

10.12.2024

System pytań i odpowiedzi z użyciem Prolog i Python

Michał Bąk
Jakub Burek
Adrian Chołody

Abstrakt

Celem projektu było stworzenie systemu (chatbota), który odpowiada na pytania w języku naturalnym, korzystając z bazy relacji napisanej w Prologu. Wykorzystano Pythona jako główny język programowania oraz do przetwarzania zapytań. oraz bibliotekę pyswip do integracji z Prologiem oraz do przetwarzania pytań.

Wstęp

Cel

Celem projektu było zaprojektowanie i wdrożenie chatbota, który odpowiada na pytania odnośnie relacji / pokrewieństwa między ludźmi. Chatbot umożliwia użytkownikowi wprowadzanie tekstu w języku naturalnym. System demonstruje połączenie logiki programowania z nowoczesnymi technologiami przetwarzania języka.

Zakres

Projekt obejmował:

- Stworzenie bazy relacji w Prologu, zawierającej informacje o rodzinnych zależnościach
- Zaprojektowanie wzorców pytań w języku naturalnym i ich mapowanie na zapytania Prolog
- Integracje Prologu z Pythonem przy użyciu biblioteki pyswip
- Testowanie systemu na różnych pytaniach dotyczących bazy relacji

Metodyka

Metodologia pracy

Analiza problemu i zaprojektowanie bazy relacji oraz iteracyjne budowanie systemu, testowanie i poprawa.

Narzędzia

Prolog (SWI-Prolog) do logiki i wnioskowania.

Python z biblioteka pyswip do pytań i obsługi użytkownika jak i do integracji obu języków.

Proces

Planowanie → Implementacja → Testowanie → Wnioski i poprawki.

I. Część teoretyczna

Technologie i teoria

Logika Prolog

Prolog opiera się na logice pierwszego rzędu, gdzie wiedza jest reprezentowana w formie faktów (np. **male(adam).**), reguł (np. **siblings(X, Y) :- parents(X, Z, W), parents(Y, Z, W), X \= Y.**), oraz zapytań logicznych (np. **cousins(X, Y).**). Prolog używa mechanizmu unifikacji do dopasowywania wzorców, szukając odpowiedzi na zapytania poprzez wnioskowanie oparte na regułach i faktach w bazie danych.

Przetwarzanie języka naturalnego

Pytania w języku naturalnym są mapowane na zapytania logiczne za pomocą wyrażeń regularnych w Pythonie. Kluczowe elementy pytania (np. imiona i relacje) są dopasowywane do wzorców, takich jak **r"who is the father of (\w+)"**, które generują zapytania w formacie Prolog, np. **father(stefan, X)**. Dzięki temu możliwe jest przetwarzanie pytań w zrozumiałej dla użytkownika formie i przekazywanie ich do Prologu.

II. Część praktyczna

Implementacja

Projektowanie architektury

Zaplanowano infrastrukturę obejmującą:

- **Prolog** do zarządzania bazą relacji logicznych, takich jak rodzice, rodzeństwo czy kuzyni.
- **Python** do przetwarzania pytań w języku naturalnym i komunikacji z Prologiem za pomocą biblioteki pyswip.

Tworzenie bazy relacji w Prologu

- Zdefiniowano fakty, np. **male(adam).**, **parents(stefan, staszek, maria).**
- Opracowano reguły, np. **siblings(X, Y)**, **cousins(X, Y)**, które pozwalają na wyciąganie logicznych wniosków z bazy danych.
- Baza relacji opisuje zależności rodzinne i umożliwia obsługę złożonych pytań.

Tworzenie wzorców pytań w Pythonie

- Zdefiniowano wzorce, które mapują pytania w języku naturalnym na zapytania Prolog
- Użyto wyrażeń regularnych, aby dopasowywać kluczowe elementy pytań, takie jak imiona i typ relacji.

Integracja z Prologiem za pomocą pyswip

- Python za pomocą **pyswip** ładował bazę danych Prologu i wysyłał do niej zapytania.
- Funkcja **prolog.query()** pozwalała dynamicznie uzyskiwać wyniki na podstawie zapytań wygenerowanych z języka naturalnego.

Wyniki i testy

- System przetestowano na wielu przykładowych pytaniach.
- Wyniki były poprawne w większości przypadków, co potwierdziło skuteczność integracji Pythona i Prologu.
- System wykazał zdolność do obsługi różnych typów relacji, takich jak rodzice, kuzyni i rodzeństwo, zapewniając użytkownikowi precyzyjne odpowiedzi.

Podsumowanie

Cel projektu został osiągnięty – system poprawnie odpowiada na pytania w języku naturalnym, mapując je na zapytania Prolog i zwracając logiczne odpowiedzi na podstawie bazy relacji.

Integracja Prologu z Pythonem okazała się skuteczna i pozwoliła na połączenie logiki wnioskowania z elastycznością obsługi języka naturalnego.

System jest modułowy i ma potencjał do dalszego rozwoju, np. o bardziej złożone pytania, większą bazę wiedzy lub dodatkowe funkcje, takie jak wsparcie dla różnych domen tematycznych. Największym wyzwaniem było precyzyjne dopasowanie wzorców pytań w Pythonie, szczególnie dla bardziej skomplikowanych zapytań logicznych. Takie problemy można rozwiązać, ulepszając reguły w Prologu i wzorce w Pythonie.

Bibliografia

1. Prolog -
<https://www.openbookproject.net/py4fun/prolog/prolog1.html>
2. Pyswip
<https://github.com/yuce/pyswip>
3. Film instruktażowy: jak połączyć python z prologiem
<https://www.youtube.com/watch?v=1jwAHiz8WXc>