

BÁC CÁO MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Đề tài : Xây dựng chatbot hỗ trợ sinh viên ICTU sử dụng RASA Framework

Người thực hiện : Nguyễn Thái Học

04 - 2024

Nội dung

- 1 Tổng quan về đề tài
- 2 Tổng quan về RASA framework

1. Tổng quan về đề tài

1.1 Giới thiệu

Bài toán sử dụng RASA framework của Python nhằm xây dựng ra một hệ thống chatbot có khả năng giao tiếp xử lý và hiểu ngôn ngữ của người dùng

1.2 Mục tiêu của đề tài

- Xây dựng được một chatbot giúp sinh viên giải đáp các thông tin, các vấn đề liên quan đến phạm vi trong trường ICTU
- Giúp cho mọi người có thể tự xây dựng một chatbot một cách dễ dàng phục vụ cho cá nhân, doanh nghiệp mà không cần phải có quá nhiều những kiến thức chuyên sâu về Natural Language Processing.

2. Tổng quan về RASA framework

2.1 Giới thiệu về RASA

- RASA là một thư viện mã nguồn mở để tự động hóa các cuộc hội thoại dựa trên văn bản và giọng nói. Chúng được sử dụng để xây dựng các chatbot AI bằng cách sử dụng xử lý ngôn ngữ tự nhiên (NLU) và Python.
- RASA cung cấp một không gian có tên là “Framework” để phát triển các chatbot xây dựng AI bằng NLU. Thông thường khi xây dựng bất kỳ một chatbot nào chúng ta đều phải xử lý:
 - ▶ Xác định ý định câu hội thoại người dùng (intent classification).
 - ▶ Xác định các thực thể hội thoại (entity extraction).
 - ▶ Xác định hành động cần thực hiện ứng với ý định câu hội thoại hiện tại của người dùng (câu trả lời).
- Để xây dựng được đầy đủ các chức năng nêu trên, cần phải xây dựng một quy trình như thu thập dữ liệu, chuẩn bị dữ liệu, huấn luyện mô hình phân loại, trích xuất thực thể. Tuy nhiên RASA đã cung cấp đầy đủ những yếu tố trên để có thể dễ dàng triển khai.

2. Tổng quan về RASA framework

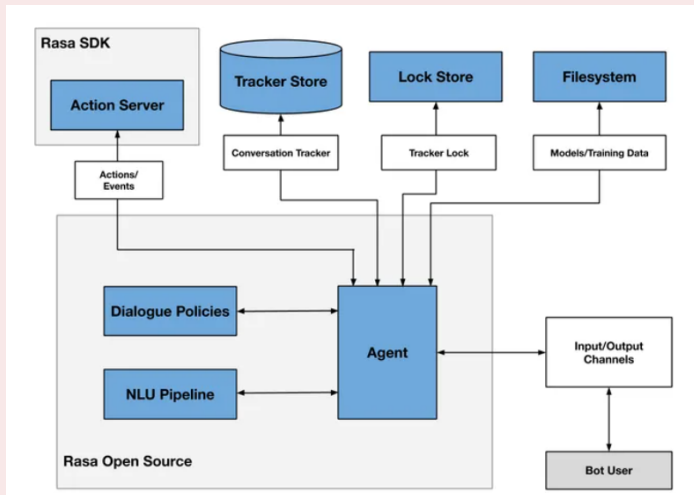
2.2 Các thành phần của RASA

- RASA NLU (Natural Language Processing) là một module đóng vai trò xử lý câu hỏi, câu hội thoại của người dùng, xác định ý định người dùng, trích chọn thực thể. RASA NLU sử dụng rất nhiều tính năng để có thể làm được những công việc trên như: tokenization, regular expression, synonym, lookup table.
- RASA Core: Các thông tin sau khi được xử lý bởi RASA NLU sẽ được chuyển tới RASA core. Tại đây RASA core quản lý xử lý các bối cảnh của cuộc hội thoại, quản lý phiên và phản hồi. RASA core bao gồm các chính sách để xử lý luồng hội thoại bao gồm:
 - ▶ RulePolicy: Sử dụng các luật để xác định hành động tiếp theo.
 - ▶ MemoizationPolicy: Sử dụng các story để xác định hành động tiếp theo.
 - ▶ TEDPolicy: Sử dụng học sâu để xác định hành động tiếp theo.

Ba Policy này hợp lại với nhau tạo thành Dialog Policies.

2. Tổng quan về RASA framework

2.2 Các thành phần của RASA



Hình 1: Các thành phần hệ thống của RASA

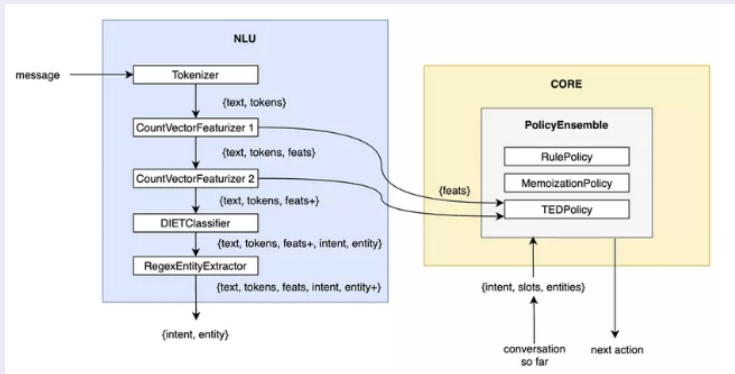
2. Tổng quan về RASA framework

2.2 Các thành phần của RASA

- Song song với hai thành phần cốt lõi trên, hệ thống chatbot RASA cần thêm các thành phần để xử lý bộ nhớ, kết nối các thành phần với nhau để hoạt động hiệu quả.
 - ▶ NLU pipeline: Là các module của RASA NLU.
 - ▶ Dialog Policies: Là các policy của RASA core.
 - ▶ Action Server: Các action đáp lại người dùng được viết bằng Python.
 - ▶ Tracker Store: Module lưu trữ các slot, entity, câu hỏi thoại (có thể coi đây là bộ nhớ lưu trữ của chatbot).
 - ▶ Lock Store: Module đảm bảo các câu hỏi thoại gửi đến chatbot được xử lý tuần tự.
 - ▶ Filesystem: Module lưu trữ và quản lý các tệp của chatbot.
 - ▶ Agent: Module xử lý chung và kết nối các thành phần khác. Trong các thành phần trên, trong quá trình xây dựng chatbot, thông thường chủ yếu ta quan tâm tới NLU pipeline, Dialog Policies, action server và sử dụng tracker để xử lý dữ liệu trong action server.

2. Tổng quan về RASA framework

2.3 Luồng hoạt động



Hình 2: Quy trình hoạt động của RASA NLU pipeline

2. Tổng quan về RASA framework

2.3 DIET Classifier

DIET classifier là một mô hình đa tác vụ có thể phân loại ý định và trích xuất thực thể, mô hình này có kết quả vượt trội so với BERT và huấn luyện nhanh hơn. Các vector đặc trưng từ bước trước được đưa vào mô hình, mô hình dự đoán đầu ra là các ý định của tin nhắn hội thoại và thực thể có trong tin nhắn. Quá trình huấn luyện, mô hình DIET thực hiện đồng thời ba tác vụ: dự đoán intent, trích chọn thực thể, dự đoán masker token (tức là một từ mà mô hình trong quá trình huấn luyện phải cố gắng đoán được, giúp cho khoảng cách các từ gần nhau hơn trên không gian biểu diễn số khi mô hình hội tụ). Hàm mất mát của mô hình sẽ bằng tổng ba hàm mất mát tương ứng với ba nhiệm vụ:

$$L_{total} = L_i + L_e + L_m \quad (1)$$

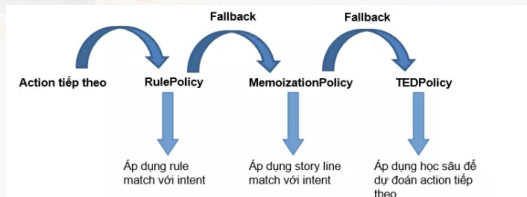
Trong đó L_i , L_e , L_m lần lượt là hàm mất mát của việc dự đoán intent, thực thể, và masker token.

2. Tổng quan về RASA framework

2.4 RASA Core pipeline

RASA Core xác định action nào được thực hiện để đáp lại câu hỏi thoại từ người dùng, action có thể là một câu thoại văn bản. RASA Core chia luồng hội thoại dựa trên 3 chính sách đã nhắc đến ở phần trên. Tại đây RASA sẽ lựa chọn các hành động đáp lại người dùng theo thứ tự:

- Áp dụng RulePolicy nếu có Rule thỏa mãn ý định (Intent) vừa được xác định.
- Nếu không có Rule thỏa mãn Intent vừa được xác định, áp dụng Story nếu có Story thỏa mãn ý định (Intent) đó.
- Nếu không có cả Rule và Story thỏa mãn ý định (Intent) vừa được xác định, sử dụng mô hình học sâu để lựa chọn hành động có khả năng nhất.



Hình 3: RASA core pipeline
BÁC CÁO MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Thank You!