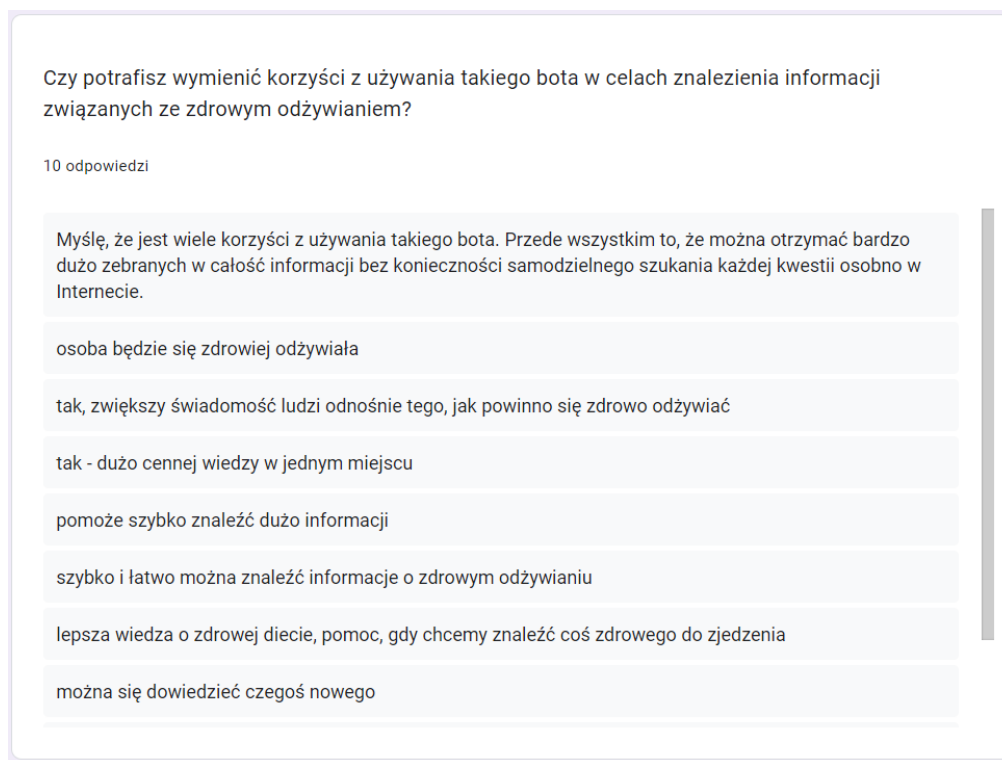
nie, był problem z dopasowaniem diety dla mnie - nie zrozumiał mojej odpowiedzi i nie zrozumiał kilku pytań związanych z zasadami zdrowego odżywiania

nie, nie potrafił np. odpowiadać na niektóre moje pytania odnośnie błędów w odżywianiu

Rysunek 61 Odpowiedź użytkowników na pytanie, czy udało się przeprowadzić wszystkie scenariusze rozmów w ChatterBocie (źródło: opracowanie własne)

Zarówno w przypadku Dialogflow, jak i Rasy bot miał problem z zaproponowaniem posiłku, co wynikało z braku odpowiedniej wiedzy na temat produktów, które użytkownik chciał wykorzystać. Jeśli chodzi o ChatterBota, problem polegał głównie na braku zrozumienia przez bota, czego użytkownik chce się dowiedzieć, nie rozumiał wypowiedzi, którą otrzymał.

Poniżej znajdują się wybrane odpowiedzi uczestników po przeprowadzeniu interakcji z FitBotem. Okazuje się, że chatbot posiada wiele zalet, które zostały zauważone i docenione przez osoby biorące udział w badaniu. Poniżej wymieniono kilka z nich (Rysunek 62).



Rysunek 62 Odpowiedzi użytkowników dotyczące tego, jakie korzyści może przynieść korzystanie z FitBota w celu znalezienia informacji na temat zdrowego odżywiania (źródło: opracowanie własne)

- **Oszczędność czasu i wiarygodne źródło informacji**

Na powyższym rysunku przedstawiono fragment odpowiedzi, które są praktycznie identyczne dla wszystkich trzech frameworków. Można zauważyć, że na pytanie dotyczące tego, jakie korzyści miałyby korzystanie z bota w celach znalezienia informacji związanych ze zdrowym odżywianiem, kilkakrotnie pojawiła się odpowiedź związana z tym, że pomogłoby to szybko i łatwo znaleźć konkretne informacje, zebrane w jednym miejscu. Zdecydowanie jest to istotna kwestia, biorąc pod uwagę to, jak ważna jest dla każdego oszczędność czasu i możliwość jak najszybszego otrzymania odpowiedzi na nurtujące kwestie. Kilku spośród uczestników odpowiedziało, że korzystanie z chatbota wpłynęłoby na poszerzenie wiedzy związanej ze zdrowym odżywianiem. Na pewno miałyby to jak najbardziej pozytywny efekt i chatbot stanowiłby rzetelne źródło informacji, które pomogłoby w prowadzeniu zdrowego trybu życia. Oszczędność czasu oraz bycie wiarygodnym źródłem informacji zostało ocenione pozytywnie zarówno dla Dialogflow, Rasy, jak i ChatterBota.

Dialogflow:

- **Sugerowane przyciski/tematy rozmów**

Co najbardziej podobało Ci się podczas prowadzenia rozmowy z chatbotem?

10 odpowiedzi

To, że chatbot sugerował możliwe odpowiedzi, aby pokierować rozmowę w odpowiednim kierunku i to, że jego odpowiedzi nie były sztywne - zawierały emotikony i zdjęcia.

Ze szybko odpowiadał.

Sugerowanie odpowiedzi, tematów rozmów, ładne wizualnie odpowiedzi

odpowiedzi wzbogacone obrazkami, podpowiedzi

używanie przez niego emotikon, życzliwe odpowiedzi

sugerowane odpowiedzi, rozkręcanie rozmowy, zadawanie przez chatbota dużo pytań o moje preferencje

ładny styl wypowiedzi, przyciski z propozycjami, prowadzenie mnie w konwersacji

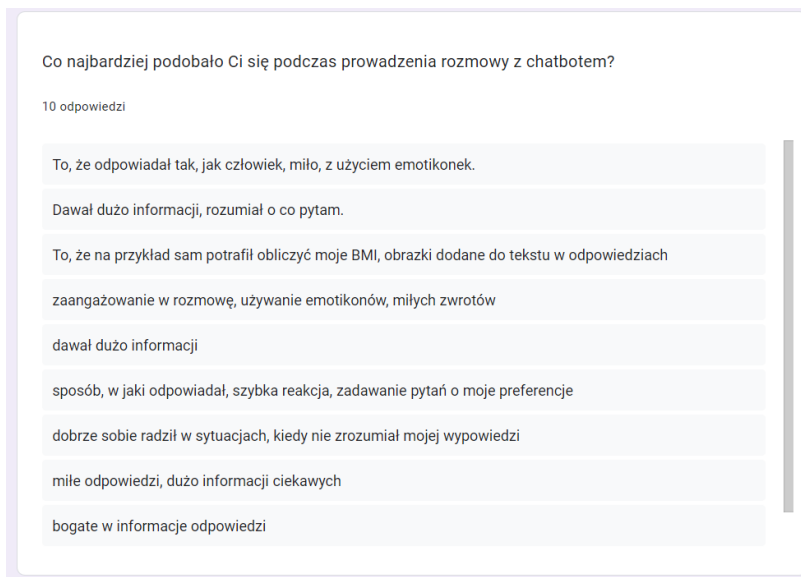
wszystko, to, że prowadził przebiegiem rozmowy

niwna rozmowa

Rysunek 63 Odpowiedzi użytkowników na pytanie, co najbardziej podobało im się podczas konwersacji z chatbotem w Dialogflow (źródło: opracowanie własne)

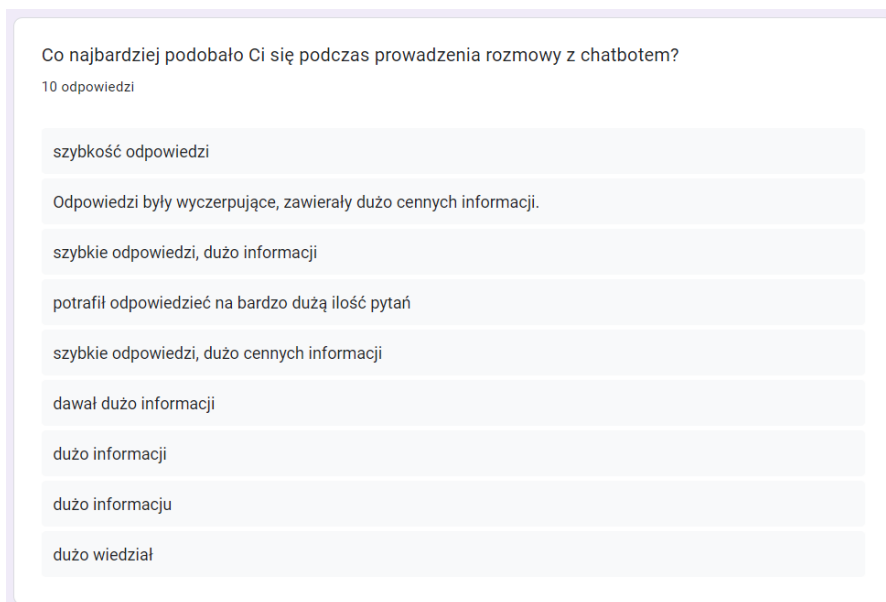
Zdecydowanej większości uczestników podczas korzystania z FitBota w Dialogflow najbardziej podobało się to, że sugerował on możliwe odpowiedzi do wyboru, posiadał przyciski i kierował rozmową. Na pewno ułatwiło to przebieg konwersacji, a dodatkowo wpłynęło na lepszy efekt wizualny (Rysunek 63).

Rasa:



Rysunek 64 Odpowiedzi użytkowników na pytanie, co najbardziej podobało im się podczas konwersacji z chatbotem w Rasie (źródło: opracowanie własne)

ChatterBot:



Rysunek 65 Odpowiedzi użytkowników na pytanie, co najbardziej podobało im się podczas konwersacji z chatbotem w ChatterBocie (źródło: opracowanie własne)

- **Bogate źródło informacji**

W przypadku Rasy oraz ChatterBota tym, na co najbardziej zwrócili uwagę uczestnicy była duża ilość informacji zawartych w odpowiedziach. Dodatkowo dla Rasy pozytywnie ocenili oni wzbogacenie tekstu o obrazki, czy też emotikony (Rysunek 64. Rysunek 65).

Oprócz wielu pozytywnych aspektów, jakie zostały docenione przez użytkowników, zauważono też różne wady w działaniu FitBota (Rysunek 66, Rysunek 67, Rysunek 68).

Dialogflow:

Czy potrafisz wymienić wady tego chatbota?

9 odpowiedzi

- nie wszystko rozumie
- Wadą jest to, że nie posiada wiedzy na temat wszystkich możliwych produktów i potrafi zaproponować do zjedzenia tylko zestaw wgranych mu tych samych posiłków.
- nie, wszystko mi się podobało
- czasami gubi wątek i odpowiada coś całkiem niezgodnego z pytaniem
- nie zawsze wszystko rozumie
- nie wszystko jest w stanie zrozumieć, jego wiedza jest ograniczona
- nie potrafi doradzić w każdym przypadku, trzeba trafić na produkty, które zna
- nie

Rysunek 66 Wymienione przez użytkowników wady narzędzia Dialogflow (źródło: opracowanie własne)

Główną wadą chatbota w Dialogflow było to, że nie był w stanie zrozumieć każdej wypowiedzi i posiadał ograniczoną wiedzę na temat niektórych produktów.

Rasa:

Czy potrafisz wymienić wady tego chatbota?

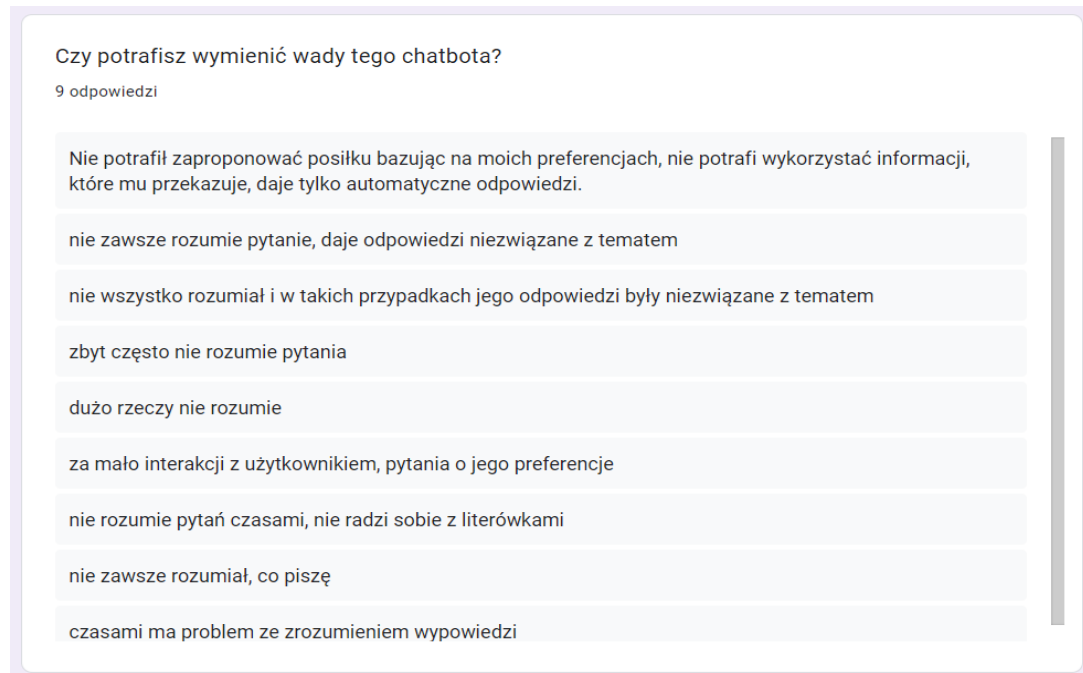
7 odpowiedzi

- nie
- Nie zawsze rozumiał o co pytałam.
- za mała wiedza, za długie odpowiedzi
- nie wszystko rozumie
- czasami odpowiedzi były za długie, za dużo tekstu, za mało zdjęć lub emotikonek
- nie zawsze doradzi, nie zna wszystkich produktów

Rysunek 67 Wymienione przez użytkowników wady narzędzia Rasa (źródło: opracowanie własne)

W przypadku Rasy również zwrócono uwagę na to, że nie zawsze odpowiada poprawnie, ale również nie spodobało się uczestnikom, że w odpowiedzi znajduje się zbyt duża ilość informacji.

ChatterBot:



Rysunek 68 Wymienione przez użytkowników wady narzędzia ChatterBot (źródło: opracowanie własne)

Głównym zarzutem dla ChatterBota była jego nieumiejętność radzenia sobie ze zrozumieniem wypowiedzi użytkowników, szczególnie podczas popełniania błędów w pisowni oraz generowanie odpowiedzi niezgodnych z tematem.

Dla wszystkich trzech frameworków po zadaniu uczestnikom pytania, czy chcieliby ponownie skorzystać z chatbota w przyszłości, otrzymano praktycznie stuprocentowo zapewnienie, że tak. Poniżej przedstawiono wybrane odpowiedzi (Rysunek 69).

Czy chciałbyś korzystać z tego chatbota w przyszłości? Uzasadnij swoją odpowiedź.

10 odpowiedzi

Tak, ze względu na to, że mógłbym lepiej się odżywiać.

Tak, bo lubię dowiadywać się informacji na temat zdrowego odżywiania

tak, można się dowiedzieć dużo rzeczy

tak, bo mi się podoba i można się od niego dużo dowiedzieć

tak, bo łatwo się z niego korzysta i często nie mam pomysłu, co można by było zrobić np. na śniadanie czy obiad

tak, jest pomocny

tak, bo dużo się dowiedziałam od niego

Rysunek 69 Odpowiedzi użytkowników dotyczące tego, czy chcieliby w przyszłości korzystać z FitBota (źródło: opracowanie własne)

Znalazło się kilka osób, które nie chciałyby skorzystać w przyszłości z któregoś z chatbotów, ale było to związane głównie z faktem, że dana osoba woli samodzielnie szukać informacji w Internecie lub zwyczajnie nie podoba jej się rozmawianie z chatbotami (Rysunek 70).

raczej nie, wolę szukać konkretnych informacji sama w sieci

raczej nie, bo nie jestem fanem korzystania z chatbotów

Rysunek 70 Odpowiedzi użytkowników, którzy nie chcieliby korzystać z FitBota (źródło: opracowanie własne)

Poniżej znajdują się odpowiedzi uczestników na pytanie, co według nich można by było usprawnić, aby chatbot był bardziej efektywny (Rysunek 71, Rysunek 72, Rysunek 73).

Dialogflow:

Zwiększenie bazy wiedzy na temat produktów i możliwych do zjedzenia posiłków, więcej pytań odnośnie indywidualnych preferencji użytkownika w przypadku tworzenia posiłku.

poszerzyć jego wiedzę, żeby proponować posiłki odpowiednio do potrzeb osoby

może dodać jeszcze więcej propozycji podczas tworzenia posiłku, może ograniczyć ilość niektórych informacji

mniej tekstu w odpowiedziach, bardziej rozbudowane pytania np. do wyszukiwania posiłku

możliwość rozmowy głosowej oprócz tekstowej

dodać więcej opcji, bardziej rozbudować

poszerzyć jego zakres wiedzy

dodać mu dodatkowe opcje

Rysunek 71 Propozycje użytkowników, jak sprawić, żeby chatbot w Dialogflow był bardziej efektywny (źródło: opracowanie własne)

W przypadku Dialogflow większość z nich zasugerowała rozszerzenie bazy wiedzy chatbota oraz redukcję informacji zawartych w odpowiedziach, wraz z dodaniem nowych opcji.

Rasa:

Rozszerzenie jego bazy wiedzy.

więcej obrazków w odpowiedziach, graficznych elementów, przycisków sugerujących odpowiedzi

przyciski sugerujące odpowiedzi, możliwe tematy dyskusji

krótsze odpowiedzi, bardziej rozbudowane opcje

przyciski, podawanie możliwych opcji wyboru

mniej tekstu, więcej emotikon i zwrotów do użytkownika

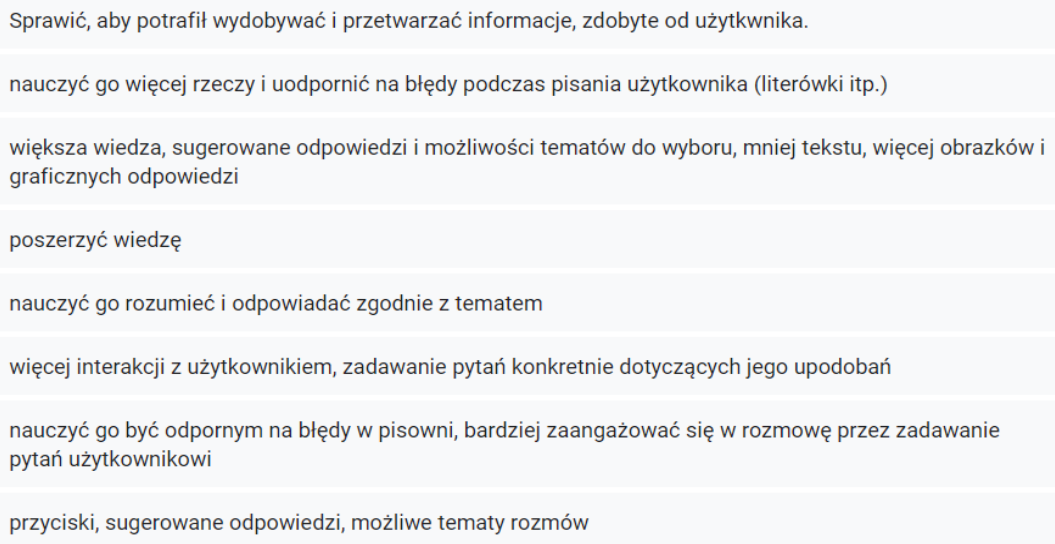
jeszcze więcej zadawać pytań użytkownikowi, poszerzyć jego możliwości doradzania, wzbogacić wiedzę o większą ilość produktów

więcej opcji

Rysunek 72 Propozycje użytkowników, jak sprawić, żeby chatbot w Rasie był bardziej efektywny (źródło: opracowanie własne)

Jeśli chodzi o Rasę, tutaj również zwrócono szczególny nacisk na rozszerzenie bazy wiedzy chatbota, jak również dodanie przycisków oraz sugestii dotyczących możliwych tematów dyskusji i odpowiedzi.

ChatterBot:



Sprawić, aby potrafił wydobywać i przetwarzać informacje, zdobyte od użytkownika.
nauczyć go więcej rzeczy i uodpornić na błędy podczas pisania użytkownika (literówki itp.)
większa wiedza, sugerowane odpowiedzi i możliwości tematów do wyboru, mniej tekstu, więcej obrazków i graficznych odpowiedzi
poszerzyć wiedzę
nauczyć go rozumieć i odpowiadać zgodnie z tematem
więcej interakcji z użytkownikiem, zadawanie pytań konkretnie dotyczących jego upodobań
nauczyć go być odpornym na błędy w pisowni, bardziej zaangażować się w rozmowę przez zadawanie pytań użytkownikowi
przyciski, sugerowane odpowiedzi, możliwe tematy rozmów

Rysunek 73 Propozycje użytkowników, jak sprawić, żeby chatbot w ChatterBocie był bardziej efektywny (źródło: opracowanie własne)

Po przeprowadzeniu konwersacji z ChatterBotem większość osób zaproponowała, że warto byłoby poszerzyć jego wiedzę, zwiększyć interakcję z użytkownikiem i uodpornić na popełniane przez niego w trakcie zadawania pytań błędy językowe.

Ostatnie pytanie w ankiecie dotyczyło wyboru przez użytkownika frameworka, w którym FitBot okazał się być najbardziej efektywny (Rysunek 74).

Po przeprowadzeniu rozmów z chatbotem, stworzonym w trzech różnych frameworkach (Dialogflow, Rasa, Chatterbot), który z nich jest Twoim zdaniem najbardziej efektywny i dlaczego?

10 odpowiedzi

Najbardziej efektywny jest dla mnie Dialogflow, ponieważ najłatwiej było przeprowadzać rozmowę, bardzo ładnie wyglądał wizualnie, najbardziej wygodnie się go używało przez połączenie z Messengerem, najbardziej poprawnie i dokładnie odpowiadał.

Najbardziej efektywna była dla mnie Rasa, bo było najmniej problemów ze zrozumieniem pytań

myślę, że Rasa, bo bot sam był w stanie wyznaczyć moje bmi, podobał mi się wizualnie, rozumiał wszystkie pytania

Dialogflow, bo najbardziej mi się podobał, rozumiał, co piszę

Myślę, że chyba Dialogflow, bo posiadał przyciski, sugestie odpowiedzi.

Myślę, że chyba Rasa, bo dawał wrażenie takiego najbardziej profesjonalnego i najbardziej mi się podobało pisanie z nim

Chyba Dialogflow, bo najłatwiej mi się rozmawiało

Myślę, że Rasa, bo odpowiadał precyzyjnie i wydawało się, że bardziej rozumie i przetwarza informacje, które mu podawałem

myślę, że Dialogflow, rozmowa najłatwiej przebiegała, tak lekko i najbardziej mi się podobało

chyba Rasa, bo najwięcej chatbot zrozumiał, rozmowa była płynna, podobał mi się też wizualnie

Rysunek 74 Odpowiedzi użytkowników, w którym frameworku chatbot był najbardziej efektywny (źródło: opracowanie własne)

Okazuje się, że połowa z uczestników zdecydowała, że najlepiej sprawdziła się Rasa, a druga połowa wybrała Dialogflow. Żadna z osób nie wskazała ChatterBota jako najbardziej efektywnego. Dialogflow i Rasa posiadają bardzo podobne funkcjonalności, zatem decyzja uczestników jest jak najbardziej zrozumiała. W obydwu z frameworków udało się z dużą skutecznością przeprowadzić zaplanowane scenariusze, odpowiedzi bota były dokładne i precyzyjne. Ponadto wypowiedzi FitBota zawierały dużo cennych informacji, wzbogaconych o obrazki oraz emotikony. Niewybranie przez uczestników ChatterBota jest najprawdopodobniej związane z faktem, iż często nie był on w stanie zrozumieć wypowiedzi oraz w najmniejszym stopniu angażował się w interakcję

z użytkownikiem, nie potrafił wydobywać przekazywanych mu od niego informacji, aby później je wykorzystać.

4. Podsumowanie i wnioski

Po przeanalizowaniu efektywności FitBota, stworzonego w trzech różnych frameworkach, zarówno pod kątem spełnienia przez niego założonych wymagań funkcjonalnych i niefunkcjonalnych, jak i po ocenie satysfakcji użytkowników, można stwierdzić, że wszystkie z wykorzystanych technologii okazały się skuteczne i pozwoliły na stworzenie pożytecznego bota. Wszystkie z narzędzi, zarówno Dialogflow, jak i Rasa oraz ChatterBot pozwoliły zbudować wirtualnego asystenta, który jest bardzo łatwy w obsłudze, potrafi udzielać szybkich odpowiedzi, oferuje pomoc o dowolnej porze – jest dostępny 24 h na dobę i możliwa jest jego integracja z różnymi platformami komunikacyjnymi. Ponadto, jest cennym źródłem informacji dotyczących zdrowej diety i prawidłowego odżywiania oraz ma dużą zdolność rozumienia i angażowania się w rozmowę z użytkownikiem. Po samodzielnym przetestowaniu możliwości i funkcjonalności oraz ocenie użytkowników można stwierdzić, że zdecydowanie najbardziej skutecznymi frameworkami do tworzenia chatbotów jest Dialogflow oraz Rasa. Obydwa narzędzia zapewniają zaawansowane możliwości rozumienia języka naturalnego, kontekstową obsługę dialogów oraz opcje dostosowywania konwersacji do konkretnych sytuacji i preferencji użytkowników.

Dialogflow posiada bardzo przyjazny i łatwy w obsłudze interfejs do tworzenia chatbota i nie wymaga posiadania specjalistycznej wiedzy, ale nie umożliwia w wersji darmowej definiowania swoich własnych, niestandardowych akcji. Rasa, w przeciwieństwie do Dialogflow, jest dużo bardziej wymagająca, ponieważ konieczne jest posiadanie podstawowych umiejętności programistycznych, ale dzięki temu możliwe jest tworzenie samodzielnych ścieżek i nauczenie bota, jak reagować na wiele różnych sposobów w zależności od sytuacji.

ChatterBot jest przydatnym narzędziem do tworzenia prostych chatbotów, które posiadają tylko podstawowe możliwości konwersacyjne. W przypadku bardziej złożonych aplikacji, dużo bardziej efektywne będą frameworki, takie jak Rasa i Dialogflow, które posiadają zdolność rozpoznawania kontekstu i dzięki temu są w stanie skutecznie prowadzić rozbudowane konwersacje z użytkownikami.

Literatura

- [1] H. Lane, H. Hapke, C. Howard, *Natural Language Processing in Action*, Manning Publications Co, Shelter Island 2019
- [2] https://pl.wikipedia.org/wiki/Jezyk_programowania [Data dostępu: 27.04.2023]
- [3] <https://www.ethnologue.com/> [Data dostępu: 27.04.2023]
- [4] W. A. Qader, M. M. Ameen, B. I. Ahmed, *An Overview of Bag of Words; Importance, Implementation, Applications, and Challenges*, IEEE, Iraq 2019
- [5] Q. Hang, Z. Chen, Z. Lu, Y. Ye, *Analysis of Bag-of-n-grams Representation's Properties Based on Textual Reconstruction*, [arXiv:1809.06502](https://arxiv.org/abs/1809.06502) [Data dostępu: 28.04.2023]
- [6] D. Jurafsky, J. Martin, *Speech and Language Processing*, Prentice Hall, Ohio, 2008
- [7] J. Wu, *Introduction to Convolutional Neural Networks*, National Key Lab for Novel Software Technology, Nanjing University, Chiny 2017
- [8] <https://www.cs.toronto.edu/~hinton/csc2535/notes/lec10new.pdf?ref=jeremyjordan.me> [Data dostępu: 04.05.2023]
- [9] S. Vajjala, B. Majumder, A. Gupta & H. Surana, *Practical Natural Language Processing*, O'Reily Media, Sebastopol 2020
- [10] D. Jurafsky, J. H. Martin, *Speech and Language Processing*, Financial Times Prentice Hall, Londyn 2008
- [11] <https://cloud.google.com/dialogflow/docs> [Data dostępu: 05.05.2023]
- [12] <https://www.k2bots.ai/abotc/adlo/dialogflow> [Data dostępu: 05.05.2023]
- [13] <https://rasa.com/docs/rasa/> [Data dostępu: 06.05.2023]
- [14] <https://towardsdatascience.com/building-a-chatbot-with-rasa-3f03ecc5b324> [Data dostępu: 06.05.2023]
- [15] <https://www.analyticsvidhya.com/blog/2021/11/an-introduction-to-chatbot-development-using-rasa/> [Data dostępu: 06.05.2023]
- [16] <https://chatterbot.readthedocs.io/en/stable/> [Data dostępu: 06.05.2023]
- [17] <https://www.techtarget.com/searchenterpriseai/definition/IBM-Watson-supercomputer> [Data dostępu: 06.05.2023]

- [18] M. Walker A., D. J. Litman, C. A. Kamm, A., *PARADISE: a framework for evaluating spoken dialogue agents*, Association for Computational Linguistics, Spain 1997
- [19] J. Carletta, *Assessing agreement on classification tasks: the kappa Statistic*, MIT Press, Cambridge, MA 1996
- [20] K. Kuligowska, *Commercial Chatbot: Performance Evaluation, Usability Metrics and Quality Standards of Embodied Conversational Agents*, Professionals Center for Business Research, 2015
- [21] M. Jadedda, N. Varia, *Perspectives for Evaluating Conversational AI*, 2017
- arXiv: <https://doi.org/10.48550/arXiv.1709.04734> [Data dostępu: 07.05.2023]
- [22] <https://pythonbasics.org/what-is-flask-python/> [Data dostępu: 07.05.2023]
- [23] <https://learn.microsoft.com/en-us/azure/bot-service/bot-service-overview?view=azure-bot-service-4.0> [Data dostępu: 12.05.2023]
- [24] https://pl.wikipedia.org/wiki/Asystent_Google [Data dostępu: 12.05.2023]
- [25] https://en.wikipedia.org/wiki/Amazon_Alexa [Data dostępu: 12.05.2023]

Załączniki

Załącznik 1 – Formularz

Pytania przed rozpoczęciem interakcji z chatbotem:

- Ile masz lat?
- Jaka jest Twoja płeć?
- Czy zdarzyło Ci się już przeprowadzać rozmowy z chatbotami przed tym eksperymentem?
- Czy możesz podać przykład pozytywnych doświadczeń interakcji z chatbotami?
- Czy masz jakieś negatywne doświadczenia związane z chatbotami?
- W skali od 1 do 10 jak oceniasz swoje dotychczasowe doświadczenie z korzystania z chatbotów?

Pytania po przeprowadzeniu rozmowy z chatbotem (takie same w przypadku każdego z frameworków):

Część 1

Pytania, w których użytkownik oceniał daną funkcjonalność chatbota w skali od 1 do 10:

- Jak łatwe było przeprowadzenie rozmowy z botem?
- Jak szybko chatbot odpowiadał na pytania?
- Jak dokładne i precyzyjne były odpowiedzi wirtualnego asystenta?
- Jak bardzo chatbot podobał Ci się wizualnie?
- Jak bardzo jesteś usatysfakcjonowany z przeprowadzonej rozmowy?
- Jak bardzo Twoim zdaniem ten chatbot byłby pomocny dla osób, które interesują się zdrowym odżywianiem i prowadzeniem zdrowego stylu życia?

Pozostałe pytania:

- Czy udało Ci się przeprowadzić wszystkie scenariusze rozmów? Jeśli nie, to powiedz, które się nie powiodły i napisz, jakie wystąpiły komplikacje.
- Czy chatbot miał problem ze zrozumieniem Twoich pytań?
- Co najbardziej podobało Ci się podczas prowadzenia rozmowy z chatbotem?

- Czy potrafisz wymienić korzyści z używania takiego bota w celach znalezienia informacji związanych ze zdrowym odżywianiem?
- Czy potrafisz wymienić wady tego chatbota?
- Co Twoim zdaniem można by było poprawić albo dodać, aby bot był bardziej efektywny?
- Czy chciałbyś korzystać z tego chatbota w przyszłości? Uzasadnij swoją odpowiedź.
- Po przeprowadzeniu rozmów z chatbotem, stworzonym w trzech różnych frameworkach (Dialogflow, Rasa, Chatterbot), który z nich jest Twoim zdaniem najbardziej efektywny i dlaczego?

Ankieta dotycząca FitBota

Hej! :) Poniższa ankieta ma na celu sprawdzenie satysfakcji z przeprowadzenia rozmowy z FitBotem, stworzonym w trzech różnych frameworkach (Dialogflow, Rasa, Chatterbot) i ich porównaniu.

Ankieta składa się z dwóch części.

I część to pytania

mające na celu sprawdzenie ogólnej wiedzy i Twojego doświadczenia w przeprowadzaniu rozmowy z chatbotami.

II część zawiera pytania dotyczące Twojego zadowolenia i doświadczeń po przeprowadzeniu konwersacji z FitBotem.

I część

Poniżej znajduje się kilka pytań ogólnych.

Jak masz na imię?

Tekst krótkiej odpowiedzi

Ile masz lat?

Tekst krótkiej odpowiedzi

Jaka jest Twoja płeć?

- Kobieta
- Mężczyzna

Czy zdarzyło Ci się już przeprowadzać rozmowy z chatbotami przed tym eksperymentem? Jeśli tak, to napisz proszę, jakie to były chatboty. Jeśli nie, to pozostaw puste pole.

Tekst długiej odpowiedzi

W skali od 1 do 10 jak oceniasz swoje dotychczasowe doświadczenie z korzystania z chatbotów?

- 1 2 3 4 5 6 7 8 9 10
- małe duże

II część

Poniżej znajdują się pytania dotyczące Twojego doświadczenia i satysfakcji z przeprowadzenia konwersacji z FitBotem, odpowiednio dla każdego z frameworków, w jakich został stworzony.

Jak łatwe było przeprowadzenie rozmowy z botem?

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Jak szybko chatbot odpowiadał na pytania?

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Jak dokładne i precyzyjne były odpowiedzi wirtualnego asystenta?

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Jak bardzo jesteś usatysfakcjonowany z przeprowadzonej rozmowy?

1	2	3	4	5	6	7	8	9	10
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Jak bardzo Twoim zdaniem ten chatbot byłby pomocny dla osób, które interesują się zdrowym odżywianiem i prowadzeniem zdrowego stylu życia?

1	2	3	4	5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Czy udało Ci się przeprowadzić wszystkie scenariusze rozmów? Jeśli nie, to powiedz, które się nie powiodły i napisz, jakie wystąpiły komplikacje.

Tekst długiej odpowiedzi

Czy chatbot miał problem ze zrozumieniem Twoich pytań?

Tak

Nie

Co najbardziej podobało Ci się podczas prowadzenia rozmowy z chatbotem?

Tekst długiej odpowiedzi

Czy potrafisz wymienić korzyści z używania takiego bota w celach znalezienia informacji związanych ze zdrowym odżywianiem?

Tekst długiej odpowiedzi

Czy potrafisz wymienić wady tego chatbota?

Tekst długiej odpowiedzi

Co Twoim zdaniem można by było poprawić albo dodać, aby bot był bardziej efektywny?

Tekst długiej odpowiedzi

Czy chciałbyś korzystać z tego chatbota w przyszłości? Uzasadnij swoją odpowiedź.

Tekst długiej odpowiedzi

Po przeprowadzeniu rozmów z chatbotem, stworzonym w trzech różnych frameworkach (Dialogflow, Rasa, Chatterbot), który z nich jest Twoim zdaniem najbardziej efektywny i dlaczego?

Tekst długiej odpowiedzi

Pełna wersja formularza jest dostępna pod linkiem

https://docs.google.com/forms/d/e/1FAIpQLSe8SV3UfbzVu5CDMReOU470qoODITskFyYg0i0FczZn34CIA/viewform?usp=pp_url.

Spis rysunków

Rysunek 1 Przykład zdania podzielonego na tokeny (źródło: opracowanie własne na podstawie [1], s. 34).....	10
Rysunek 2 Reprezentacja wektorowa (one-hot-vectors) dla zdania (źródło: opracowanie własne na podstawie [1], s. 36).....	10
Rysunek 3 Zapisanie zdania w postaci bag-of-word vector (źródło: opracowanie własne na podstawie [1], s. 40).....	11
Rysunek 4 Etap konwolucyjnej sieci neuronowej (źródło: opracowanie własne na podstawie [1], s. 226).....	17
Rysunek 5 Rekurencyjna sieć neuronowa (źródło: opracowanie własne na podstawie [1]) s. 250.....	19
Rysunek 6 Panel do zarządzania chatbotem za pośrednictwem narzędzia Dialogflow (źródło: opracowanie własne).....	25
Rysunek 7 Schemat architektury Rasy (źródło: opracowanie własne na podstawie [15])	26
Rysunek 8 Sposób przekazania osobnych, krótkich dialogów do trenowania chatbota (źródło: opracowanie własne na podstawie [16])	27
Rysunek 9 Sposób przekazania rozbudowanego dialogu do uczenia chatbota (źródło: opracowanie własne na podstawie [16]).....	28
Rysunek 10 Przekazanie chatbotowi korpusu językowego do wytrenowania (źródło: opracowanie własne na podstawie [16]).....	28
Rysunek 11 Ekran powitalny FitBota w narzędziu Dialogflow (źródło: opracowanie własne).....	35
Rysunek 12 Przykład radzenia sobie FitBota z rozpoznaniem danych od użytkownika, pomimo błędów występujących w pisowni (źródło: opracowanie własne)	36
Rysunek 13 Przykład odpowiedzi chatbota, która wzbogacona jest o emotikony i zdjęcia (źródło: opracowanie własne).....	37
Rysunek 14 Panel do tworzenia i obsługi chatbota w narzędziu Dialogflow (źródło: opracowanie własne).....	39
Rysunek 15 Przykładowe intencje stworzone do wytrenowania FitBota (źródło: opracowanie własne).....	40
Rysunek 16 Przykładowa konstrukcja intencji w Dialogflow (źródło: opracowanie własne).....	41

Rysunek 17 Przykładowe zdefiniowanie encji dla owoców (fruit) (źródło: opracowanie własne).....	42
Rysunek 18 Przykład konwersacji z FitBotem za pośrednictwem Messengera (źródło: opracowanie własne).....	43
Rysunek 19 Przykład odpowiedzi chatbota związanej z dietą dla użytkownika (źródło: opracowanie własne).....	44
Rysunek 20 Odpowiedź chatbota, jaka dieta będzie najlepsza dla użytkownika (źródło: opracowanie własne).....	44
Rysunek 21 Odpowiedź chatbota dotycząca informacji o diecie Dąbrowskiej (źródło: opracowanie własne).....	45
Rysunek 22 Odpowiedź chatbota związana ze szczegółowym pytaniem użytkownika o dietę Dąbrowskiej (źródło: opracowanie własne).....	45
Rysunek 23 Wskazówki udzielane przez bota, związane ze zdrowym odżywianiem (źródło: opracowanie własne).....	46
Rysunek 24 Odpowiedź bota na pytanie dotyczące jednej z zasad zdrowego odżywiania (źródło: opracowanie własne).....	46
Rysunek 25 Odpowiedź bota na szczegółowe pytanie dotyczące jednego z błędów w żywieniu (źródło: opracowanie własne)	47
Rysunek 26 Informacje udzielone przez bota odnośnie do błędów żywieniowych (źródło: opracowanie własne).....	47
Rysunek 27 Sugerowane przez bota możliwości produktów, dla jakich można sprawdzić wartości odżywcze (źródło: opracowanie własne)	47
Rysunek 28 Przekazane przez bota informacje dotyczące wartości odżywczych dla mięsa (źródło: opracowanie własne).....	48
Rysunek 29 Przekazane przez bota dane dotyczące wartości odżywczych dla mięsa – c.d. (źródło: opracowanie własne)	49
Rysunek 30 Odpowiedź bota związana z wartościami odżywczymi chleba razowego (źródło: opracowanie własne).....	49
Rysunek 31 Odpowiedź bota w przypadku, kiedy nie rozpozna produktu, o jaki pyta użytkownik (źródło: opracowanie własne).....	50
Rysunek 32 Informacje przekazane przez bota dotyczące diet leczniczych (źródło: opracowanie własne).....	51

Rysunek 33 Odpowiedź dotycząca diety leczniczej – ciąg dalszy (źródło: opracowanie własne).....	51
Rysunek 34 Konwersacja z botem dotycząca znalezienia użytkownikowi posiłku, którego potrzebuje (źródło: opracowanie własne).....	54
Rysunek 35 Struktura najważniejszych plików, potrzebnych do stworzenia bota za pomocą Rasy (źródło: opracowanie własne)	54
Rysunek 36 Przykładowy intent dotyczący tworzenia posiłku (źródło: opracowanie własne).....	55
Rysunek 37 Fragment pliku rules.yml, w którym definiowane są reguły działania chatbota (źródło: opracowanie własne)	55
Rysunek 38 Zdefiniowanie dialogu za pomocą intencji oraz akcji w pliku stories.yml (źródło: opracowanie własne).....	56
Rysunek 39 Przykład akcji, zdefiniowanej w pliku actions.py (źródło: opracowanie własne).....	57
Rysunek 40 Odpowiedź bota stworzonego za pomocą Rasy dotycząca diety dla użytkownika (źródło: opracowanie własne)	57
Rysunek 41 Odpowiedź bota na szczegółowe pytania dotyczące zdrowego odżywiania (źródło: opracowanie własne).....	58
Rysunek 42 Odpowiedź chatbota na pytanie związane z dietami (źródło: opracowanie własne).....	58
Rysunek 43 Odpowiedź FitBota dotycząca wartości odżywczych produktów (źródło: opracowanie własne).....	59
Rysunek 44 Wyświetlenie przez FitBota wartości odżywczych dotyczących sera w formie tabeli (źródło: opracowanie własne)	59
Rysunek 45 Fragment odpowiedzi FitBota dotycząca wartości odżywczych słodczy (źródło: opracowanie własne).....	60
Rysunek 46 Konwersacja użytkownika z FitBotem dotycząca znalezienia posiłku na śniadanie (źródło: opracowanie własne).....	60
Rysunek 47 Ciąg dalszy konwersacji dotyczącej znalezienia posiłku użytkownikowi (źródło: opracowanie własne).....	61
Rysunek 48 Struktura plików, potrzebnych do stworzenia chatbota za pomocą narzędzia ChatterBot (źródło: opracowanie własne)	62

Rysunek 49 Przykładowe dialogi, zdefiniowane dla kategorii przywitania (źródło: opracowanie własne).....	63
Rysunek 50 Struktura folderu diet_corpus, w którym zdefiniowane zostały dialogi dla poszczególnych kwestii dotyczących diety (źródło: opracowanie własne).....	63
Rysunek 51 Odpowiedź FitBota dotycząca błędów podczas jedzenia (źródło: opracowanie własne).....	65
Rysunek 52 Odpowiedź FitBota na bardziej szczegółowe pytania związane z błędami podczas jedzenia (źródło: opracowanie własne).....	65
Rysunek 53 Podanie przez FitBota informacji dotyczących diet (źródło: opracowanie własne).....	66
Rysunek 54 Dostarczenie przez bota informacji na temat diet leczniczych (źródło: opracowanie własne).....	66
Rysunek 55 Odpowiedź FitBota dotycząca kaloryczności pieczywa (źródło: opracowanie własne).....	67
Rysunek 56 Podanie przez bota danych dotyczących wartości odżywczych jajek (źródło: opracowanie własne).....	67
Rysunek 57 Pomoc FitBota w dobraniu posiłku na drugie śniadanie (źródło: opracowanie własne).....	68
Rysunek 58 Wyniki ankiety dotyczące ich doświadczenia w przeprowadzaniu rozmów (źródło: opracowanie własne).....	75
Rysunek 59 Odpowiedź użytkowników na pytanie, czy udało się przeprowadzić wszystkie scenariusze rozmów w Dialogflow (źródło: opracowanie własne)	77
Rysunek 60 Odpowiedź użytkowników na pytanie, czy udało się przeprowadzić wszystkie scenariusze rozmów w Rasie (źródło: opracowanie własne).....	77
Rysunek 61 Odpowiedź użytkowników na pytanie, czy udało się przeprowadzić wszystkie scenariusze rozmów w ChatterBocie (źródło: opracowanie własne).....	78
Rysunek 62 Odpowiedzi użytkowników dotyczące tego, jakie korzyści może przynieść korzystanie z FitBota w celu znalezienia informacji na temat zdrowego odżywiania (źródło: opracowanie własne).....	79
Rysunek 63 Odpowiedzi użytkowników na pytanie, co najbardziej podobało im się podczas konwersacji z chatbotem w Dialogflow (źródło: opracowanie własne)	80
Rysunek 64 Odpowiedzi użytkowników na pytanie, co najbardziej podobało im się podczas konwersacji z chatbotem w Rasie (źródło: opracowanie własne)	81

Rysunek 65 Odpowiedzi użytkowników na pytanie, co najbardziej podobało im się podczas konwersacji z chatbotem w ChatterBocie (źródło: opracowanie własne)	81
Rysunek 66 Wymienione przez użytkowników wady narzędzia Dialogflow (źródło: opracowanie własne).....	82
Rysunek 67 Wymienione przez użytkowników wady narzędzia Rasa (źródło: opracowanie własne).....	82
Rysunek 68 Wymienione przez użytkowników wady narzędzia ChatterBot (źródło: opracowanie własne).....	83
Rysunek 69 Odpowiedzi użytkowników dotyczące tego, czy chcieliby w przyszłości korzystać z FitBota (źródło: opracowanie własne).....	84
Rysunek 70 Odpowiedzi użytkowników, którzy nie chcieliby korzystać z FitBota (źródło: opracowanie własne).....	84
Rysunek 71 Propozycje użytkowników, jak sprawić, żeby chatbot w Dialogflow był bardziej efektywny (źródło: opracowanie własne)	85
Rysunek 72 Propozycje użytkowników, jak sprawić, żeby chatbot w Rasie był bardziej efektywny (źródło: opracowanie własne)	85
Rysunek 73 Propozycje użytkowników, jak sprawić, żeby chatbot w ChatterBocie był bardziej efektywny (źródło: opracowanie własne)	86
Rysunek 74 Odpowiedzi użytkowników, w którym frameworku chatbot był najbardziej efektywny (źródło: opracowanie własne).....	87

Spis tabel

Tabela 1 Zalety i wady narzędzia Dialogflow	70
Tabela 2 Zalety i wady frameworku Rasa	71
Tabela 3 Zalety i wady narzędzia ChatterBot.....	73
Tabela 4 Porównanie funkcjonalności wszystkich trzech frameworków – Rasy, Dialogflow i ChatterBota.....	73
Tabela 5 Porównanie niektórych funkcjonalności pomiędzy trzema frameworkami.....	76
Tabela 6 Porównanie wybranych funkcjonalności między trzema narzędziami	76