RASA Kütüphanesi**

RASA Kütüphanesi

Chat Botlarında Metni Anlama ve Karar Verme Süreçleri

1. Metni Anlama (Understanding Text)

a. Natural Language Understanding (NLU)

NLU, metnin anlamını çıkarma sürecidir. NLU, aşağıdaki bileşenlerden oluşur:

- **Intent Recognition:** Kullanıcının ne yapmak istediğini anlama (örneğin, rezervasyon yapmak, bilgi almak).
- **Entity Recognition:** Metinde geçen önemli varlıkları tanımlama (örneğin, tarih, saat, isim).
- **Context Understanding:** Metnin bağlamını anlama ve önceki konuşmaları dikkate alarak anlam çıkarmak.

b. Rule-based Approaches (Kural Tabanlı Yaklaşımlar)

Kural tabanlı yaklaşımlar, önceden tanımlanmış kurallar kullanarak metni anlamaya çalışır:

- **Pattern Matching:** Belirli anahtar kelimeleri veya ifadeleri tanıyarak cevap verme. Örneğin, "hava durumu" kelimesi geçerse hava durumu bilgisi verme.
- **Decision Trees:** Karar ağaçları kullanarak çeşitli koşullara göre hareket etme.
- **If-Then Rules:** Basit if-then mantığı ile karar verme.

c. Neural Approaches (Sinirsel Yaklaşımlar)

Sinirsel yaklaşımlar, derin öğrenme tekniklerini kullanarak metnin anlamını çıkarır:

- **Word Embeddings:** Kelimeleri vektörlere dönüştürerek anlamlarını temsil etme (örneğin, Word2Vec, GloVe).
- **Transformer Models:** BERT, GPT gibi modeller, metinleri daha derin ve bağlama duyarlı olarak anlama kapasitesine sahiptir.
- **Recurrent Neural Networks (RNN):** Özellikle sıralı verileri (örneğin, cümleler) anlamada kullanılır.

2. Ne Yapacağına Karar Verme (Deciding What to Do Next)

a. Rule-based Approaches (Kural Tabanlı Yaklaşımlar)

Kural tabanlı yaklaşımlar, sabit kurallara dayalıdır ve belirli girdilere karşılık belirli çıktılar üretir:

- **Scripted Responses:** Belirli anahtar kelimelere veya ifadelere sabit cevaplar verir.
- **Flowcharts:** Diyalog akışını yönlendiren önceden tanımlanmış akış şemaları kullanır.
- **State Machines:** Farklı durumlar arasında geçiş yaparak karar verir. Her duruma belirli kurallar atanmıştır.

b. Neural Approaches (Sinirsel Yaklaşımlar)

Sinirsel yaklaşımlar, öğrenme ve adaptasyon yetenekleriyle ne yapacağına karar verir:

- **Reinforcement Learning:** Kullanıcının tepkilerini analiz ederek öğrenir ve performansını zamanla iyileştirir.
- **Sequence-to-Sequence Models:** Girdi metnini anlamak ve buna göre çıktı metni üretmek için kullanılır.
- **Attention Mechanisms:** Metnin hangi kısmının daha önemli olduğunu belirleyerek daha doğru kararlar alır.

Örnek Bir Çalışma Senaryosu:

1. **Metin Girişi:** Kullanıcı "Bugün hava nasıl?" diye sorar.
2. **NLU:**
    - **Intent Recognition:** Kullanıcı hava durumu hakkında bilgi almak istiyor.
    - **Entity Recognition:** Bugün (tarih varlığı).
3. **Rule-based Approach:**
    - Kural: "hava durumu" ifadesi geçerse hava durumu API'sini çağır ve sonucu kullanıcıya göster.
4. **Neural Approach:**
    - Transformer modeli, "Bugün hava nasıl?" sorusunu analiz eder ve bağlamı dikkate alarak hava durumu bilgisi API'sine sorgu yapar.
5. **Cevap Üretimi:** "Bugün hava güneşli ve 25 derece" cevabını üretir ve kullanıcıya sunar.

Bu yaklaşımlar, chat botlarının metni anlama ve uygun cevabı verme süreçlerinde nasıl çalıştığını göstermektedir. Rule-based yaklaşımlar daha basit ve hızlıdır, ancak esneklikleri sınırlıdır. Neural yaklaşımlar daha karmaşık ve esnek olup, öğrenme yetenekleri sayesinde daha doğru ve bağlama duyarlı cevaplar verebilirler.

RASA Domain Dosyaları

Giriş

RASA, açık kaynaklı bir sohbet botu framework'üdür ve doğal dil işleme (NLP) yetenekleri ile kullanıcılarla etkileşim kurar. RASA'nın yapı taşlarından biri olan domain dosyaları, botun genel yapılandırmasını ve işleyişini tanımlar. Bu çalışmada, RASA'da domain dosyalarının ne işe yaradığı, nasıl kullanıldığı ve bileşenlerinin detayları incelenecektir.

Domain Dosyasının Rolü

Domain dosyası, bir RASA chatbot'unun "beyni" olarak kabul edilir. Botun işleyişini belirleyen birçok bileşeni içerir:

- **Intents (Niyetler):** Kullanıcı niyetlerini belirtir.
- **Entities (Varlıklar):** Kullanıcı mesajlarından çıkarılan önemli bilgileri belirtir.
- **Slots (Yuva Alanları):** Diyalog sırasında depolanan bilgileri temsil eder.
- **Responses (Yanıtlar):** Botun kullanıcılara verdiği cevapları tanımlar.
- **Actions (Eylemler):** Botun gerçekleştirdiği görevleri veya verdiği cevapları belirtir.
- **Forms (Formlar):** Bilgi toplamak için kullanılan çok adımlı diyalogları tanımlar.
- **Session Config (Oturum Yapılandırması):** Kullanıcı oturumlarının nasıl yönetileceğini belirler.
-

Domain Dosyasının Yapısı

Domain dosyası genellikle YAML formatında yazılır ve aşağıdaki ana bölümleri içerir:

Intents (Niyetler)

Kullanıcıların bot ile etkileşime girerken belirttikleri niyetleri tanımlar. Örneğin:

```
intents:
  - greet
  - goodbye
  - request_weather
  - book_flight
```

Entities (Varlıklar)

Kullanıcı mesajlarından çıkarılması gereken önemli bilgileri belirtir. Örneğin:

```
entities:
  - location
  - time
  - flight_number
```

Slots (Yuva Alanları)

Diyalog sırasında toplanan bilgileri depolamak için kullanılır. Slotlar, entitilerden, kullanıcı mesajlarından veya özel mantıklardan doldurulabilir. Örneğin:

```
slots:
  location:
    type: text
  time:
    type: text
  flight_number:
    type: text
```

Responses (Yanıtlar)

Botun kullanıcılara verdiği hazır cevapları tanımlar. Örneğin:

```
responses:
  utter_greet:
    - text: "Hello! How can I help you today?"
  utter_goodbye:
    - text: "Goodbye! Have a great day!"
  utter_ask_location:
    - text: "Which location are you interested in?"
```

Actions (Eylemler)

Botun gerçekleştirebileceği görevleri veya verebileceği cevapları belirtir. RASA, özel eylemleri Python kodu ile de destekler. Örneğin:

```
actions:
  - action_check_weather
  - action_book_flight
```

Forms (Formlar)

Bilgi toplamak için kullanılan çok adımlı diyalogları tanımlar. Örneğin:

```yaml
forms:
  flight_booking_form:
    required_slots:
      departure_city:
        - type: from_entity
          entity: location
      arrival_city:
        - type: from_entity
          entity: location
      travel_date:
        - type: from_entity
          entity: time
```

Session Config (Oturum Yapılandırması)

Kullanıcı oturumlarının nasıl yönetileceğini belirler. Örneğin:

```yaml
session_config:
  session_expiration_time: 60
  carry_over_slots_to_new_session: true
```

Örnek Domain Dosyası

Aşağıda, basit bir domain dosyasının örneğini bulabilirsiniz:

```yaml
version: "3.1"

intents:
  - greet
  - goodbye
  - affirm
  - deny
  - mood_great
  - mood_unhappy
  - bot_challenge

responses:
  utter_greet:
    - text: "Hey! How are you?"

  utter_cheer_up:
    - text: "Here is something to cheer you up:"
      image: "https://i.imgur.com/nGF1K8f.jpg"

  utter_did_that_help:
    - text: "Did that help you?"
```

```
  utter_happy:
    - text: "Great, carry on!"

  utter_goodbye:
    - text: "Bye"

  utter_iamabot:
    - text: "I am a bot, powered by Rasa."

session_config:
  session_expiration_time: 60
  carry_over_slots_to_new_session: true
```

Bu domain dosyası, botun temel işlevlerini ve kullanıcı ile olan etkileşimini tanımlar. Kullanıcıdan gelen mesajlara göre botun nasıl yanıt vereceğini belirler.

Sonuç

RASA'da domain dosyası, chatbot'un genel yapısını ve işleyişini belirleyen ana bileşendir. Bu dosya, botun kullanıcı niyetlerini anlama, gerekli bilgileri çıkarma, yanıt verme ve gerekli eylemleri gerçekleştirme süreçlerini tanımlar. Doğru yapılandırılmış bir domain dosyası, botun kullanıcı etkileşimlerinde başarılı olmasını sağlar.