

# Testowanie kontekstu rozmowy chatbota

*W skład tego wchodzi prowadzenie zwykłej rozmowy, pomoc w rozwiązaniu problemu (np. wyszukanie odpowiednich treści w aplikacji), odciążenie konsultantów, dokonanie zmiany statusu produktu oferowanego przez instytucję finansową itd.. Sam proces nauki języka przez bota to nie wszystko. Oprócz poprawnego rozpoznawania wpisywanych zdań, każdy bot musi umieć prowadzić rozmowę tak, aby odpowiedzieć na zestaw pytań danego użytkownika, ewentualnie prowadzić tą rozmowę tak, aby użytkownik osiągnął zamierzony cel. Niniejszy artykuł proponuje strategię, która umożliwi testowanie wybranych problemów rozumienia kontekstu przez chatbota.*

## 1. Zrozumienie kontekstu przez Bota

Każdy chatbot jest z natury inteligentny w rozumieniu programistycznym. Jest zdolny do wykazania pewnego poziomu inteligencji poprzez wykorzystanie technologii takich jak: uczenie maszynowe, big data, przetwarzanie języka naturalnego itp. Zrozumienie kontekstu przez bota jest dość trudnym dla niego aspektem. Aby to osiągnąć, bot musi analizować nie tylko słowa, ale także dane wejściowe jak czas, liczba, kwota, 4 ostatnie znaki karty, 3 ostatnie znaki numeru telefonu, dzień, data, historia rozmowy, ton, struktura zdania itp. [1] Istnieją zdania w każdym języku, które są bardzo trudne dla bota do zrozumienia:

- Chcę zamówić pizzę, nie obiad,
- Chcę herbatę, nie kawę
- Świetnie! Już pół roku na to czekam i nic!

To ostatnie zdanie wcale nie wyraża zadowolenia człowieka. Natomiast bot, analizując słowa, może zrozumieć kontekst tego zdania pozytywnie i wpaść w błędny kontekst rozmowy. Bot powinien przede wszystkim rozpoznawać użytkownika, zapamiętywać historię rozmów ze wszystkimi jej parametrami. [1]. Twórcy bota powinni dobrze przemyśleć proces definiowania oraz tworzenia jego kontekstu, aby nie tracić oraz nie wykonywać pracy, która okaże się niepotrzebna. Jeśli chatbot ma rozwiązać konkretne problemy użytkownika czy udzielić informacji dotyczące np. produkty finansowego, to dobrą strategią będzie zdefiniowanie oraz utworzenie scenariuszy dla tych problemów.

Same scenariusze jednak nie wystarczą. Niezwykle ważnym czynnikiem, zapewniającym dobrą jakość bota, jest skuteczna klasyfikacja przez niego zdań (fraz), które wprowadza osoba pisząca. Dopiero połączenie tych dwóch mechanizmów pozwoli utworzyć bota, który sprosta postawionym przez twórców wymaganiom. Problemy, na jakie narażony jest bot podczas uczenia się rozumienia języka naturalnego opisuje artykuł [2]. W dalszej części artykułu przyjmujemy założenie, że bot poprawnie klasyfikuje wprowadzone przez człowieka treści, natomiast proces nauki rozumienia przez niego kontekstu rozmowy odbywa się za pomocą odpowiednio zdefiniowanych scenariuszy.

## 2. Testowanie rozmów w cyklu wytwarzania bota. Testowanie kategoriami

W celu zrozumienia problemu testowania kontekstu bota przez czytelnika, wprowadzone zostaną definicje:

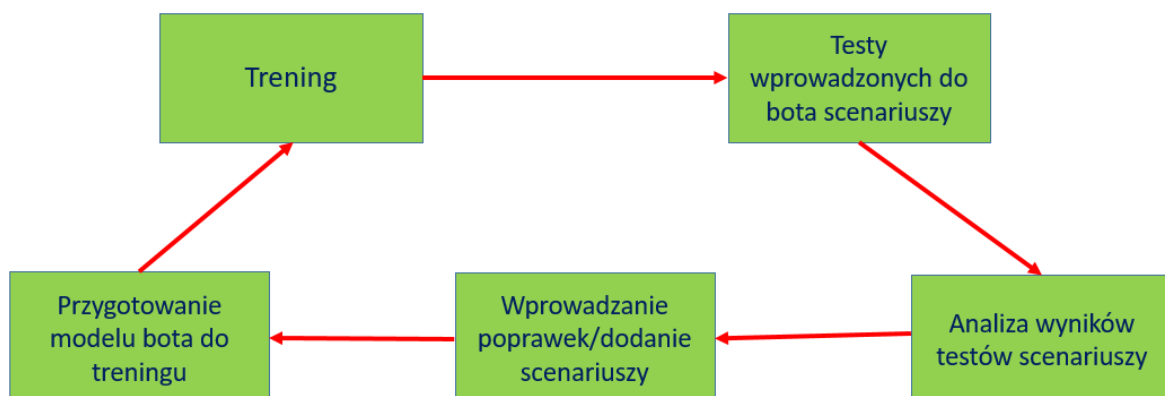
**Definicja 1.** Niech  $\{K_1, K_2, \dots, K_n\}$  będą kategoriami, które klasyfikuje bot (pogrupowane w kategorie zdania lub frazy) oraz niech  $\{a_1, a_2, \dots, a_m\}$  będzie zbiorem odpowiedzi bota (akcji). Uporządkowany ciąg, w którym wartościami są kategorie oraz akcje, będziemy nazywali scenariuszem rozmowy bota.

**Definicja 2.** Cykl bota względem kontekstu rozmów to czas pomiędzy dwoma sąsiednimi treningami wraz ze wszystkimi zmianami, które zostały wprowadzone w scenariuszach bota od strony samych scenariuszy oraz od strony technicznej (obsługa zachowania bota kodem programu).

Stabilizacja kontekstu rozmowy bota polega na utworzeniu pewnej liczby kombinacji scenariuszy, które mają wspólny główny temat. Analogicznie jak w pozycji [2], podejście do testowania scenariuszy rozmów może być poprowadzone poprzez:

- automatyzację testów za pomocą napisania własnego frameworka oraz własnego mechanizmu generowania raportów z odpowiednimi metrykami,
- wykorzystanie do testów środowiska, które stanowi silnik bota,
- pomieszczenie dwóch wyżej wymienionych strategii.

Każdy chatbot, podczas procesu jego wytwarzania, poddawany jest na ogół cyklicznym treningom. Jednak ilość fraz, które mogą zostać poddane testom (w testach samych scenariuszy) jest dość duża i nietestowalna manualnie (ze względu na czas testów). Dlatego testowanie schematu rozmowy (rozumiane jako regresja testów scenariuszy) powinno być na tyle zoptymalizowane, aby w miarę szybko otrzymać wyniki testów. Ogólny schemat testowania samych scenariuszy prezentuje niżej umieszczony schemat:



Rysunek 1: Cykliczny schemat treningu i poprawy scenariuszów bota

Przykład pomoże zrozumieć wyżej opisane treści.

**Przykład 1.** Załóżmy, że bot potrafi udzielać informacji o ubezpieczeniach samochodu. W tym celu utworzone są takie kategorie językowe (przewidziane jako te, które wprowadzi piszący z botem) jak:

- *Ubezpieczenie\_Samochodu\_ZapytanieKlienta*,
- *Ubezpieczenie\_OC*
- *Pojazd\_RokProdukcji*
- *Pojazd\_Marka*
- *Prawojazdy\_CzasPosiadania*
- *Kategoria\_Zaprzeczenie*

Odpowiedzi bota, które są niezbędne w każdym scenariuszu będą miały postać:

- *Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*: Udzielam ubezpieczenia zarówno OC jak i AC, proszę o informację, którym ubezpieczeniem jest Pan zainteresowany.
- *Akcja\_Marka\_Samochod*: Proszę o podanie marki samochodu.
- *Akcja\_Pojazd\_RokProdukcji*: Proszę o rok produkcji samochodu.
- *Akcja\_Prawojazdy\_CzasPosiadania*: Proszę o informację jak długo posiada Pan prawo jazdy.
- *Akcja\_WypadekDrogowy*: Czy kiedykolwiek spowodował Pan wypadek drogowy?
- *Akcja\_OfertaUbezpieczenia*: Oferta jaką mogę Panu zaoferować to OC ze składką 500 PLN rocznie. Jeżeli jest Pan zainteresowany zapraszam do złożenia wniosku w linku: <link>.

Kwota ubezpieczenia w ostatniej odpowiedzi bota to wartość zmienna, dlatego jest oznaczona żółtym tłem, analogicznie jak zmienne w pozycji [2]. Scenariusz oferty ubezpieczenia definiujemy zgodnie ze schematem:

#### **[Oferta\_Ubezpieczenia\_OC\_BezWypadku]**

- ^ *Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
  - *Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Marka\_Samochod*
- ^ *Pojazd\_Marka*
  - *Akcja\_Pojazd\_RokProdukcji*
- ^ *Akcja\_Pojazd\_RokProdukcji*
  - *Akcja\_Prawojazdy\_CzasPosiadania*
- ^ *Prawojazdy\_CzasPosiadania*
  - *Akcja\_WypadekDrogowy*
- ^ *Kategoria\_Zaprzeczenie*
  - *Akcja\_OfertaUbezpieczenia*

Założmy, że każda kategoria (wpisane słowa człowieka) z wyżej ułożonego scenariusza ma po 100 fraz:

Nazwa Kategorii	Treningowe zdania dla kategorii
-----------------	---------------------------------

<p><i>Ubezpieczenie_Samochodu_ZapytanieKlienta</i></p>	<ol style="list-style-type: none"> <li>1. Dzień dobry chciałbym ubezpieczyć samochód.</li> <li>2. Chcę ubezpieczyć samochód</li> <li>3. Chciałbym ubezpieczyć swój samochodu</li> <li>4. Czy mogę ubezpieczyć u was samochód?</li> <li>5. Czy możecie ubezpieczyć mój samochód?</li> <li>6. Mogę ubezpieczyć samochód waszej firmie?</li> <li>7. Sprawdź mi czy da radę ubezpieczyć mój samochód</li> <li>8. Czy w waszej ubezpieczalni można ubezpieczyć samochody?</li> <li>9. Da się u was ubezpieczyć mój samochodzik?</li> <li>10. Chcę sprawdzić czy dałoby radę ubezpieczyć u was mój samochodzik.</li> </ol> <p>.....</p> <p>100. Powiedz mi czy mógłbym tu ubezpieczyć mój samochód.</p>
<p><i>Ubezpieczenie_OC</i></p>	<ol style="list-style-type: none"> <li>1. Interesuje mnie ubezpieczenie OC.</li> <li>2. OC</li> <li>3. Chcę OC</li> <li>4. Jestem zdecydowany na OC</li> <li>5. Zależy mi tylko na OC</li> <li>6. Nie chcę AC, chcę OC</li> <li>7. Na razie tylko OC</li> <li>8. Preferuję OC</li> <li>9. Mam na myśli OC</li> <li>10. Ja chcę OC</li> </ol> <p>.....</p> <p>100. OC wystarczy.</p>
<p><i>Pojazd_Marka</i></p>	<ol style="list-style-type: none"> <li>1. Fiat 126p</li> <li>2. Fiacik 126p</li> <li>3. Mam Audi</li> <li>4. Ja mam Audi A3</li> <li>5. Chcę ubezpieczyć Toyote</li> <li>6. Chodzi mi ubezpieczyć Toyote</li> <li>7. Myślę o ubezpieczeniu dla Suzuki</li> <li>8. To miałyby być ubezpieczenie Suzuki</li> <li>9. Zależy mi żeby ubezpieczyć BMW</li> <li>10. Moje autko to BMW</li> </ol> <p>....</p> <p>100. A mam Golfa</p>
<p><i>Pojazd_RokProdukcji</i></p>	<ol style="list-style-type: none"> <li>1. 1989 r.</li> <li>2. Ten rok to 2003</li> <li>3. To rok 1999</li> <li>4. 1989 rocznik</li> <li>5. Rocznik to 2011</li> <li>6. Jego rocznik to 2017</li> <li>7. Jest to rok 2006</li> <li>8. Był to rok 1999</li> <li>9. Mówię o roczniku 2001</li> <li>10. To było roku 2003</li> </ol> <p>...</p> <p>100. Rok 2011</p>
<p><i>Prawojazdy_CzasPosiadania</i></p>	<ol style="list-style-type: none"> <li>1. Mam prawo jazdy od 7 lat</li> <li>2. Mam prawo jazdy 15 lat</li> </ol>

	3. Już 5 lat mam prawo jazdy 4. Mam prawo jazdy równo 25 lat 5. A no będzie już 2 lata od kiedy mam prawo jazdy 6. Od kiedy mam prawo jazdy minęło 7 lat 7. O ja to mam prawo jazdy już 8 lat 8. Zdaje się, że ma prawo jazdy od 9 lat 9. Od 5 lat jestem posiadaczem prawa jazdy 10. Mam prawko od 11 lat ..... 100. Ja mam prawko do jazdy od 12 lat.
<i>Kategoria_Zaprzeczenie</i>	1. Nigdy. 2. Nie 3. Ależ skąd 4. Nope 5. Oczywiście, że nie 6. Absolutnie 7. Niet 8. Na szczęście nie 9. Nie, nigdy 10. Nie nie nie ..... 100. Oczywiście, że nie

Tabela 1: Tabela zawierająca opis kategorii wraz z frazami

Ilość kombinacji testów scenariusza **Oferta\_Ubezpieczenia\_OC\_BezWypadku** jest równa 600, jeśli testy zostaną przeprowadzone tylko za pomocą danych uczących. Jeśli bot przykładowo zawiera 200 scenariuszy, to manualne testowanie nie wchodzi w grę ze względu na czas trwania testów.

Istnieje jednak pewna strategia, która pozwoli **zmniejszyć ilość testów z liczby 600 tylko do jednego testu – jest to testowanie scenariusza za pomocą podawania kategorii**. Jeżeli twórcy bota podejną do testowania scenariuszy w ten sposób, że testy będą możliwe do przeprowadzenia poprzez podanie kategorii na „wejście”. Przy takim podejściu testy automatyczne (niewykluczone, że również testy manualne) będą miały schemat zbliżony do definicji scenariusza:

#### **[@Test\_OfertaUbezpieczenia\_BezWypdku]**

<b>^ Ubezpieczenie_Samochodu_ZapytanieKlienta</b>	<b>– wejście_1</b>
- Akcja_Ubezpieczenie_Samochodu_ZapytanieKlienta	<b>– warunek oczekiwany_1</b>
<b>^ Ubezpieczenie_OC</b>	<b>– wejście_2</b>
- Akcja_Marka_Samochod	<b>– warunek oczekiwany_2</b>
<b>^ Pojazd_Marka</b>	<b>– wejście_3</b>
- Akcja_Pojazd_RokProdukcji	<b>– warunek oczekiwany_3</b>
<b>^ Akcja_Pojazd_RokProdukcji</b>	<b>– wejście_4</b>
- Akcja_Prawojazdy_CzasPosiadania	<b>– warunek oczekiwany_4</b>
<b>^ Prawojazdy_CzasPosiadania</b>	<b>– wejście_5</b>
- Akcja_WypadekDrogowy	<b>– warunek oczekiwany_5</b>
<b>^ Kategoria_Zaprzeczenie</b>	<b>– wejście_6</b>
- Akcja_OfertaUbezpieczenia	<b>– warunek oczekiwany_6</b>

Takie podejście pozwoli znacznie zmniejszyć ilość testów, ale ma jedną wadę. Nauczenie się scenariuszy przez bota jest testowane oddzielnie niż klasyfikacja. W takiej sytuacji twórcy bota muszą być pewni, że klasyfikacja oraz testy języka zostały wykonane z akceptowalnym wynikiem zgodnie z miarami dla języka [2]. Użytkownik, który rozmawia z botem nie zna kategorii, wpisuje tylko frazy. Wobec tego twórcy bota muszą zadbać o to, aby testy języka i jego rozumienia dawały bardzo dobre podparte miarami wyniki, zanim bot zacznie uczyć się kontekstu [2]. Pomimo tego, że ilość scenariuszy może osiągnąć liczbę 200,300 a nawet 400, testerzy nie mogą rezygnować ze zwykłych manualnych testów. Analogicznie jak w każdym systemie informatycznym, zarówno w chatbocie testowanie manualne jest potrzebne, ponieważ mogą wyjść błędy, które nie zostaną wykryte przez automat.

### 3. Testowanie kontekstu

Testowanie zrozumienia kontekstu rozmowy przez bota może być przeprowadzone na wiele sposobów. Wybór strategii testowania kontekstu rozmowy jest silnie uzależniony od przeznaczenia bota. Jeżeli bota jest przeznaczony do tego, aby udzielać kredytów, to musi rozpoznać liczbę rat, wiedzieć jakie jest dolne i górne ograniczenie tych rat (bank nikomu nie da kredytu na 5000 rat). Z kolei kredyty gotówkowe na ogół dostaje się na liczbę rat odpowiadającą liczbie pełnych lat, czyli podzieloną przez 12. Bot wobec tego musi wiedzieć, że nie może dać kredytu gotówkowego na 13 rat bądź 53 raty. Wybrane problemy testowania kontekstu oraz scenariuszy rozmów zostaną zaprezentowane w kolejnych rozdziałach.

#### 3.1 Brak zrozumienia kontekstu bota

Podczas rozmowy z botem, może zdarzyć się taka sytuacja, że bot nie będzie potrafił odnaleźć się w kontekście rozmowy. Taka sytuacja może wystąpić w kilku przypadkach:

- użytkownik wpisze frazę, której bot nie zrozumie,
- użytkownik wpisze frazę, którą bot potrafi sklasyfikować, ale ta fraza nie pasuje do kontekstu rozmowy oraz nie jest początkiem innego scenariusza wprowadzonego do danych uczących bota.

Problem ze zrozumieniem frazy jest dokładnie opisany w pozycji [2], zatem w tym rozdziale skupimy się na frazach, które chatbot rozumie ale nie potrafi ich przypisać do kontekstu rozmowy. W celu dokładnego wyjaśnienia problemu wykorzystamy scenariusz z rozdziału:

#### **[Oferta\_Ubezpieczenia\_OC\_BezWypadku]**

*^ Ubezpieczenie\_Samochodu\_ZapytanieKlienta*

*- Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*

*^ Ubezpieczenie\_OC*

*- Akcja\_Marka\_Samochod*

*^ Pojazd\_Marka*

*- Akcja\_Pojazd\_RokProdukcji*

*^ Akcja\_Pojazd\_RokProdukcji*

*- Akcja\_Prawojazdy\_CzasPosiadania*

*^ Prawojazdy\_CzasPosiadania*

- Akcja\_WypadekDrogowy
- ^ Kategoria\_Zaprzeczenie
  - Akcja\_OfertaUbezpieczenia

Założmy, że istnieje kategoria *Pojazd\_NrVIN*, która zawiera frazy typu:

- Mój numer VIN to WBA5L31000G482017
- Mam w aucie nr VIN WOLPD8EGXB8114596
- Mam VIN WBAAS71080FY30158
- ...

oraz nie jest początkiem żadnego scenariusza (występuje tylko w środku scenariuszy i nie ma odpowiedzi 1:1). Skoro bot potrafi zrozumieć kontekst. Przykładem braku zrozumienia kontekstu przez bota może być taka rozmowa użytkownika:

### Przykład 2.

#### [Oferta\_Ubezpieczenia\_OC\_BezWypadku]

- ^ Ubezpieczenie\_Samochodu\_ZapytanieKlienta
  - Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta
- ^ Ubezpieczenie\_OC
  - Akcja\_Marka\_Samochod
- ^ Pojazd\_Marka
  - Akcja\_Pojazd\_RokProdukcji
- ^ **Pojazd\_NrVIN**

W takim przypadku bot nie będzie wiedział co ma odpowiedzieć, a jego twórcy powinni się zabezpieczyć przed taką sytuacją. Taki stan rzeczy ma miejsce dlatego, że nie wszystkie kategorie bota są początkiem scenariusza bo nie miałyby to sensu w niektórych przypadkach, a twórcy bota nie są w stanie przewidzieć wszystkich wtrąceń użytkownika i zdefiniować do nich odpowiedzi. Reakcja bota na „przyznanie się” do niezrozumienia kontekstu należy do twórców bota. Może to być np. akcja z tekstem oraz przyciskami:

„Nie potrafię zrozumieć Twojego kontekstu rozmowy. W tej sytuacji nie mogę pomóc, ale mogę połączyć Cię z agentem. Jeśli nie wyrażasz zgody na połączenie, to zrezygnuj z rozmowy. Wybierz odpowiedni przycisk.”

Rezygnuję z rozmowy

Połącz z agentem

Twórcy bota powinni wiedzieć po każdym treningu, jak bot radzi sobie z wyżej opisaną sytuacją. W tym celu doskonale sprawdzą się testy automatyczne zdefiniowane za pomocą kategorii i akcji. Założmy, że akcja wyświetlająca brak zrozumienia kontekstu przez bota ma nazwę

*Akcja\_BrakZrozumieniaKontekstu*. Testy automatyczne, które weryfikują wspomniany wcześniej brak zrozumienia mogą mieć postać:

**[@Test\_Oferta\_Ubezpieczenia\_OC\_BezWypadku\_1]**

- ^ *Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
  - *Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Marka\_Samochod*
- ^ *Pojazd\_Marka*
  - *Akcja\_Pojazd\_RokProdukcji*
- ^ *Pojazd\_NrVIN*
  - *Akcja\_BrakZrozumieniaKontekstu*
- \* *Przycisk\_Rezygnacji\_Z\_Rozmowy*
  - *Akcja\_Rezygnacja\_z\_Rozmowy*

oraz

**[@Test\_Oferta\_Ubezpieczenia\_OC\_BezWypadku\_2]**

- ^ *Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
  - *Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Marka\_Samochod*
- ^ *Pojazd\_Marka*
  - *Akcja\_Pojazd\_RokProdukcji*
- ^ *Pojazd\_NrVIN*
  - *Akcja\_BrakZrozumieniaKontekstu*
- \* *Przycisk\_Przelaczenie\_Do\_Agenta*
  - *Akcja\_Przelaczenie\_Do\_Agenta*

Wyżej przedstawione testy mają za zadanie sprawdzić, czy bot zachowa się prawidłowo w sytuacji, gdy nie zrozumie kontekstu użytkownika. W pierwszym przypadku test sprawdza, czy nastąpi zakończenie rozmowy, a w drugim przypadku będzie sprawdzona akcja przełączenia do konsultanta. Test, który sprawdziłby pojawienie się akcji *Akcja\_BrakZrozumieniaKontekstu* nie byłby wystarczający, ponieważ istnieją dwa zachowania, które umożliwia bot dla użytkownika w sytuacji braku zrozumienia kontekstu.

### 3.2 Testowanie zmiany kontekstu

W trakcie przebiegu rozmów z chatbotem nieuniknione są takie sytuacje, podczas których klient nie przeprowadzi rozmowy od początku do końca w taki sposób w jaki zaplanowali to Twórcy bota. Inaczej mówiąc, osoba pisząca nie odwzoruje jednego, konkretnego wyuczonego scenariusza. Zmiany kontekstu rozmowy przez osobę piszącą będą zdarzały się dość często. Każdy bot powinien umieć sobie radzić ze zmianą kontekstu, ponieważ, nie dojdzie do takiej sytuacji, żeby 100% rozmów odbywało się w taki sposób, w jaki są zdefiniowane zaplanowane scenariusze.



Bot musi być zaprojektowany przez twórców tak, aby potrafił przerwać scenariusz A oraz zacząć rozmawiać wg scenariusza B, jeśli A nie zostanie poprowadzony do ostatniej zdefiniowanej wypowiedzi. W celu lepszego zrozumienia problemu poniżej przedstawiony zostanie przykład opisujący takie zjawisko:

### Przykład 3.

Założmy, że mamy zdefiniowane dwa scenariusze:

#### [Oferta\_Ubezpieczenia\_OC\_BezWypadku]

- ^ *Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
  - *Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Marka\_Samochod*
- ^ *Pojazd\_Marka*
  - *Akcja\_Pojazd\_RokProdukcji*
- ^ *Pojazd\_RokProdukcji*
  - *Akcja\_Prawojazdy\_CzasPosiadania*
- ^ *Prawojazdy\_CzasPosiadania*
  - *Akcja\_WypadekDrogowy*
- ^ *Kategoria\_Zaprzeczenie*
  - *Akcja\_OfertaUbezpieczenia*

oraz

#### [Oferta\_Ubezpieczenia\_Samochód\_Z\_Zagranicy]

- ^ *Ubezpieczenie\_Samochodu\_Z\_Zagranicy*
  - *Akcja\_Ubezpieczenie\_Samochodu\_Z\_Zagranicy*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Oferta\_OC*

W przypadku gdyby człowiek będzie rozmawiał z botem, najpierw zgodnie ze scenariuszem **Oferta\_Ubezpieczenia\_OC\_BezWypadku**, to problem zmiany kontekstu nie występuje. Musimy jednak wziąć pod uwagę taką możliwość, że człowiek podczas rozmowy z botem zacznie innym kontekst rozmowy po odpowiedzi np. *Akcja\_Pojazd\_RokProdukcji*, czyli wpisze frazę z kategorii *Ubezpieczenie\_Samochodu\_Z\_Zagranicy*, która jest początkiem innego kontekstu rozmowy, a mianowicie początkiem scenariusza **Oferta\_Ubezpieczenia\_Samochód\_Z\_Zagranicy**. Schemat rozmowy w takim przypadku będzie miał taki przebieg:

- ^ *Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
  - *Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Marka\_Samochod*
- ^ *Pojazd\_Marka*
  - *Akcja\_Pojazd\_RokProdukcji*
- ^ ***Ubezpieczenie\_Samochodu\_Z\_Zagranicy*** (Zmiana kontekstu rozmowy)
  - *Akcja\_Ubezpieczenie\_Samochodu\_Z\_Zagranicy*

*Ubezpieczenie\_OC*

- *Akcja\_Oferta\_OC*

Wyżej opisana rozmowa może być przykładem automatycznego testu, który sprawdza zachowanie bota przy zmianie kontekstu rozmowy. Mając przykładowo 200-300 scenariuszy, nie jest możliwe manualne testowanie zmian kontekstu po każdym treningu zgodnie z definicją 2.

Założmy, że oprócz scenariuszy z przykładu 3, mamy również w zbiorze treningowym bota takie scenariusze:

### **[Oferta\_Ubezpieczenie\_Maszyny]**

*^ Ubezpieczenie\_Maszyny\_ZapytanieKlienta*

- *Akcja\_Ubezpieczenie\_Maszyny\_ZapytanieKlienta*

*^ Rodzaj\_Maszyny*

- *Akcja\_Maszyna\_RokProdukcji*

*^ Maszyna\_Rok\_Produkcyj*

- *Akcja\_Kraj\_Produkcji*

*^ Maszyna\_Kraj\_Produkcji*

- *Akcja\_Maszyna\_Waga*

*^ Maszyna\_Waga*

- *Akcja\_Oferta\_KwotaUbezpieczeniaRoczna*

oraz

### **[Ubezpieczenie\_Polisa\_Życie]**

*^ Ubezpieczenie\_PolisaNaŻycie*

- *Akcja\_Ubezpieczenie\_PolisaNaŻycie*

- *Akcja\_Wiek\_Osoby*

*^ Ubezpieczenie\_Wiek\_Osoby*

- *Akcja\_Sporty\_Ekstremalne*

*^ Ubezpieczenie\_Sporty\_Ekstremalne*

- *Akcja\_Palenie\_Papierosy*

*^ Ubezpieczenie\_Palenie\_Papierosy*

- *Akcja\_Choroby\_Przewlekłe*

*^ Ubezpieczenie\_Choroby\_Przewlekłe*

- *Akcja\_Miejsce\_Zamieszkanie*

*^ Ubezpieczenie\_Miejsce\_Zamieszkanie*

- *Akcja\_Wysokość\_Składka*

*^ Ubezpieczenie\_Wysokość\_Składka*

- *Akcja\_Okres\_Ubezpieczenie*

*^ Ubezpieczenie\_Okres*

- *Akcja\_Oferta\_UbezpieczenieNaŻycie*

Testy, które zweryfikują zachowanie bota podczas zmiany kontekstu użytkownika mogą mieć postać:

**[@Test\_Oferta\_Ubezpieczenia\_OC\_BezWypadku& Oferta\_Ubezpieczenie\_Maszyny]**

- ^ *Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
  - *Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Marka\_Samochod*
- ^ *Pojazd\_Marka*
  - *Akcja\_Pojazd\_RokProdukcji*
- ^ *Pojazd\_RokProdukcji*
  - *Akcja\_Prawojazdy\_CzasPosiadania*
- ^ *Ubezpieczenie\_Samochodu\_Z\_Zagranicy* (zmiana kontekstu)
  - *Akcja\_Ubezpieczenie\_Samochodu\_Z\_Zagranicy*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Oferta\_OC*

**[@Test\_Oferta\_Ubezpieczenia\_Samochód\_Z\_Zagranicy&Test\_Oferta\_Ubezpieczenia\_OC\_BezWypadku]**

- ^ *Ubezpieczenie\_Samochodu\_Z\_Zagranicy*
  - *Akcja\_Ubezpieczenie\_Samochodu\_Z\_Zagranicy*
- ^ *Ubezpieczenie\_Samochodu\_ZapytanieKlienta* (zmiana kontekstu)
  - *Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Marka\_Samochod*
- ^ *Pojazd\_Marka*
  - *Akcja\_Pojazd\_RokProdukcji*
- ^ *Pojazd\_RokProdukcji*
  - *Akcja\_Prawojazdy\_CzasPosiadania*
- ^ *Prawojazdy\_CzasPosiadania*
  - *Akcja\_WypadekDrogowy*
- ^ *Kategoria\_Zaprzeczenie*
  - *Akcja\_OfertaUbezpieczenia*

**[@Test\_Oferta\_Ubezpieczenia\_OC\_BezWypadku&Oferta\_Ubezpieczenia\_Samochód\_Z\_Zagranicy]**

- ^ *Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
  - *Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*
- ^ *Ubezpieczenie\_OC*
  - *Akcja\_Marka\_Samochod*
- ^ *Pojazd\_Marka*
  - *Akcja\_Pojazd\_RokProdukcji*
- ^ *Ubezpieczenie\_Maszyny\_ZapytanieKlienta* (zmiana kontekstu)
  - *Akcja\_Ubezpieczenie\_Maszyny\_ZapytanieKlienta*
- ^ *Rodzaj\_Maszyny*
  - *Akcja\_Maszyna\_RokProdukcji*
- ^ *Maszyna\_Rok\_Prokucji*
  - *Akcja\_Kraj\_Produkcji*
- ^ *Maszyna\_Kraj\_Produkcji*

- Akcja\_Maszyna\_Waga
- ^ Maszyna\_Waga
  - Akcja\_Oferta\_KwotaUbezpieczeniaRoczna

**[@Test\_Oferta\_Ubezpieczenie\_Maszyny&Ubezpieczenie\_Polisa\_Życie]**

- ^ Ubezpieczenie\_Maszyny\_ZapytanieKlienta
  - Akcja\_Ubezpieczenie\_Maszyny\_ZapytanieKlienta
- ^ Rodzaj\_Maszyny
  - Akcja\_Maszyna\_RokProdukcji
- ^ Maszyna\_Rok\_Produkcyj
  - Akcja\_Kraj\_Produkcyj
- ^ Ubezpieczenie\_PolisaNaŻycie (zmiana kontekstu)
  - Akcja\_Ubezpieczenie\_PolisaNaŻycie
  - Akcja\_Wiek\_Osoby
- ^ Ubezpieczenie\_Wiek\_Osoby
  - Akcja\_Sporty\_Ekstremalne
- ^ Ubezpieczenie\_Sporty\_Ekstremalne
  - Akcja\_Palenie\_Papierosy
- ^ Ubezpieczenie\_Palenie\_Papierosy
  - Akcja\_Choroby\_Przewlekłe
- ^ Ubezpieczenie\_Choroby\_Przewlekłe
  - Akcja\_Miejsce\_Zamieszkanie
- ^ Ubezpieczenie\_Miejsce\_Zamieszkanie
  - Akcja\_Wysokość\_Składka
- ^ Ubezpieczenie\_Wysokość\_Składka
  - Akcja\_Okres\_Ubezpieczenie
- ^ Ubezpieczenie\_Okres
  - Akcja\_Oferta\_UbezpieczenieNaŻycie

**[@Test\_Ubezpieczenie\_Polisa\_Życie&Oferta\_Ubezpieczenie\_Maszyny]**

- ^ Ubezpieczenie\_PolisaNaŻycie
  - Akcja\_Ubezpieczenie\_PolisaNaŻycie
  - Akcja\_Wiek\_Osoby
- ^ Ubezpieczenie\_Wiek\_Osoby
  - Akcja\_Sporty\_Ekstremalne
- ^ Ubezpieczenie\_Sporty\_Ekstremalne
  - Akcja\_Palenie\_Papierosy
- ^ Ubezpieczenie\_Maszyny\_ZapytanieKlienta (zmiana kontekstu)
  - Akcja\_Ubezpieczenie\_Maszyny\_ZapytanieKlienta
- ^ Rodzaj\_Maszyny
  - Akcja\_Maszyna\_RokProdukcji
- ^ Maszyna\_Rok\_Produkcyj
  - Akcja\_Kraj\_Produkcyj
- ^ Maszyna\_Kraj\_Produkcyj
  - Akcja\_Maszyna\_Waga

^ Maszyna\_Waga

- Akcja\_Oferta\_KwotaUbezpieczeniaRoczna

Sens istnienia tego rodzaju testów jest taki, że twórcy bota będą wiedzieli po dym treningu, czy bot będzie umiał odnaleźć się w sytuacji zmiany kontekstu czy będzie próbował kontynuować „stary” kontekst, co spowoduje udzielanie niepoprawnych odpowiedzi i sprawi wrażenie, że bot jest mało inteligentny. Przykład błędu zrozumienia kontekstu mógłby być np. taki:

**[@Test\_Oferta\_Ubezpieczenia\_OC\_BezWypadku& Oferta\_Ubezpieczenie\_Maszyny]**

^ Ubezpieczenie\_Samochodu\_ZapytanieKlienta

- Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta

^ Ubezpieczenie\_OC

- Akcja\_Marka\_Samochod

^ Pojazd\_Marka

- Akcja\_Pojazd\_RokProdukcji

^ Pojazd\_RokProdukcji

- Akcja\_Prawojazdy\_CzasPosiadania

^ Ubezpieczenie\_Samochodu\_Z\_Zagranicy (zmiana kontekstu)

- **Akcja\_WypadekDrogowy**

Twórcy chatbota, oprócz zmiany kontekstu wypowiedzi w środku scenariusza, muszą wiedzieć, że bot prawidłowo zacznie prowadzić nowy kontekst, również w przypadku, gdy poprzedni scenariusz przejdzie od początku do końca. Niżej przedstawiony jest test, który weryfikuje taką sytuację:

**Przykład 4.**

**[@Test\_Ubezpieczenia\_OC\_BezWypadku&Oferta\_Ubezpieczenie\_Maszyny&Polisa\_Życie]**

^ Ubezpieczenie\_Samochodu\_ZapytanieKlienta (**Oferta\_Ubezpieczenia\_OC\_BezWypadku**)

- Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta

^ Ubezpieczenie\_OC

- Akcja\_Marka\_Samochod

^ Pojazd\_Marka

- Akcja\_Pojazd\_RokProdukcji

^ Pojazd\_RokProdukcji

- Akcja\_Prawojazdy\_CzasPosiadania

^ Prawojazdy\_CzasPosiadania

- Akcja\_WypadekDrogowy

^ Kategoria\_Zaprzeczenie

- Akcja\_OfertaUbezpieczenia

^ Ubezpieczenie\_Maszyny\_ZapytanieKlienta (**Oferta\_Ubezpieczenie\_Maszyny**)

- Akcja\_Ubezpieczenie\_Maszyny\_ZapytanieKlienta

^ Rodzaj\_Maszyny

- Akcja\_Maszyna\_RokProdukcji

^ Maszyna\_Rok\_Prokukcji

- Akcja\_Kraj\_Produkcji

^ Maszyna\_Kraj\_Produkcji

- Akcja\_Maszyna\_Waga
- ^ Maszyna\_Waga
  - Akcja\_Oferta\_KwotaUbezpieczeniaRoczna
- ^ Ubezpieczenie\_PolisaNaŻycie (Ubezpieczenie\_Polisa\_Życie)
  - Akcja\_Ubezpieczenie\_PolisaNaŻycie
  - Akcja\_Wiek\_Osoby
- ^ Ubezpieczenie\_Wiek\_Osoby
  - Akcja\_Sporty\_Ekstremalne
- ^ Ubezpieczenie\_Sporty\_Ekstremalne
  - Akcja\_Palenie\_Papierosy
- ^ Ubezpieczenie\_Palenie\_Papierosy
  - Akcja\_Choroby\_Przewlekłe
- ^ Ubezpieczenie\_Choroby\_Przewlekłe
  - Akcja\_Miejsce\_Zamieszkanie
- ^ Ubezpieczenie\_Miejsce\_Zamieszkanie
  - Akcja\_Wysokość\_Składka
- ^ Ubezpieczenie\_Wysokość\_Składka
  - Akcja\_Okres\_Ubezpieczenie
- ^ Ubezpieczenie\_Okres
  - Akcja\_Oferta\_UbezpieczenieNaŻycie

Test z wyżej przedstawionego przykładu sprawdza, czy bot będzie umiał przejść przez 3 scenariusze po kolei. Różni się on od poprzednich testów z tego rozdziału tym, że kontekst rozmowy nie jest przerywany w połowie czy w środku scenariusza, tylko każdy scenariusz przechodzi od początku do końca. Takie zachowanie powinno być poddawane testom obok testowania przerywania kontekstu rozmowy. Mechanizm, który umożliwi przejście testów z przykładu 4 możemy wykorzystać, aby zdefiniować przywitanie oraz pożegnanie się z botem przez użytkownika. Niżej mamy zdefiniowane scenariusze powitania i pożegnań:

#### **Przykład 5.**

##### **[Powitanie]**

- ^ Powitanie
  - Akcja\_Powitanie

oraz

##### **[Pożegnanie]**

- ^ Pożegnanie
  - Akcja\_Pożegnanie

W celu sprawdzenia jak bot poradzi sobie z frazami typu: *Cześć, Halo, Siema, Dzień Dobry* itd. również można zastosować testy automatyczne, zaprojektowane za pomocą kategorii i akcji. Przykładami takich testów są:

## **[@Test\_Powitanie&Oferta\_Ubezpieczenia\_Samochód\_Z\_Zagranicy&Pożegnanie]**

### *^ Powitanie*

*- Akcja\_Powitanie*

### *^ Ubezpieczenie\_Samochodu\_Z\_Zagranicy*

*- Akcja\_Ubezpieczenie\_Samochodu\_Z\_Zagranicy*

### *^ Ubezpieczenie\_AC*

*- Akcja\_Oferta\_AC*

### *^ Pożegnanie*

*- Akcja\_Pożegnanie*

## **[@Test\_Powitanie&Oferta\_Ubezpieczenia\_OC\_BezWypadku&Oferta\_Ubezpieczenia\_Samochód\_Z\_Zagranicy&Pożegnanie]**

### *^ Powitanie*

*- Akcja\_Powitanie*

### *^ Ubezpieczenie\_Samochodu\_ZapytanieKlienta (Oferta\_Ubezpieczenia\_OC\_BezWypadku)*

*- Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta*

### *^ Ubezpieczenie\_OC*

*- Akcja\_Marka\_Samochod*

### *^ Pojazd\_Marka*

*- Akcja\_Pojazd\_RokProdukcji*

### *^ Pojazd\_RokProdukcji*

*- Akcja\_Prawojazdy\_CzasPosiadania*

### *^ Prawojazdy\_CzasPosiadania*

*- Akcja\_WypadekDrogowy*

### *^ Kategoria\_Zaprzeczenie*

*- Akcja\_OfertaUbezpieczenia*

### *^ Ubezpieczenie\_Samochodu\_Z\_Zagranicy (Oferta\_Ubezpieczenia\_Samochód\_Z\_Zagranicy)*

*- Akcja\_Ubezpieczenie\_Samochodu\_Z\_Zagranicy*

### *^ Ubezpieczenie\_OC*

*- Akcja\_Oferta\_OC*

### *^ Pożegnanie*

*- Akcja\_Pożegnanie*

Przypadki testowe mogą zawierać złączenia dwóch lub więcej scenariuszy, które są zdefiniowane w zbiorze scenariuszy bota. Dłuższe rozmowy z botem (kilka scenariuszy - jeden po drugim) mogą wystąpić, jeżeli użytkownik zacznie rozmawiać kategoriami, które są nazywana small talk luźne rozmowy dotyczące np. literatury, filmów, muzyki itd. Nie są to oczywiście jedyne przypadki kilku kontekstów rozmowy następujących po sobie. Jeśli instytucja posiadająca bota, który tłumaczy definicję np. 15 produktów z oferty tej instytucji, to teoretycznie o każdą z nich może zapytać człowiek podczas jednej rozmowy, z to oznacza, że bot musi poprowadzić 15 scenariuszy i umieć wejść w odpowiedni kontekst.

### 3.3 Testowanie odporności na zapętlenie

Osoby pracujące przy tworzeniu bota muszą się liczyć z tym, że dana kategoria rozmowy może występować w środku scenariusza lub może być początkiem kontekstu rozmowy. Analogiczny sposób rozmowy występuje również pomiędzy ludźmi. Twórcy bota powinni tak przygotować bota, aby umiał sobie poradzić z takimi sytuacjami, ponieważ w przeciwnym przypadku mógłby wpaść w pętlę, co nie jest pożądanym zachowaniem. Dla lepszego zrozumienia wcześniej przedstawionego problemu posłużymy się przykładami scenariuszy opisujących proces blokady karty w banku. Przedstawione scenariusze opisują proces blokowania karty w dwóch różnych sytuacjach:

#### **[Zgubienie\_Karta]**

##### **^ Zgubienie\_Karty**

- Akcja\_Blokowanie\_Lub\_Zastrzezenie

##### **^ Różnica\_Blokada\_Zastrzezenie**

- Akcja\_Wyjaśnienie\_Blokada\_Zastrzezenie\_Różnica

##### **^ Blokada\_Karty**

- Akcja\_Potwierdzenie\_Blokady

##### **^ Potwierdzenie\_Blokada**

- Akcja\_Blokada\_Wykonanie

#### **[Blokada\_Karta]**

##### **^ Blokada\_Karty**

- Akcja\_Powód\_Blokady

##### **^ Karta\_Kradzież**

- Akcja\_Zalecenie\_Zastrzezenie\_Karty

##### **^ Zgoda**

- Akcja\_Potwierdzenie\_Zastrzezenia

##### **^ Potwierdzenie\_Zastrzezenie**

- Akcja\_Zastrzezenie\_Wykonanie

Niżej umieszczona tabela prezentuje frazy w kategoriach umieszczonych w scenariuszach **[Zgubienie\_Karta]** oraz **[Blokada\_Karta]**:

<b>Nazwa Kategorii</b>	<b>Treningowe zdania dla kategorii</b>
<i>Zgubienie_Karty</i>	<ol style="list-style-type: none"><li>1. Zgubiłem kartę co mam zrobić?</li><li>2. Co w sytuacji kiedy zgubiłem kartę?</li><li>3. Zgubiłem swoją kartę, co teraz?</li><li>4. Nie mam pojęcia gdzie jest moja karta, co w takiej sytuacji?</li><li>5. Proszę pomóż mi, chyba zgubiłem kartę.</li><li>6. ...</li></ol>
<i>Różnica_Blokada_Zastrzezenie</i>	<ol style="list-style-type: none"><li>1. Czy się różni blokada i zastrzezenie?</li><li>2. Jaka jest różnica pomiędzy blokadą i zastrzezeniem.</li></ol>



	<ol style="list-style-type: none"> <li>3. Wyjaśnij mi na czym polega różnica w zablokowaniu i zastrzeżeniu.</li> <li>4. A co to za różnica zastrzec a zablokować?</li> <li>5. A to jest jakaś różnica w blokadzie i zastrzeżeniu karty?</li> <li>6. ...</li> </ol>
<i>Blokada_Karty</i>	<ol style="list-style-type: none"> <li>1. Zablokuj mi kartę</li> <li>2. Zablokuj moją kartę</li> <li>3. Proszę szybko zablokuj moją kartę</li> <li>4. Musisz natychmiast zablokować mi kartę!</li> <li>5. Zblokuj moją kartę.</li> <li>6. ...</li> </ol>
<i>Potwierdzenie_Blokada</i>	<ol style="list-style-type: none"> <li>1. Potwierdzam blokadę karty</li> <li>2. Tak, zablokuj kartę</li> <li>3. Zgadzam się, zablokuj kartę</li> <li>4. Oczywiście, że potwierdzam blokadę karty</li> <li>5. Jasne, potwierdzam. Blokuj kartę</li> <li>6. ...</li> </ol>
<i>Karta_Kradzież</i>	<ol style="list-style-type: none"> <li>1. Ukradli mi kartę</li> <li>2. Ktoś ukradł moją kartę</li> <li>3. Ktoś skradł mi kartę!</li> <li>4. Moja karta została skradziona</li> <li>5. Ktoś musiał ukraść mi kartę!</li> <li>6. ...</li> </ol>
<i>Zgoda</i>	<ol style="list-style-type: none"> <li>1. Zgoda</li> <li>2. Tak</li> <li>3. Okej</li> <li>4. Tak, zgadzam się</li> <li>5. Oczywiście!</li> <li>6. ...</li> </ol>
<i>Potwierdzenie_Zastrzezenie</i>	<ol style="list-style-type: none"> <li>1. Potwierdzam zastrzeżenie karty</li> <li>2. Tak, zastrzeż kartę</li> <li>3. Zgadzam się, zastrzeż kartę</li> <li>4. Oczywiście, że potwierdzam zastrzeżenie karty</li> <li>5. Jasne, potwierdzam. Zastrzeż kartę</li> <li>6. ...</li> </ol>

Tabela 2: Frazy dotyczące kategorii dla scenariuszy [Zgubienie\_Karta], [Blokada\_Karta]

W scenariuszach **[Zgubienie\_Karta]** oraz **[Blokada\_Karta]** występuje kategoria *Blokada\_Karty*, ale w różnych miejscach. Pomimo tego, że kategoria ta ma to samo znaczenie w obu rozmowach, to w zależności od tego jak zacznie się rozmowa, bot musi różnie odpowiedzieć na frazę z kategorii *Blokada\_karty*:

- W scenariuszu [Zgubienie\_Karta] mamy odpowiedź *Akcja\_Potwierdzenie\_Blokady*,
- W scenariuszu [Blokada\_Karta] mamy odpowiedź *Akcja\_Powód\_Blokady*.

W scenariuszu **[Blokada\_Karta]** bot nie może poprosić o potwierdzenie blokady takiej karty jak w scenariuszu **[Zgubienie\_Karta]**, ponieważ wymagane jest podanie przez użytkownika powodu chęci blokady karty. Z procedur występujących w wielu bankach wynika, że jeżeli karta została

skradziona, to zalecane jest zastrzeżenie karty ze względów na bezpieczeństwo danych. Jeżeli bot by postąpił jak w scenariuszu **[Zgubienie\_Karta]** byłoby to niezgodne z procedurami i mogłoby się skończyć poważnymi konsekwencjami jak utrata pieniędzy klienta banku, a dla banku mogłoby to oznaczać duży spadek zaufania i utratę reputacji na co instytucje finansowe nie mogą sobie pozwolić. Zapętlenie bota polega na tym, że zmieni on kontekst rozmowy w przypadku, gdy użytkownik wpisze frazę z kategorii *Blokada\_Karty*, jeżeli użytkownik zacznie rozmowę od kategorii *Zgubienie\_Karty*:

#### **Przykład 6.**

```
^ Zgubienie_Karty
  - Akcja_Blokowanie_Lub_Zastrzezenie
^ Różnica_Blokada_Zastrzezenie
  - Akcja_Wyjaśnienie_Blokada_Zastrzezenie_Różnica
^ Blokada_Karty
  - Akcja_Powód_Blokady
```

Powód zamiaru zablokowania karty jest podany na początku rozmowy – jest to zgubienie karty przez klienta. Twórcy bota muszą wiedzieć, że nie wpadnie on w zły kontekst odpowiedzi w przypadku, gdy dana kategoria jest jednocześnie w wielu historyjkach, a w co najmniej jednej z nich jest początkiem. Sprawdzenie odporności bota na zapętlenie rozmowy mogą potwierdzić *automatyczne testy regresji*, czyli testowanie definicji scenariusz tak, jak jest to opisane w rozdziale 2. Jeżeli bot nie będzie odporny na tego typu problemy, to framework testujący naukę scenariuszy przez bota powinien to wykazać. Można takie sytuacje testować również manualnie po treningu, o ile nie jest to zbyt duże obciążenie dla testerów.

### 3.4 Testowanie powrotu do kontekstu po wtrąceniu frazy niezwiązanej z kontekstem

W procesie tworzenia chatbota należy wziąć pod uwagę takie sytuacje, w których bot będzie musiał umieć powrócić do kontekstu rozmowy po wpisaniu zdania, które nie jest zdefiniowane w aktualnie prowadzonym przez bota kontekście rozmowy.

**Definicja 3.** *Wtrąceniem w rozmowę z botem nazywamy takie zdanie, dla którego istnieje kategoria, która dodatkowo posiada odpowiedź bota na zawarte w niej frazy.*

Inaczej mówiąc, wtrącenie to scenariusz, która na ogół będzie złożony z jednej kategorii oraz jednej odpowiedzi bota (odpowiedź może być podzielona na dwie lub więcej akcji). Obsługa wtrąceń jest nieunikniona podczas procesu tworzenia bota, ponieważ osoba rozmawiająca z botem może przejść wyuczony scenariusz od początku do końca, ale w międzyczasie zapytać o dodatkową rzecz, np. o koszty usługi lub procesu. W takiej sytuacji bot musi sobie poradzić z wtrąceniem dodatkowej wypowiedzi, umieć na nią odpowiedzieć i dodatkowo wrócić do kontekstu przerwanej rozmowy. Załóżmy, że mamy w bazie zdefiniowane scenariusze:

#### **[Blokada\_Karta]**

```
^ Blokada_Karty
  - Akcja_Powód_Blokady
^ Karta_Kradzież
  - Akcja_Zalecenie_Zastrzezenie_Karty
^ Zgoda
```

- Akcja\_Potwierdzenie\_Zastrzezenia
- ^ Potwierdzenie\_Zastrzezenie
- Akcja\_Zastrzezenie\_Wykonanie

oraz

#### **[Koszt\_Zastrzezenie]**

- ^ Koszt\_Zastrzezenie
- Akcja\_Informacja\_Koszt\_Zastrzezenie

W przypadku scenariusza **[Blokada\_Karta]**, gdy bot zaleca zastrzezenie karty, klient może się na to od razu zgodzić albo w międzyczasie zapytać ile taka usługa będzie go kosztowała, tak jak w scenariuszu **[Koszt\_Zastrzezenie]**. Po odpowiedzi bota ile kosztuje zastrzezenie karty bot musi powrócić do poprzedniego kontekstu dotyczącego akcji zastrzegania karty. Aby sprawdzić czy bot potrafi sobie poradzić z sytuacjami, w których klient wtrąca coś do danego scenariusza można zastosować automatyczne lub manualne. Przykład takich testów są przedstawione niżej:

#### **[@Test\_1\_Blokada\_Karta& Koszt\_Zastrzezenie]**

- |                                       |  |
|---------------------------------------|--|
| ^ Blokada_Karty                       | (prośba o zastrzezenie)                  |
| - Akcja_Powód_Blokady                 | (pytanie bota o powód)                   |
| ^ Karta_Kradzież                      | (informacja o kradzieży karty)           |
| - Akcja_Zalecenie_Zastrzezenie_Karty  | (sugestia zastrzezenia karty)            |
| ^ Koszt_Zastrzezenie                  | (pytanie o koszt zastrzezenia)           |
| - Akcja_Informacja_Koszt_Zastrzezenie | (podanie kosztu zastrzezenia)            |
| ^ Zgoda                               | (klient wyraża zgodę)                    |
| - Akcja_Potwierdzenie_Zastrzezenia    | (bota pyta o potwierdzenie zastrzezenia) |
| ^ Potwierdzenie_Zastrzezenie          | (klient potwierdza zastrzezenie)         |
| - Akcja_Zastrzezenie_Wykonanie        | (potwierdzenie zastrzezenia karty)       |

#### **[@Test\_2\_Blokada\_Karta& Koszt\_Zastrzezenie]**

- ^ Blokada\_Karty
- Akcja\_Powód\_Blokady
- ^ Koszt\_Zastrzezenie
- Akcja\_Informacja\_Koszt\_Zastrzezenie
- ^ Karta\_Kradzież
- Akcja\_Zalecenie\_Zastrzezenie\_Karty
- ^ Zgoda
- Akcja\_Potwierdzenie\_Zastrzezenia
- ^ Potwierdzenie\_Zastrzezenie
- Akcja\_Zastrzezenie\_Wykonanie

#### **[@Test\_3\_Blokada\_Karta& Koszt\_Zastrzezenie]**

- ^ Blokada\_Karty
- Akcja\_Powód\_Blokady
- ^ Karta\_Kradzież

- Akcja\_Zalecenie\_Zastrzezenie\_Karty
- ^ Zgoda
  - Akcja\_Potwierdzenie\_Zastrzezenia
- ^ Koszt\_Zastrzezenie
  - Akcja\_Informacja\_Koszt\_Zastrzezenie
- ^ Potwierdzenie\_Zastrzezenie
  - Akcja\_Zastrzezenie\_Wykonanie

#### [@Test\_4\_Blokada\_Karta& Koszt\_Zastrzezenie]

- ^ Blokada\_Karty
  - Akcja\_Powod\_Blokady
- ^ Karta\_Kradziez
  - Akcja\_Zalecenie\_Zastrzezenie\_Karty
- ^ Zgoda
  - Akcja\_Potwierdzenie\_Zastrzezenia
- ^ Potwierdzenie\_Zastrzezenie
  - Akcja\_Zastrzezenie\_Wykonanie
- ^ Koszt\_Zastrzezenie
  - Akcja\_Informacja\_Koszt\_Zastrzezenie

W rozdziale 3.2 opisane są testy dotyczące zachowania bota w przypadku zrozumienia kontekstu rozmowy. Różnica pomiędzy takim brakiem zrozumienia a wtrąceniem jest taka, że w przypadku braku zrozumienia kontekstu użytkownik wpisze frazę w nieodpowiedniej kolejności, a kategoria dla tej frazy nie jest początkiem żadnego scenariusza (nie ma dla niej żadnego, nawet krótkiego na zasadzie pytanie - odpowiedź). Obsługa wtrąceń ma taką zaletę, że nie musimy wpisywać wielu wariantów scenariuszy do zbioru treningowego bota (co zwiększa czas treningu), wystarczy tylko jeden. W przypadku, gdy bot potrafi zrealizować kilka usług i można zapytać o koszt każdej usługi, to liczba wymaganych do wprowadzenia scenariuszy byłaby ogromna. Wtrącenia eliminują ten problem, co znacznie optymalizuje proces wytwarzania bota.

### 3.5 Wyłapanie sprzeczności w definicji scenariuszy rozmów

Podczas definiowania scenariuszy dla chatbota może dojść do sytuacji, w której osoby wprowadzające te scenariusze spowodują niezamierzone występowanie sprzeczności w przebiegu rozmowy z botem. Aby to wyjaśnić, wprowadzona zostanie definicja:

**Definicja 3.** Załóżmy, że istnieją dwa scenariusze mające wspólny początkowy ciąg zdarzeń. Sprzecznością w zbiorze scenariuszy bota nazywamy taką sytuację, w której na pewien wspólny ciąg kategorii i akcji zdefiniowane są dwie różne odpowiedzi (różne względem bazy akcji bota).

Aby ułatwić zrozumienie wyżej przedstawionej definicji, podamy kilka przykładów sprzeczności w scenariuszach:

**Przykład 2.** Załóżmy, że mamy zdefiniowane dwa scenariusze dotyczące ubezpieczenia samochodu, analogicznie jak w poprzednim rozdziale:

**[OfertaUbezpieczenia\_BezWypdku\_1]**

^ Ubezpieczenie\_Samochodu\_ZapytanieKlienta

- Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta

^ Ubezpieczenie\_OC

- Akcja\_Marka\_Samochod

^ Pojazd\_Marka

- Akcja\_Pojazd\_RokProdukcji

^ Akcja\_Pojazd\_RokProdukcji

- Akcja\_Prawojazdy\_CzasPosiadania

^ ...

oraz

**[OfertaUbezpieczenia\_BezWypdku\_2]**

^ Ubezpieczenie\_Samochodu\_ZapytanieKlienta

- Akcja\_Marka\_Samochod

^ Ubezpieczenie\_OC

- Akcja\_Ubezpieczenie\_Samochodu\_ZapytanieKlienta

^ Pojazd\_Marka

- Akcja\_Pojazd\_RokProdukcji

^ Akcja\_Pojazd\_RokProdukcji

- Akcja\_Prawojazdy\_CzasPosiadania

^ ...

Tak zdefiniowane scenariusze wprowadzają „chaos” w zbiorze treningowym dla scenariuszy bota, ponieważ wymagamy od bota, żeby odpowiedział jednocześnie dwoma **różnymi** akcjami na tą samą kategorię, która jest początkiem kontekstu rozmowy. Tego rodzaju sprzeczności mogą powstać na wiele sposobów. Dany scenariusz może mieć w pewnym miejscu do wyboru przez użytkownika dwie kategorie połączone operatorem logicznym LUB.

Założmy, że duże litery alfabetu to nazwy kategorii, a małe litery to nazwy akcji. Klika sytuacji, w których występują konflikty prezentuje niżej przedstawiona tabela:

Przykład 2.1	Przykład 2.2	Przykład 2.3	Przykład 2.4	Przykład 2.5
[Scenariusz_1] ^ A U B - a ^ C - b	[Scenariusz_3] ^ B U C - a ^ C - d	[Scenariusz_5] ^ A U B U D - a ^ C - b	[Scenariusz_7] ^ A U B - a ^ C - b	[Scenariusz_9] ^ A U B - a ^ C - b - c - d
[Scenariusz_2] ^ A - b ^ C - c	[Scenariusz_4] ^ B - b ^ C - d	[Scenariusz_6] ^ B - c ^ C - b	[Scenariusz_8] ^ B - a ^ C - b - c	[Scenariusz_9] ^ B - a ^ C - c - d

Tabela 3: Przykłady parami sprzecznych ze sobą scenariuszy

Przedstawione w tabeli przykłady prezentują sprzeczności parami w każdej kolumnie. Twórcy bota powinni wprowadzić zabezpieczenia w procesie wytwarzania bota, które wykryją takie sytuacje, zanim bot zostanie poddany treningowi.

Dobrym rozwiązaniem jest opracowanie frameworku, który z automatu będzie mógł wyłapać takie sytuacje przez rozpoczęciem treningu kontekstu rozmów. Trening bota nie powinien się odbyć, jeśli kod walidujący definicję scenariuszy znajdzie sprzeczność w zbiorze scenariuszy. Testowanie istnienia sprzeczności polega tak naprawdę na napisaniu przykładów sprzeczności w scenariuszach (może być w postaci skryptu lub testów automatycznych), a dokładniej mówiąc jest to testowanie skuteczności wyłapywania sprzeczności przez automat stworzony w tym celu. Jeśli automat testujący zwróci pozytywny wynik testu (co oznacza że istnieje sprzeczność), to znaczy, że bot wyłapie niepoprawnie zdefiniowane parami sprzeczne scenariusze i nie dopuści do treningu. Taka sytuacja, w postaci jasnego raportu, co z czym stanowi konflikt będzie sygnałem do poprawy błędów w scenariuszach. Po wprowadzeniu poprawek i wyeliminowaniu sprzeczności w zbiorze treningowym należy powtórnie poddać bota treningowi. Jeśli okaże się, że podczas poprawek powstały inne sprzeczne scenariusze niż poprzednio, to czynność tę należy powtarzać, aż do wyeliminowania błędów w definicji scenariuszy bota.

### 3.7 Testowanie kontekstu uzależnionego od wartości zmiennych

W procesie definiowania rozmów chatbota, jego twórcy na pewnym poziomie procesu wytwarzania mogą spotkać się z taką sytuacją, że kontekst rozmowy powinien być uzależniony np. od wartości wprowadzonej liczby przez użytkownika. Zmiennymi mogą mieć różny typ wartości: liczby, państwa, imiona, miasta, kwoty, numer telefonu, numer karty itd.

Zmienne można podzielić na dwa typy ze względu na ich pochodzenie:

- a) zmienne, które będą mogły być zaimportowane z baz danych, plików, plików słownikowych,
- b) zmienne, które będą musiały być uzupełnione przez osoby, które wprowadzają dane uczące dla chatbota.

Przykładem zmiennych, które mogą być zaczytane z bibliotek są np. nazwy miast, nazwy państw. Przykładem zmiennych, które powinny być uzupełnione przez osoby definiujące kontekst rozmowy, są wartości, które muszą być dostosowane do procesów biznesowych instytucji, np. liczba rat kredytu.

W rozmowie, w której występują wartości zmienne ważne są trzy podstawowe aspekty:

- bot musi rozpoznać typ zmiennej,
- bot musi potrafić zmienić kontekst rozmowy w zależności o wartości zmiennej, która ma dolne i górne ograniczenie zbioru możliwych do wprowadzenia wartości,
- w przypadku, gdy dana wartość (liczba lub słowo) może być przyporządkowana do więcej niż jednego typu zmiennej, bot również musi umieć poprowadzić rozmowę i wybrać ten typ zmiennej, który jest pożądanym w danym scenariuszu.

Scenariusze uwzględniające wartości zmiennych prezentuje niżej umieszczona tabela:

Treść scenariusza	Klient chce rozłożyć na raty zakup sprzętu w sklepie, a bot pyta na jaki okres czasu klient chciałby uzyskać taki kredyt. Możliwa ilość rat oferowana przez bank to liczby w zakresie 12-36.		
Nazwa scenariusza	Liczba_rat_1	Liczba_rat_2	Liczba_rat_3
Fraza użytkownika	Chcę kredyt na 150 rat	Chcę kredyt na 24 raty	Chcę kredyt na 3 raty
Odpowiedź bota	Wybrana przez Pana ilość rat jest za duża . Proszę wybrać liczbę z zakresu od 12 do 36 włącznie.	Kwota raty w przypadku 24 miesięcy będzie wynosiła 136 PLN. Czy akceptuje Pan taką wartość raty?	Wybrana przez Pana ilość rat jest za mała. Proszę wybrać liczbę z zakresu od 12 do 36 włącznie.

Liczba rat kredytu w scenariuszu Liczba\_rat\_1 może reprezentować wiele typów zmiennych, np.: 3 ostatnie znaki numeru telefonu, kwota do spłaty, numer domu, numer mieszkania, ilość rat, koszty utrzymania gospodarstwa domowego itd. Bot w takim przypadku musi dopasować odpowiedni typ zmiennej do kontekstu.

Testowanie scenariusza z wartościami zmiennych łączy w sobie dwie strategie testów: **podział zbioru wartości wejściowych zmiennych na klasy równoważności oraz testowanie wartości brzegowych**. Liczbę rat kredytu , zgodnie z założeniami, dzieli się na 3 zbiory:

- za mała ilość, liczby mniejsze od 12,
- poprawnie wprowadzona ilość raty, liczby z przedziału 12-36 włącznie,
- za duża ilość rat, liczby większe niż 36.

Nie można też zapomnieć o przetestowaniu zachowania bota w przypadku podania liczb znajdujących się przy granicach przedziałów, czyli: 11,12,36,37.

Przyjmując, że kategoria dotycząca fraz z liczbą rat, które chce klient ma nazwę *Kredyt\_Liczba\_Rat*, a zmienna to *IloscRat*, definiujemy scenariusz dla bota w następujący sposób:

#### **[Kredyt\_Ratalny]**

```

^ Kredyt_Liczba_Rat { IloscRat }
- { Jeśli IloscRat < 12: Akcja_Zanizona_Liczba_Rat,
    Jeśli 12 ≤ IloscRat ≤ 36: Akcja_Oblicz_Kwote_Raty,
    Jeśli IloscRat >36: Akcja_Przekroczona_Liczba_Rat}

```

Wyrażenie *Kredyt\_Liczba\_Rat { IloscRat }* oznacza, że bot wymaga od użytkownika podania liczby rat wraz z wprowadzoną frazą. Akcja zdefiniowana jest na 3 sposoby, w zależności od przedziału wartości zmiennej *IloscRat*.

Framework, który umożliwi testowanie wyżej przedstawionych scenariuszy musi być tak zdefiniowany, aby umożliwił podanie wartości zmiennej oprócz podawania samych kategorii. Takie podejście wymaga rozszerzenia podejścia, które zostało opisane w rozdziale 2. Wymagane będzie tutaj **testowanie mieszane**, czyli takie, które umożliwia podanie samej kategorii albo kategorii razem z frazą zawierającą zmienną. Przykładowe takie testy mogą mieć postać:

### **[@Test\_Kredyt\_Ratalny\_1]**

^ Kredyt\_Liczba\_Rat: [Chcę kredyt na 11 rat]  
- Akcja\_Zanizona\_Liczba\_Rat,

### **[@Test\_Kredyt\_Ratalny\_2]**

^ Kredyt\_Liczba\_Rat: [Chcę kredyt na 12 rat]  
- Akcja\_Oblicz\_Kwote\_Raty,

### **[@Test\_Kredyt\_Ratalny\_3]**

^ Kredyt\_Liczba\_Rat: [Chcę kredyt na 36 rat]  
- Akcja\_Oblicz\_Kwote\_Raty

### **[@Test\_Kredyt\_Ratalny\_4]**

^ Kredyt\_Liczba\_Rat: [Chcę kredyt na 37 rat]  
- Akcja\_Przekroczona\_Liczba\_Rat,

gdzie wyrażenie *Kredyt\_Liczba\_Rat: [Chcę kredyt na 11 rat]* będzie oznaczało podanie wejściowej kategorii wraz z frazą. Mieszanie kategorii wraz z frazą, która zawiera zmienną, umożliwi przetestowanie każdego wariantu kontekstu bota w zależności od wartości zmiennej, co więcej umożliwi również przetestowanie warunków brzegowych.

## Podsumowanie

Celem napisania artykułu było zaprezentowanie podstawowych problemów związanych z testowaniem kontekstu rozmowy prowadzonej przez bota. Wymienione problemy nie wyczerpują wszystkich sytuacji, z którymi muszą sobie poradzić twórcy bota. Nie mniej jednak problemy te są bardzo ważne, a bez ich weryfikacji oraz obsługi żaden bot nie będzie wykazywał dobrej jakości, która umożliwi wdrożenie go na produkcję. W przypadku dużej ilości scenariuszy w danych treningowych bota, o wiele bardziej pożądane jest testowanie automatyczne niż manualne. Każdy trening jest nieprzewidywalny, co sprawia, że opisane w artykule rodzaje testów powinny być wykonywane cyklicznie po każdym treningu.

## Referencje

[1] <https://marcinkordowski.com/co-sprawia-ze-bot-jest-tak-inteligentny/>

[2] <http://testerzy.pl/baza-wiedzy/podstawy-strategii-testowania-jezyka-chatbota>

Marek Żukowicz jest absolwentem matematyki na Uniwersytecie Rzeszowskim. Jest testerem oprogramowania w firmie Ailleron. Jego zainteresowania skupiają się wokół testowania, matematyki, AI, zastosowania modeli matematycznych w procesie testowania.



Izabela Czech studiuje zarządzanie w Wyższej Szkole Zarządzania i Bankowości w Krakowie, specjalność reklama i grafika komputerowa. Jest testerką oprogramowania w firmie Ailleron. Jej zainteresowania dotyczą testowania oprogramowania, biografii znanych osób oraz tematów związanych z II wojną światową.