

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG THÁI NGUYÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

**Đề tài: “Xây dựng chatbot hỗ trợ sinh viên trường Đại học CNTT & TT
sử dụng RASA framework”**

Giảng viên hướng dẫn : T.S Trần Văn Khánh
Sinh viên thực hiện : Nguyễn Thái Học
Lớp : CNTT K19 CLC

Thái Nguyên, năm 2024

MỤC LỤC

DANH SÁCH HÌNH VẼ	2
LỜI NÓI ĐẦU	3
1 TỔNG QUAN VỀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN VÀ RASA FRAMEWORK	4
1.1 Tổng quan về xử lý ngôn ngữ tự nhiên	4
1.1.1 Khái niệm	4
1.1.2 Ứng dụng của xử lý ngôn ngữ tự nhiên	5
1.1.3 Cách thức hoạt động	7
1.1.4 Các kỹ thuật xử lý ngôn ngữ tự nhiên	9
1.1.5 Thách thức của mô hình NLP	12
1.2 Cở sở lý thuyết về RASA framework	13
1.2.1 Khái niệm	13
1.2.2 Các thành phần của RASA	13
1.2.3 Luồng hoạt động	15
2 ỨNG DỤNG RASA FRAMEWORK XÂY DỰNG CHATBOT	18
2.1 Ngôn ngữ và công cụ	18
2.1.1 Ngôn ngữ	18
2.1.2 Công cụ	18
2.1.3 Cài đặt RASA Framework	19
2.2 Xây dựng chatbot	21
2.2.1 Xây dựng intent	21
2.2.2 Xây dựng utter	22
2.2.3 Xây dựng stories	22

MỤC LỤC

2.2.4	Huấn luyện và thử nghiệm mô hình	23
KẾT LUẬN		24
TÀI LIỆU THAM KHẢO		25
PHỤ LỤC CODE		26
NHẬN XÉT CỦA GIẢNG VIÊN		27

DANH SÁCH HÌNH VẼ

1.1	ChatGPT một ứng dụng nổi bật của xử lý ngôn ngữ tự nhiên	4
1.2	Phân tích tình cảm sử dụng NLP	5
1.3	NLP trong nhận dạng thực thể	6
1.4	Truy xuất thông tin sử dụng NLP	7
1.5	Kỹ thuật tokenization xử lý văn bản	8
1.6	Bag of Words	8
1.7	TF-IDF	9
1.8	Thuật toán Logistic Regression	10
1.9	Thuật toán Decsion Tree	10
1.10	Mạng Covolutional Neural Netwok	11
1.11	Mạng thần kinh Recurrent Neural Network	11
1.12	Phương pháp Auto Encoder	12
1.13	Kiến trúc transformer	12
1.14	Các thành phần hệ thống của RASA	14
1.15	Quy trình hoạt động của RASA NLU pipeline	15
1.16	Quy trình hoạt động RASA Core	17
2.1	Giao diện VS Code	18
2.2	Cài đặt extensions	19
2.3	Cài đặt RASA Framework thông qua trình quản lý gói PIP	20
2.4	Kiểm tra version của RASA	20
2.5	Cấu trúc thư mục của RASA	21
2.6	Tạo ra các thực thể	22
2.7	Xây dựng các utter	22

2.8	Tạo dựng kịch bản	23
2.9	Thử nghiệm chabot	23

LỜI NÓI ĐẦU

Xử lý ngôn ngữ tự nhiên (NLP) vài năm trở lại đây đang là một trong những lĩnh vực hấp dẫn nhất của trí tuệ nhân tạo (AI). Lĩnh vực này đang mang đến một cuộc cách mạng về khả năng hiểu ngôn ngữ của con người, ngôn ngữ lập trình và thậm chí cả các chuỗi sinh học và hoá học, chẳng hạn như cấu trúc DNA và protein. Các mô hình AI mới nhất còn có khả năng phân tích cảm xúc, làm thơ, sáng tác nhạc, vẽ tranh...

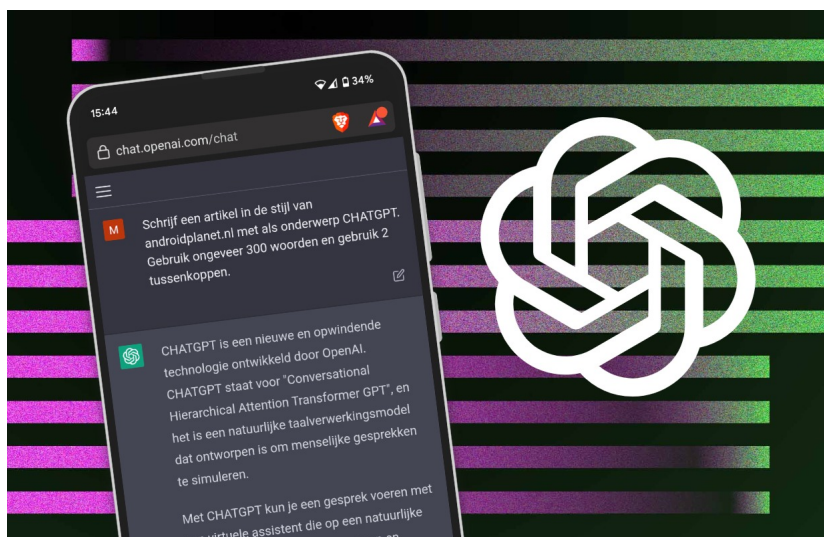
Nhận thấy tầm quan trọng cũng như ứng dụng của NLP trong xã hội hiện tại cùng với mong muốn tạo ra một sản phẩm có ích có thể giúp đỡ các bạn sinh viên trong trường nên em đã chọn "Xây dựng chatbot hỗ trợ sinh viên trường Đại học CNTT & TT sử dụng RASA framework" là đề tài để nghiên cứu kết thúc học phần môn học này.

CHƯƠNG 1: TỔNG QUAN VỀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN VÀ RASA FRAMEWORK

1.1 Tổng quan về xử lý ngôn ngữ tự nhiên

1.1.1 Khái niệm

Xử lý ngôn ngữ tự nhiên (NLP - Natural Language Processing) là một lĩnh vực khoa học máy tính và trí tuệ nhân tạo nhằm mục đích làm cho máy tính hiểu được ngôn ngữ của con người. NLP sử dụng ngôn ngữ học để tính toán, nghiên cứu về cách hoạt động của ngôn ngữ và các mô hình khác nhau dựa trên số liệu thống kê, học máy và học sâu. Những công nghệ này cho phép máy tính phân tích và xử lý dữ liệu văn bản hoặc giọng nói, đồng thời nắm bắt được ý nghĩa đầy đủ của chúng, bao gồm cả ý định và cảm xúc của người nói hoặc người viết.



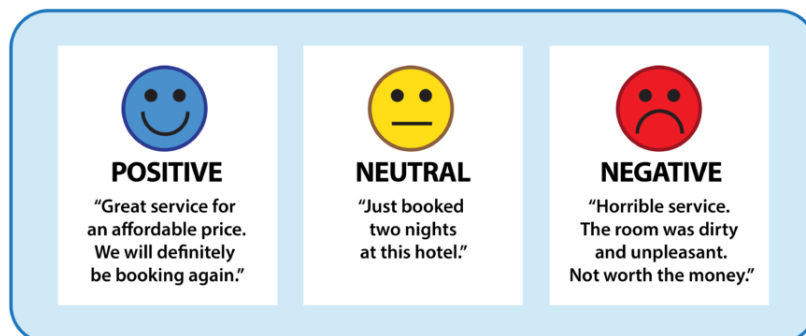
Hình 1.1: ChatGPT một ứng dụng nổi bật của xử lý ngôn ngữ tự nhiên

1.1.2 Ứng dụng của xử lý ngôn ngữ tự nhiên

NLP được sử dụng cho nhiều nhiệm vụ liên quan đến ngôn ngữ, bao gồm trả lời các câu hỏi, phân loại văn bản theo nhiều cách khác nhau và trò chuyện với người dùng. Một số tác vụ có thể ứng dụng NLP để giải quyết:

- Phân tích tình cảm là quá trình phân loại cảm xúc của văn bản. Nối chung đầu vào của mô hình phân loại cảm xúc là một đoạn văn bản và đầu ra là xác suất mà cảm xúc được thể hiện là tích cực, tiêu cực hoặc trung tính. Thông thường, xác suất này dựa trên các tính năng của mô hình học sâu để nắm bắt các mối liên quan phụ thuộc giữa các từ trong văn bản. Phân tích tình cảm được sử dụng để phân loại đánh giá cảm xúc khách hàng trên nhiều nền tảng trực tuyến khác nhau cũng như cho các ứng dụng thích hợp liên quan đến phân loại cảm xúc.

SENTIMENT ANALYSIS



Given text, sentiment analysis classifies its emotional quality.

Hình 1.2: Phân tích tình cảm sử dụng NLP

- Dịch máy là tự động dịch giữa hai ngôn ngữ khác nhau. Đầu vào của mô hình như vậy là những văn bản cần được dịch và đầu ra là văn bản đã được dịch sang một ngôn ngữ mong muốn khác. Google Translate là một ứng dụng điển hình. Các phương pháp dịch máy hiệu quả còn có thể phân biệt các từ có nghĩa và tương đồng nhau. Một số hệ thống còn thực hiện nhận dạng ngôn ngữ nhằm phân loại văn bản theo ngôn ngữ này hay ngôn ngữ khác.

- Nhận dạng thực thể được đặt tên nhằm mục đích trích xuất các thực thể trong một đoạn văn bản thành các danh mục được xác định trước như tên cá nhân, tổ chức, địa điểm và số lượng. Đầu vào của loại mô hình này là văn bản còn đầu ra là các thực thể có cùng vị trí bắt đầu và kết thúc nhưng khác tên. Nhận dạng thực thể rất hữu ích trong các ứng dụng như tóm tắt bài báo và phòng chống những văn bản có thông tin sai lệch.

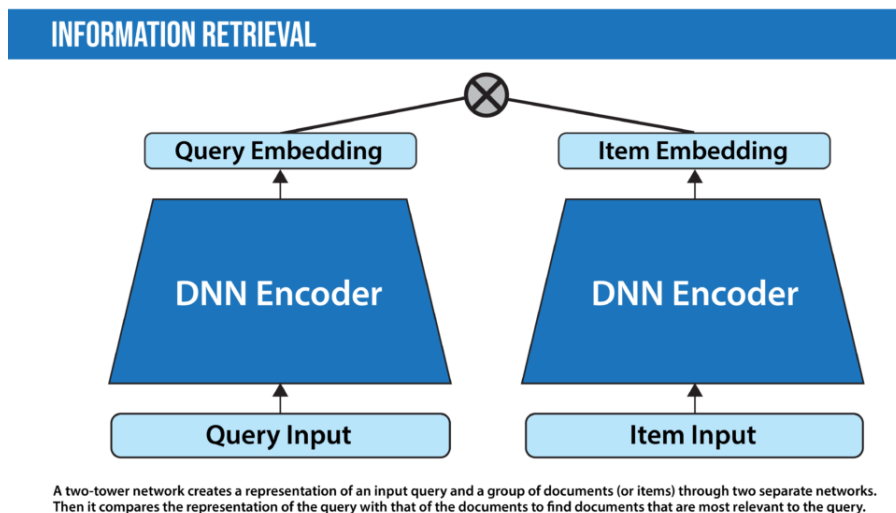
NAMED ENTITY RECOGNITION (NER) TAGGING



spaCy named entity recognition tagging of the first paragraph of Andrew Ng's Wikipedia page. "NORP" stands for nationalities or religious or political groups. Note that spaCy incorrectly labels "AI" as "GPE," for geopolitical entity.

Hình 1.3: NLP trong nhận dạng thực thể

- Các mô hình sử dụng cú pháp mã hoá các quy tắc ngữ pháp để sửa ngữ pháp trong văn bản. Trong đó đầu vào là dữ liệu gồm các câu không đúng ngữ pháp và đầu ra là dữ liệu có các câu đúng ngữ pháp.
- Tạo văn bản hay còn gọi là tạo ngôn ngữ tự nhiên là tạo ra văn bản tương tự như con người viết. Những mô hình như vậy có thể được tinh chỉnh để tạo ra văn bản ở các thể loại và định dạng khác nhau. Việc tạo ra văn bản được thực hiện bằng cách sử dụng các mô hình như: LSTM, BERT, GPT-4, LaMDA và các mô hình khác. Chúng rất hữu ích cho tính năng tự động hoàn thành đoạn văn và xây dựng những chatbot.
- Truy xuất thông tin tìm thấy các tài liệu phù hợp nhất với truy vấn. Đây là vấn đề mà mọi hệ thống tìm kiếm và đề xuất đều gặp phải. Mục đích không phải là trả lời một truy vấn cụ thể và là truy xuất, từ một tập hợp tài liệu rất lớn. Hệ thống truy xuất tài liệu chủ yếu thực hiện hai quy trình: lập chỉ mục và đối chiếu.



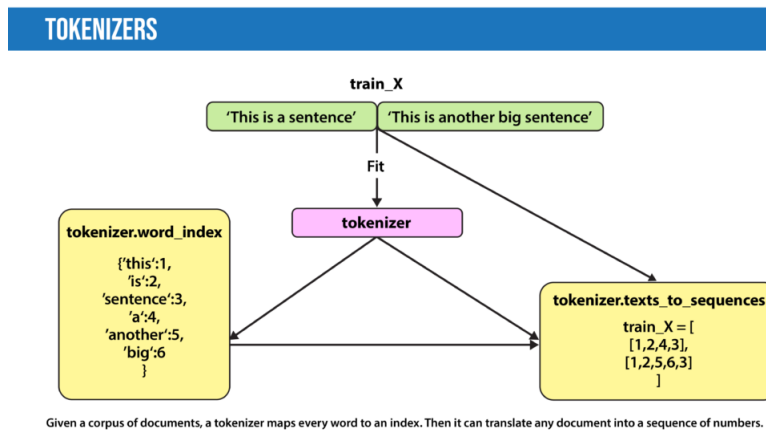
Hình 1.4: Truy xuất thông tin sử dụng NLP

- Tóm tắt văn bản là nhiệm vụ rút ngắn văn bản để làm nổi bật những thông tin, ý chính nhất

1.1.3 Cách thức hoạt động

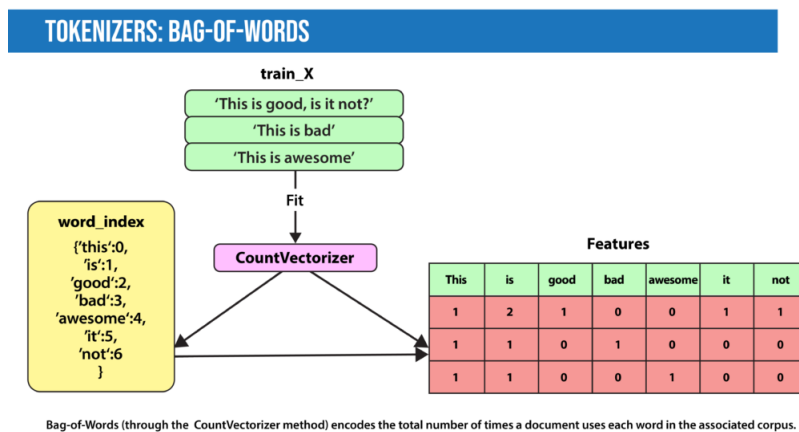
Các mô hình NLP hoạt động bằng cách tìm các mối quan hệ giữa các phần cấu thành của ngôn ngữ, ví dụ: các chữ cái, từ và câu được tìm thấy trong tập dữ liệu văn bản. Kiến trúc NLP sử dụng nhiều phương pháp khác nhau để xử lý trước dữ liệu, trích xuất tính năng và mô hình hoá:

- Xử lý dữ liệu: Trước khi mô hình xử lý văn bản cho một tác vụ cụ thể, văn bản thường được xử lý trước để cải thiện hiệu suất của mô hình hoặc chuyển thành các kí tự mà mô hình có thể hiểu được. Một số kỹ thuật được sử dụng trong quá trình này như: Stemming and lemmatization (từ gốc và từ vựng), sentence segmentation (phân đoạn câu), tokenization. Hình 1.4 là một ví dụ về phương pháp tokenization trong việc xử lý dữ liệu. Phương pháp này chia văn bản thành các từ riêng lẻ và các câu. Kết quả trả về thường bao gồm một chỉ mục và văn bản được mã hoá, chúng giúp cho các mô hình ngôn ngữ có thể bỏ qua các từ, câu không quan trọng.
- Trích xuất đặc trưng: Hầu hết các kỹ thuật học máy thông thường đều hoạt động trên các tính năng, thường là các con số được mã hoá và quy chuẩn từ một tài liệu được tạo ra bởi Bag-of-Words (túi từ), TF-IDF.



Hình 1.5: Kỹ thuật tokenization xử lý văn bản

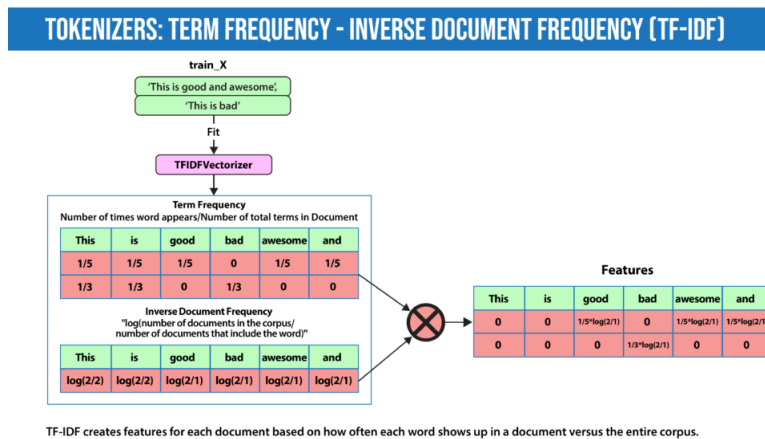
+ Bag-of-Words: Phương pháp này đếm số lần mỗi từ hoặc sự kết hợp của n từ xuất hiện trong văn bản.



Hình 1.6: Bag of Words

+ TF-IDF: Phương pháp này dùng để đánh giá tầm quan trọng của các từ đối với văn bản. Một từ rất quan trọng nếu xuất hiện nhiều lần trong một tài liệu. Tuy nhiên có một số từ mặc dù xuất hiện nhiều những thực tế lại không quan trọng. Nhìn chung phương pháp này giúp trích lọc ra các từ hiếm và thấp nhưng lại có vai trò quan trọng đối với văn bản.

- Word2Vec: Được giới thiệu vào năm 2013, sử dụng mạng thần kinh để cố gắng dự đoán các từ xung quanh cho một từ mục tiêu và ngược lại dự đoán từ mục tiêu dựa vào các từ xung quanh.



Hình 1.7: TF-IDF

1.1.4 Các kỹ thuật xử lý ngôn ngữ tự nhiên

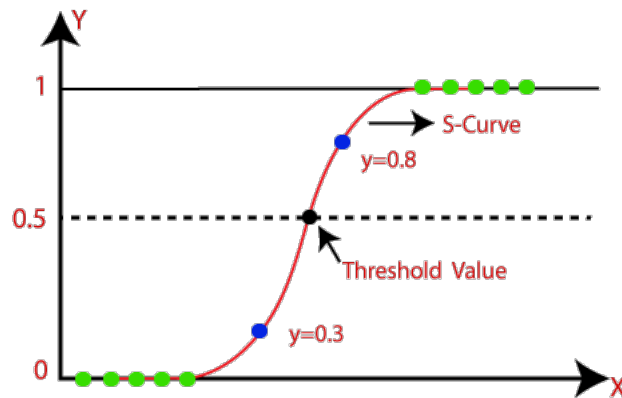
Các bài toán liên quan đến NLP có rất nhiều phương pháp hữu ích để có thể giải quyết. Nhìn chung có thể chia các kỹ thuật này thành hai loại: Phương pháp học máy truyền thống và phương pháp học sâu.

a. Phương pháp học máy truyền thống

+ Logistic Regression: Hồi quy logistic là một thuật toán phân loại có giám sát nhằm mục đích dự đoán xác suất xảy ra một sự kiện dựa trên một số đầu vào. Trong NLP, mô hình hồi quy logistic có thể được áp dụng để giải quyết các vấn đề như phân tích tình cảm, phát hiện thư rác và phân loại độc tính.

Hồi quy logistic dự đoán đầu ra của một biến phân loại phụ thuộc, kết quả phải là một giá trị phân loại hoặc rời rạc, chúng đưa ra các giá trị nằm trong khoảng từ 0 đến 1. Đường cong hình chữ S chính là hàm logistic nhằm dự đoán hai giá trị tối đa (0 hoặc 1). Đường cong từ hàm logistic cho biết khả năng xảy ra điều gì đó như mail có bị spam hay không, văn bản này có thuộc loại này hay là không...

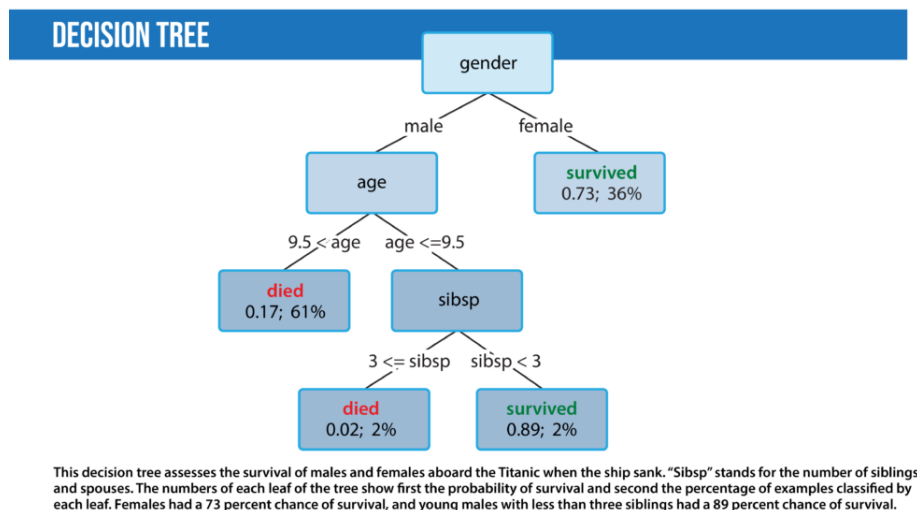
+ Navie Bayes Classification: Thuật toán Navie Bayes là một thuật toán học có giám sát, dựa trên định lý Bayes và được sử dụng để giải các bài toán phân loại. Chúng được sử dụng chủ yếu trong phân loại văn bản bao gồm tập dữ liệu huấn luyện nhiều chiều. Đây là một trong những thuật toán phân loại đơn giản và hiệu quả nhất giúp cho xây dựng các mô hình nhanh và đưa ra dự đoán nhanh chóng. Một số ứng dụng của thuật toán này đó là lọc thư rác,



Hình 1.8: Thuật toán Logistic Regression

phân tích tình cảm và phân loại bài viết.

+ Decision Tree: Cây quyết định là một kỹ thuật học có giám sát, có thể được sử dụng cho cả bài toán phân loại và hồi quy, nhưng chủ yếu được ưa chuộng hơn để giải các bài toán Phân loại. Đây là một trình phân loại có cấu trúc cây, trong đó các nút bên trong biểu thị các tính năng của tập dữ liệu, các nhánh biểu thị các quy tắc quyết định và mỗi nút lá biểu thị kết quả.

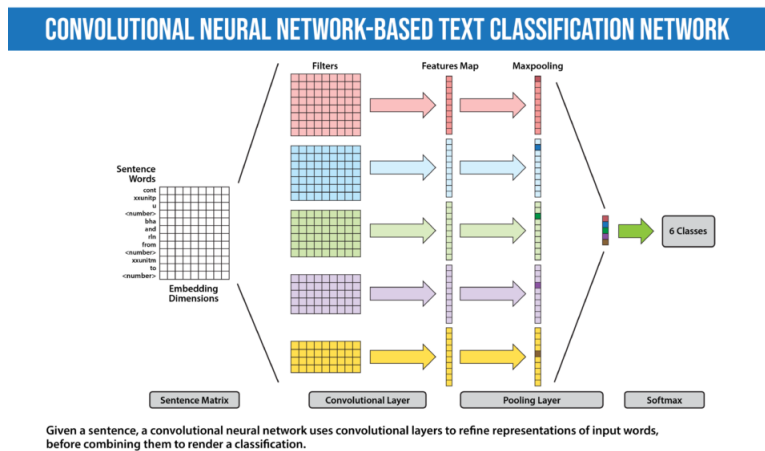


Hình 1.9: Thuật toán Decsion Tree

b. Phương pháp học sâu

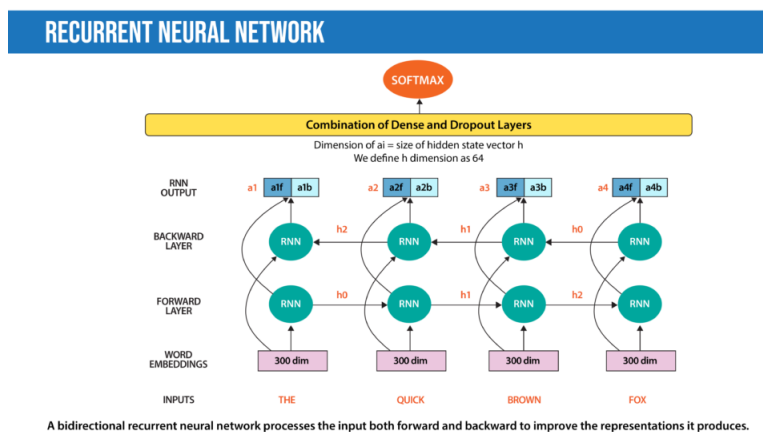
+ Convolutional Neural Network: Ý tưởng sử dụng mạng CNN (Convolutional Neural Network) để phân loại văn bản đã được áp dụng từ rất lâu. Thay vì dữ liệu đầu vào là các

pixel của hình ảnh thì đầu vào là các câu hoặc tài liệu được biểu diễn dưới dạng ma trận các từ.



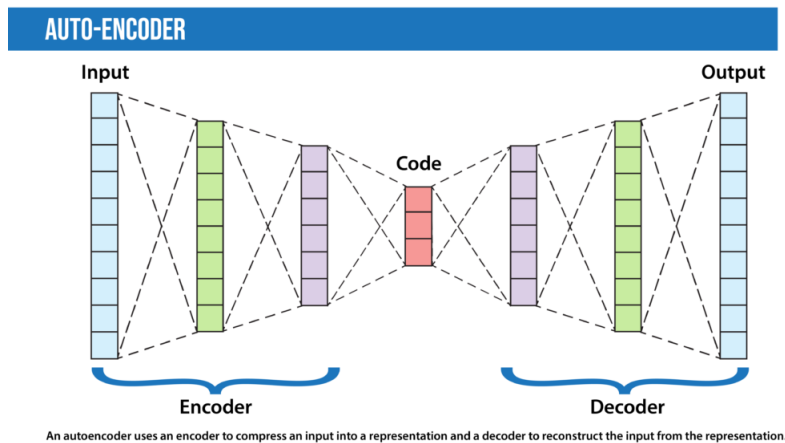
Hình 1.10: Mạng Covolutional Neural Network

+ Recurrent Neural Network: Mạng thần kinh RNN giúp cho mô hình học và ghi nhớ thông tin trước đó tốt hơn. Việc ghi nhớ nhiều thông tin trước đó hay là một từ hoặc cụm từ trong câu trước đó giúp cho việc phân loại văn bản trở nên đa dạng hơn. RNN cũng đã được sử dụng để tạo ra các bằng chứng toán học và dịch suy nghĩ của con người thành lời nói.



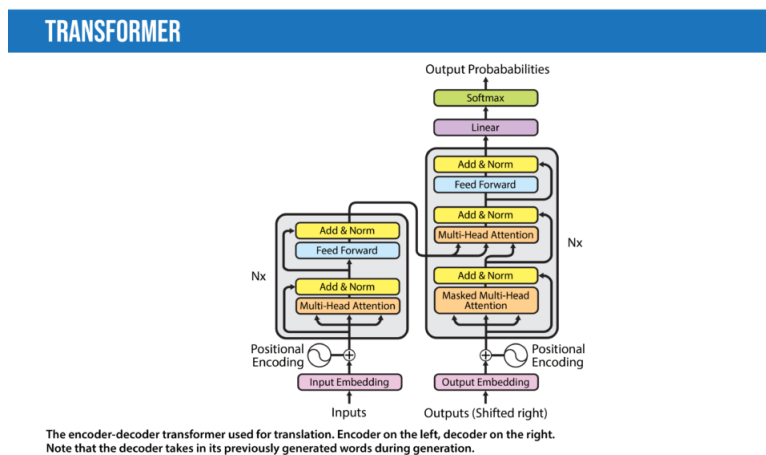
Hình 1.11: Mạng thần kinh Recurrent Neural Network

+ Auto Encoder: Auto Encoder là bộ mã hoá giải có khả năng ánh xạ gần đúng từ này qua từ khác. Khi dữ liệu đầu vào có số chiều quá lớn, phương pháp này vô hình chung được sử dụng để giảm đi kích thước dữ liệu và biểu diễn dữ liệu dưới một dạng vector để phù hợp với các yêu cầu của các mô hình riêng biệt.



Hình 1.12: Phương pháp Auto Encoder

+ Transformer: Transformer là một kiến trúc xuất hiện từ năm 2017. Mô hình này thay vì việc xử lý từng từ một thì xử lý tất cả các từ trong một lúc, làm giảm đi tốc độ đào tạo và chi phí so với RNN.



Hình 1.13: Kiến trúc transformer

1.1.5 Thách thức của mô hình NLP

Các mô hình NLP gặp nhiều thách thức do tính phức tạp và đa dạng của ngôn ngữ tự nhiên. Một số thách thức này bao gồm sự mơ hồ, tính biến đổi, sự phụ thuộc vào ngữ cảnh, ngôn ngữ tượng hình, tính đặc thù của miền, nhiều và thiếu dữ liệu được gắn nhãn.

1.2 Cở sở lý thuyết về RASA framework

1.2.1 Khái niệm

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 kì 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.

1.2.2 Các thành phần của RASA

a. Thành phần cơ bản

RASA có hai thành phần cơ bản chính là RASA NLU và RASA Core.

+ 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:

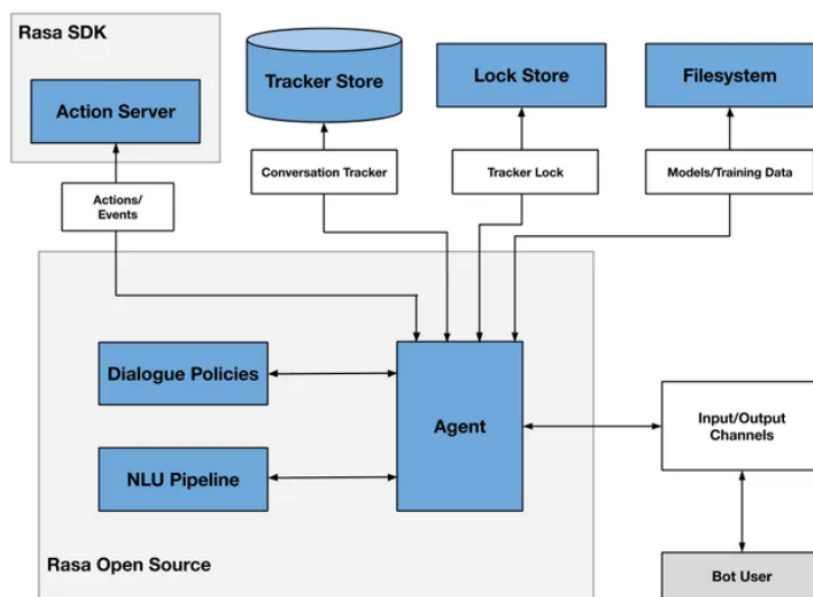
- RulePolicy: Sử dụng các luật để xác định hành động tiếp theo.

- 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.

b. Thành phần hệ thống

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.



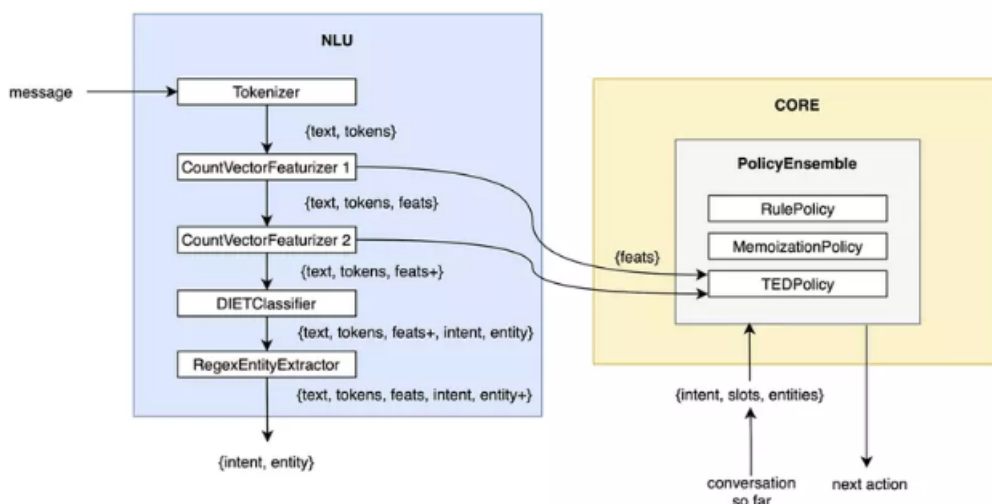
Hình 1.14: Các thành phần hệ thống của RASA

- 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.

1.2.3 Luồng hoạt động

Khi người dùng đưa vào một câu thoại (messenger). Tin nhắn này đi vào RASA NLU, xử lý đầu vào, trích chọn vector đặc trưng, áp dụng các bộ trích chọn thực thể và xác định ý định của người dùng. Sau khi xác định được ý định, trích chọn được các thực thể trong tin nhắn, dữ liệu được đẩy sang RASA Core, dữ liệu các ý định và thực thể từ các tin nhắn cũng sẽ được đưa vào mô hình để xử lý tính toán và hành động tiếp theo để đáp trả lại người dùng.



Hình 1.15: Quy trình hoạt động của RASA NLU pipeline

a. RASA NLU Pipeline

Bao gồm nhiều module để có thể xử lý đầu vào câu văn bản của người dùng: tokenizer, trích chọn đặc trưng (features extraction), trích xuất thực thể (regex entity, extractor,...), bộ phân loại (classifier). Trong hình trên câu văn bản được đưa theo thứ tự:

- Tokenization
- CountVectorFeaturizer1

- CountVectorFeaturizer2
- DIETClassifier
- RegexEntityExtractor

Đây là các thành phần để có thể cấu hình cho module này. Tuy nhiên không nhất thiết là phải tuân theo thứ tự cũng như phải có đầy đủ những thành phần này.

b. 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:

$$L_{total} = L_i + L_e + L_m \quad (1.1)$$

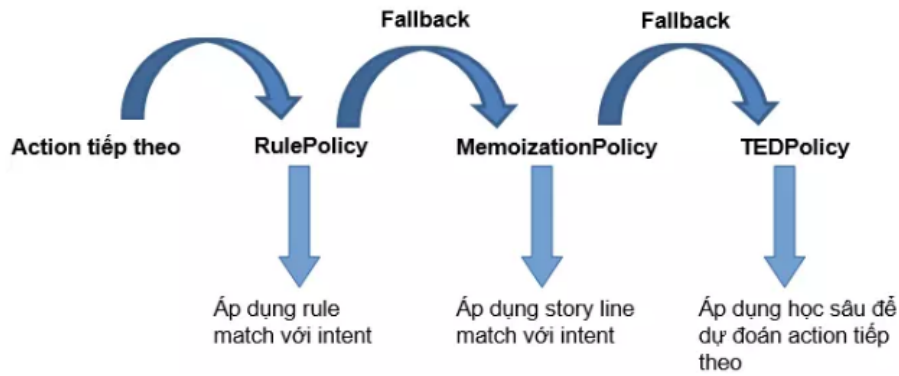
Trong đó L_i , L_e , L_m lần lượt là hàm mất mát của việc dự đoán intent, thực thể, và masker token.

c. 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 1.16: Quy trình hoạt động RASA Core

CHƯƠNG 2: ỨNG DỤNG RASA FRAMEWORK XÂY DỰNG CHATBOT

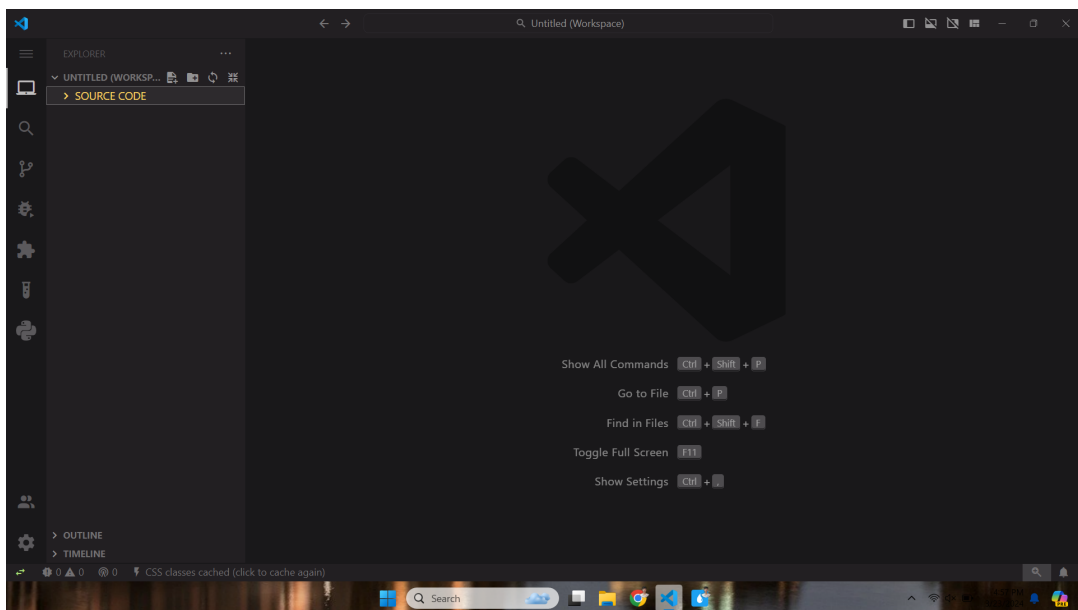
2.1 Ngôn ngữ và công cụ

2.1.1 Ngôn ngữ

Sử dụng Python là ngôn ngữ chính để xây dựng website và hệ thống gợi ý lọc cộng tác. Bên cạnh đó sử dụng thêm HTML, CSS để tạo backend và frontend cho trang web.

2.1.2 Công cụ

Sử dụng VS Code (Visual Studio Code) làm trình biên dịch chính cho chương trình. Download VS Code tại đây 'link download'. Sau khi tải về tiến hành giải nén và cài đặt.



Hình 2.1: Giao diện VS Code

Trong VS Code hỗ trợ một số tiện ích giúp cho việc xây dựng chương trình trở nên nhanh chóng và thuận tiện hơn. Một số tiện ích nên cài đặt như: Code Runner, Python, HTML CSS Support.



Hình 2.2: Cài đặt extensions

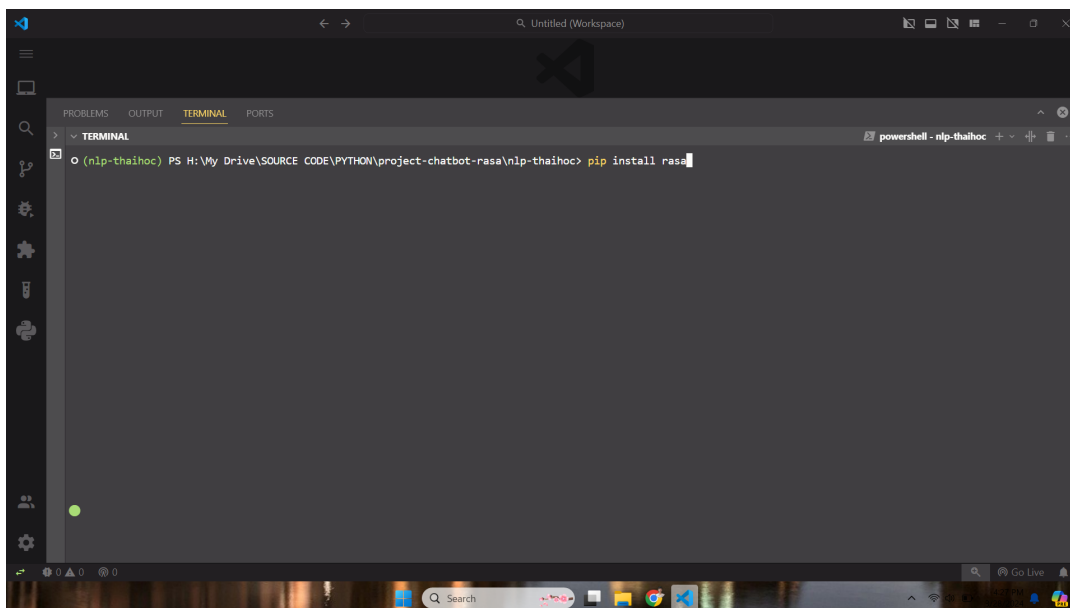
Tải trình biên dịch Python tại: 'link download' làm biến môi trường để biên dịch chương trình python, các file mã nguồn .py.

2.1.3 Cài đặt RASA Framework

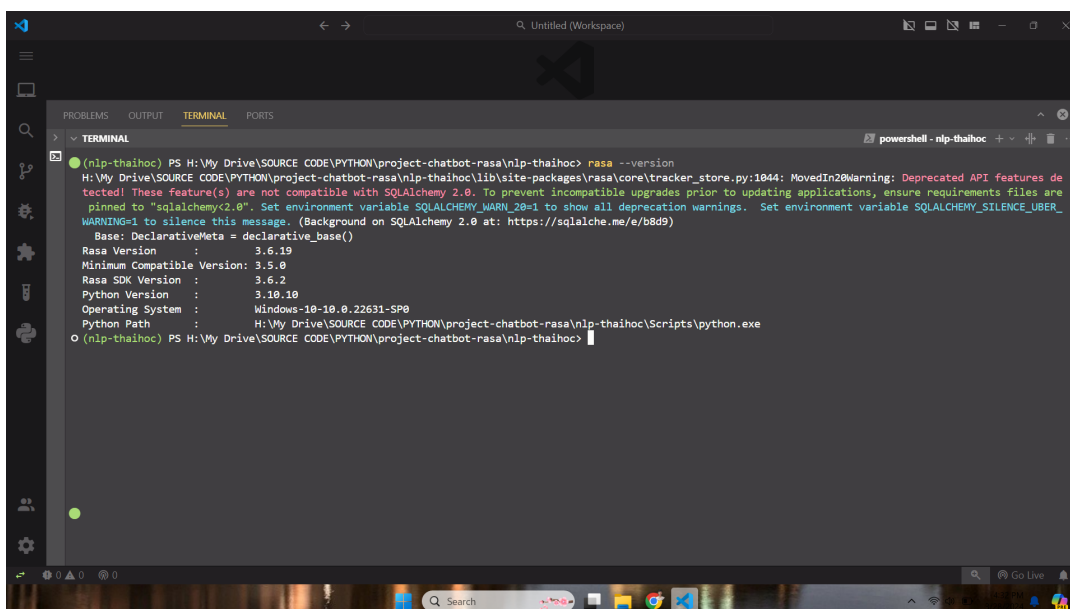
Để bắt đầu xây dựng chương trình cần cài đặt một số thư viện cần thiết. Python hỗ trợ cài đặt thư viện thông qua trình quản lý gói PIP (cho phép cài đặt và quản lý các thư viện tích hợp sẵn trong thư viện tiêu chuẩn). Để cài đặt bất kể một thư viện nào, tại command line (giao diện dòng lệnh). Gõ pip install <tên thư viện cần cài đặt>.

Thông thường nếu cài các thư viện đơn lẻ, việc cài đặt thường sẽ sử dụng như cách trên. Đối với những bài toán cần cài đặt nhiều thư viện như hình trên để rút ngắn thời gian, thì ta cần tạo một file có đuôi .txt, sau đó đưa vào những thư viện cần cài đặt tương ứng với phiên bản muốn cài. Tại command line, gõ pip install -r <tên file .txt chứa các thư viện cần cài đặt>.

Sau khi cài đặt xong, kiểm tra xem RASA đã được cài đặt thành công hay chưa. Nếu thành công màn hình sẽ hiển thị như hình (2.4)



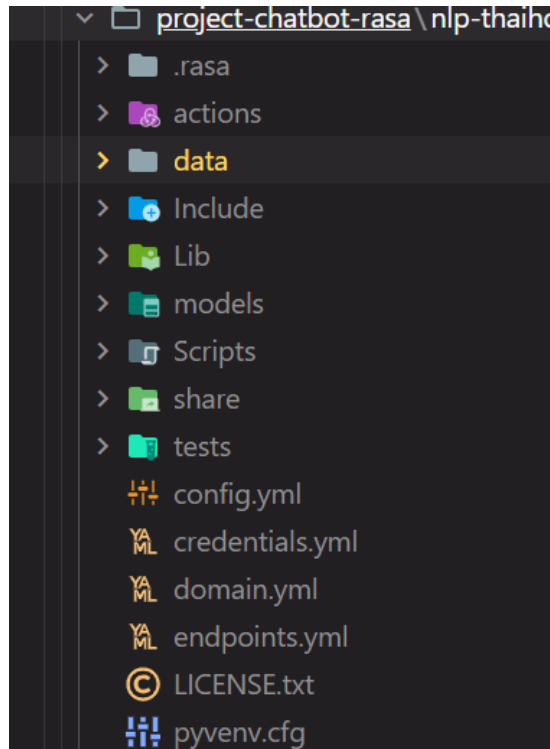
Hình 2.3: Cài đặt RASA Framework thông qua trình quản lý gói PIP



Hình 2.4: Kiểm tra version của RASA

2.2 Xây dựng chabot

Để bắt đầu khởi chạy RASA sử dụng lệnh 'rasa init', khởi chạy toàn bộ cấu hình và cái file liên quan. Sau khi khởi chạy xong cấu trúc thư mục có dạng:



Hình 2.5: Cấu trúc thư mục của RASA

Trong đó ta quan tâm đến bốn file chính:

- domain.yml: file này bao gồm các dữ liệu mà chatbot sẽ trả lời người dùng.
- nlu.yml: Chứa các thực thể ý đồ của người dùng
- stories.yml, rules.yml: Chứa các kịch bản giữa người dùng và chatbot.

2.2.1 Xây dựng intent

Trước khi xây dựng các thực thể hay nói cách khác là tạo ra các câu hỏi hoặc là các tin nhắn mà người dùng có thể đưa vào. Ví dụ ở đây (hình 2.6) là các câu hỏi về năm thành lập của trường, giải thích từ viết tắt ICTU. Chúng ta tạo ra càng nhiều intent (nhiều câu hỏi) thì mô hình sẽ càng có độ chính xác cao.


```
# infomation school
- intent: information_school
  examples: |
    - trường đại học công nghệ thông tin và truyền thông (ICTU) thành lập khi nào
    - ICTU được thành lập vào năm nào
    - trường bạn thành lập được bao nhiêu năm rồi
    - sinh nhật trường là ngày bao nhiêu

# information school (1)
- intent: information_school_1
  examples: |
    - ICTU là gì
    - tại sao lại gọi là ICTU
    - ICTU là viết tắt của từ gì vậy
    - trường ICTU là trường nào
```

Hình 2.6: Tạo ra các thực thể

2.2.2 Xây dựng utter

Tiếp theo tạo ra các utter (lời nói), được sử dụng để chỉ các hành động của chatbot để tạo ra các phản hồi hoặc tin nhắn trả lời cho người dùng. Trong Rasa, các hành động "utter" thường được sử dụng để gửi các câu trả lời hoặc thông điệp cho người dùng. Ví dụ, trong một tình huống nếu người dùng gửi một yêu cầu "Xin chào!" (có trong các thực thể được xây dựng có tên là greet), bot có thể sử dụng hành động "utter_greet" để gửi lại một lời chào như "Chào bạn! Tôi có thể giúp gì cho bạn?".

```
# infomation school
- intent: information_school
  examples: |
    - trường đại học công nghệ thông tin và truyền thông (ICTU) thành lập khi nào
    - ICTU được thành lập vào năm nào
    - trường bạn thành lập được bao nhiêu năm rồi
    - sinh nhật trường là ngày bao nhiêu

# information school (1)
- intent: information_school_1
  examples: |
    - ICTU là gì
    - tại sao lại gọi là ICTU
    - ICTU là viết tắt của từ gì vậy
    - trường ICTU là trường nào
```

Hình 2.7: Xây dựng các utter

2.2.3 Xây dựng stories

Sau khi đã xác định được intent cũng như các utter tương ứng. Ta tiến hành tạo ra các kịch bản mà có thể sẽ xảy ra giữa người dùng và chatbot. Ví dụ ta có ba intent: chào, hỏi về

tên, hỏi về tuổi và ba utter tương ứng để trả lời các intent đó. Ta có thể xây dựng kịch bản như:

- Người dùng chào -> chatbot chào lại -> người dùng hỏi tên -> chatbot đáp lại tên
- Người dùng chào -> chatbot chào lại -> người dùng hỏi tuổi -> chatbot đáp lại tuổi

Tất cả các kịch bản được tạo nằm trong file stories.yml. Trong đó steps biểu diễn trình tự cho đoạn hội thoại. Intent là ý đồ người dùng, utter là lời phản hồi của chatbot.

```
- story: happy path
  steps:
  - intent: greet
  - action: utter_greet
  - intent: happy_communication_1
  - action: utter_happy_communication_1
  - intent: happy_communication_2
  - action: utter_happy_communication_2
```

Hình 2.8: Tạo dựng kịch bản

2.2.4 Huấn luyện và thử nghiệm mô hình

Sau khi đã có được các dữ liệu chuẩn bị từ các bước trên. Tiến hành huấn luyện mô hình bằng lệnh 'rasa train'. Cấu hình của RASA mặc định huấn luyện qua 100 epochs. Ta có thể điều chỉnh các bước pipeline, và số epochs trong file config.yml. Sau khi huấn luyện xong, ta thử nghiệm bằng cách gõ 'rasa shell'.

```
Your input -> Xin chào
Xin chào bạn! Rất vui được gặp bạn ở đây. Tôi là một mô hình ngôn ngữ AI được gọi là ChatbotICTU, được đào tạo bởi kĩ sư AI của trường đại học CNTT & TT Thái Nguyên. Tôi có thể xử lý các câu hỏi và cung cấp các thông tin liên quan về trường đại học CNTT & TT Thái Nguyên
Your input -> hi
Xin chào. Rất vui được gặp bạn ở đây. Tôi có thể giúp gì cho bạn
Your input -> bạn tên là gì
Cứ gọi tôi là ChatbotICTU hay bot nhé
Your input -> bạn sinh ra ở đâu
Có lẽ là tôi sinh ra vào năm 2024, vì tôi được kĩ sư tạo ra vào năm nay
Your input -> 
```

Hình 2.9: Thử nghiệm chatbot

Chatbot đã nhận diện và trả lời khá đúng với các ý đồ, câu hỏi của người dùng. Như vậy ta đã xây dựng thành công một chatbot đơn giản, hỗ trợ trả lời các câu hỏi.

KẾT LUẬN

Bài báo cáo đã nghiên cứu về ứng dụng của RASA framework trong việc hỗ trợ tạo ra chatbot đơn giản. Sau đó đã áp dụng các cơ sở lý thuyết để xây dựng một chatbot xoay quanh việc giải đáp các vấn đề cho trường Đại học CNTT & TT Thái Nguyên. Thông qua quá trình nghiên cứu em đã:

- Hiểu được khái niệm, ứng dụng, cách thức hoạt động của RASA framework
- Xây dựng chatbot sử dụng RASA framework.
- Thử nghiệm thành công chatbot trên máy tính cục bộ.

Mặc dù đã rất nỗ lực, xong với kiến thức và thời gian còn hạn chế nên không thể tránh khỏi những thiếu sót. EM rất mong nhận được sự cảm thông, nhận xét, đánh giá từ các thầy cô để bài của nhóm được hoàn thiện và phát triển hơn nữa.

TÀI LIỆU THAM KHẢO

- [1] <https://medium.com/the-research-nest/using-the-diet-classifier-for-intent-classification-in-dialogue-489c76e62804>
- [2] <https://rasa.com/blog/introducing-dual-intent-and-entity-transformer-diet-state-of-the-art-performance-on-a-lightweight-architecture/>
- [3] <https://rasa.com/blog/intents-entities-understanding-the-rasa-nlu-pipeline/>
- [4] <https://rasa.com/blog/bending-the-ml-pipeline-in-rasa-3-0/>
- [5] <https://rasa.com/blog/dialogue-policies-rasa-2/>
- [6] <https://rasa.com/blog/response-retrieval-models/>

PHỤ LỤC CODE

Link source code: Thai Hoc Nguyen - NLP - Project

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái nguyên, ngày ... tháng ... năm 2024

Giảng viên