

KebaBot - Chatbot obsługujący restaurację

Projekt końcowy na przedmiot PJN

Mikołaj Michalczak

Mateusz Nosal

Jacek Oleksy

Jan Sporszil

Gr. 42 (2)

Diagflow

Jest to narzędzie firmy Google do tworzenia oraz integrowania konwersacyjnych aplikacji takich jak chatboty oraz voiceboty. Wykorzystuje przetwarzanie języka naturalnego wraz z uczeniem maszynowym oraz wsparciem sztucznej inteligencji. Wspiera głównie języki europejskie i azjatyckie oraz ich dialekty. Od 2019 roku dostępne jest wsparcie dla języka polskiego. Możemy powiązać naszą aplikację z wieloma platformami oraz komunikatorami np. Asystent Google, Facebook Messenger, Slack lub umieścić na naszej własnej stronie internetowej.

Narzędzie pozwala nam przetworzyć wiadomość użytkownika na wiele sposobów. Pozwala nam określić potrzeby, polecenia oraz prośby. Może zidentyfikować parametry, które nas interesują do wykonania konkretnego działania. Śledzi również kontekst rozmowy w ramach jej postępowania.



Zadania KebaBota

Nasz bot ma za zadanie zebrać od klienta niezbędne informacje do złożenia zamówienia w restauracji. Powinien przeprowadzić rozmowę tak, żeby nakierować użytkownika na uzupełnienie braków, jeśli takie występują. Na końcu powinien przedstawić propozycję zamówienia do jego potwierdzenia. W przypadku negatywnej odpowiedzi, proces zaczyna się od początku.

Realizacja zadania

Pierwszym krokiem jest zdefiniowanie intencji użytkownika i określeniu zdarzeń w ramach, których przewidujemy, że będzie się pojawiać.

Intents

- customer kebabshop 
-  ↳ customer kebabshop - fallback Contexts: customerkebabshop-followup
- ↳ customer kebabshop - yes
- ↳ customer kebabshop - no
- ↳ customer kebabshop - cancel

W naszym projekcie utworzyliśmy „customer kebabshop”. W ramach tego zawierają się wszystkie możliwe odpowiedzi, którymi bot będzie się posługiwać. Dodatkowo można rozbudować intencję o prostsze przypadki występujące w jej ramach. Tutaj widzimy, że

mamy przygotowane odpowiedzi na tak, nie, odmowy wykonania czynności oraz fallback, która pojawia się w momencie braku dopasowania do czegokolwiek.



W ramach tworzenia intencji możemy utworzyć kontekst i przekazać informacje, które mogą być przenoszone do innych typów potrzeb. Można również wybrać zdarzenie kiedy ma się uruchomić.

Contexts

Add input context

2 customerkebabshop-followup  Add output context

Events

 Welcome  Add event

W naszym przypadku wypełniliśmy Events, ponieważ obracamy się tylko w ramach tego tematu. Następnym krokiem jest wypełnienie fraz trenujących. Podajemy tam zdania, na które ma reagować bot. Lista ta nie musi być duża, ponieważ Diagflow jest w stanie na podstawie kilku przykładów wygenerować wiele podobnych przypadków.

Training phrases

Search training phrases 








When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an 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
” I want a large mixed mixed sauce lunch
” I would like a medium spicy chicken pita
” I need a small veggie box
” i want a large beef burger
” I want a chicken
” I want a mild big rollo
” I want spicy lunch
” I want a mild sauce
” I want a big rollo with spicy sauce
” I need the rollo by today noon

Tak wyglądają przykładowe zdania wykorzystane do nauki KebaBota. Zaznaczone kolorem części zdania są tak zwanymi Entities, czyli parametry wykorzystywane przez nas do konkretnych celów. Każdy kolor odpowiada innej grupie zmiennych. Bot jest w stanie wyciągnąć te informacje ze zdania i zachować je do późniejszego wykorzystania. Definiujemy je blok niżej.

Action and parameters



REQUIRED 	PARAMETER NAME 	ENTITY 	VALUE	IS LIST 	PROMPTS 
<input checked="" type="checkbox"/>	typeofitem	@typeofitem	Stypeofitem	<input checked="" type="checkbox"/>	Hello, what are...
<input checked="" type="checkbox"/>	size	@size	Ssize	<input type="checkbox"/>	What size are y...
<input checked="" type="checkbox"/>	date-time	@sys.date-time	Sdate-time	<input type="checkbox"/>	What time do yo...
<input checked="" type="checkbox"/>	sauce	@sauce	Ssauce	<input checked="" type="checkbox"/>	What sauce woul...
<input checked="" type="checkbox"/>	meat	@meat	Smeat	<input type="checkbox"/>	What meat would...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

[+ New parameter](#)

Tutaj wskazujemy na zmienne, które pojawiają się w ramach konwersacji z użytkownikiem. Mamy tutaj do dyspozycji szereg opcji:

- required – wskazujemy, że informacja jest wymagana do spełnienia prośby użytkownika
- parameter name – nazwa parametru
- entity – encja (do tego przejdziemy później); musi być wcześniej zdefiniowana
- value – referencja do parametru; można przypisać wartość z góry lub skorzystać z

predefiniowanych zmiennych

is list – wskazujemy czy wartości mogą być zawarte jako lista

prompts – przygotowana wiadomość dla użytkownika w przypadku braku danej informacji

Ostatnim polem jest odpowiedź końcowa. Pojawia się w momencie, gdy bot uzyskał wszystkie wymagane atrybuty. Zostaje ona zaprezentowana dla użytkownika w celu potwierdzenia zamówienia

Responses ⓘ

DEFAULT +

Text Response ⓘ

1	We've received your \$size.original \$meat.original \$sauce.original \$typeOfitem.original order! It will be delivered at \$date-time.original. Do you confirm?
2	Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation ⓘ

Jak widać można tutaj także używać wartości parametrów, które zdefiniowaliśmy punkt wcześniej.

Przy tworzeniu parametrów zostały wspomniane encje. Nie pojawiają się one znikąd. Musimy je utworzyć, aby później bot mógł z nich korzystać.

Entities

Custom System

Search entities

- @ meat
- @ sauce
- @ size
- @ typeOfitem

Realizujemy to w zakładce „Entities”. Na powyższym obrazku widzimy, że mamy ich kilka utworzonych. W każdym z nich definiujemy wartości jakie mogą się pojawiać w ramach pojedynczego parametru oraz ich synonimy.

typeOfItem

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

box	box
rollo	rollo, tortilla
lunch	lunch
pita	pita, pocket bread
burger	burger, hamburger
Click here to edit entry	

[+ Add a row](#)

Na przykładzie dań w restauracji, możemy dobrze zobaczyć schemat tworzenia parametru. Podajemy na początku wartość podstawową, a następnie jej aliasy. Poza tworzeniem synonimów mamy tutaj możliwość tworzenia encji za pomocą wyrażeń regularnych lub fuzzy matching, gdzie dopasowanie nie musi być w stu procentach dokładne.

Istnieją również predefiniowane moduły, które można dodać do bota. Jednym z nich jest Small Talk. Dostajemy tam paczkę pogrupowanych wiadomości użytkowników, którą wypełniamy odpowiedziami naszego bota.

Small Talk

SAVE

Your agent can learn how to support small talk without any extra development. By default, it will respond with predefined phrases. Use the form below to customize responses to the most popular requests.

```
User: How are you?  
Agent: Wonderful as always. Thanks for asking.  
  
User: You're so sweet.  
Agent: Thanks! The feeling is mutual.
```

Enable

Small Talk Customization Progress

 6%

 About agent

 44%

About agent to tylko jeden z tematów, który można wypełnić.

Ważną zakładką jest dla nas również Training. Mamy tam dostęp do konwersacji z naszym botem. Kiedy w trakcie testowania naszego rozwiązania natrafimy na przypadek, gdzie bot niepoprawnie zareagował do wiadomości, możemy to poprawić.

i want a mild big rollo

Dec 12 2 REQUESTS 1 NO MATCH

USER SAYS i want a mild big rollo

PARAMETER NAME	ENTITY	RESOLVED VALUE	
sauce	@sauce	mild	×
size	@size	big	×
typeofitem	@typeOfItem	rollo	×

INTENT customer kebabshop

CONTEXT customerkebabshop-followup

OUT

USER SAYS 20

INTENT Click to assign

CLOSE

APPROVE

Na przykładzie takiej wymiany zdań widzimy też jak bot przetwarza wiadomość. Możemy poprawić go ustawiając intencję tam, gdzie nie została przez niego znaleziona. W taki sposób można przeglądać wszystkie testowe konwersacje lub nawet te, które odbywają się z użytkownikami korzystającymi z naszej aplikacji. Narzędzie to pozwala nam ciągle usprawniać działanie naszego chatbota, karmiąc go nowymi scenariuszami rozmów.

Przykład działania

Odpowiedzi na zielonym tle należą do KebaBota

Hello

Hey there!

I want to make an order

What kind of meal are you looking for?

I want burger

What size are you looking for?

big one

What sauce would you like?

spicy

What meat would you like?

chicken

We've received your big chicken spicy burger order! It will be delivered at one. Do you confirm?

yes, I do

That's great, your order has been taken

Jak widać chatbot sam przeprowadził nas przez proces zamówienia. Tak jak było to założone w tworzeniu intencji. Pytał nas krok po kroku o potrzebne informacje. W momencie, gdy wszystkie zostały zebrane potwierdził je.