



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN HỆ THỐNG THÔNG TIN

KHÓA LUẬN TỐT NGHIỆP



**XÂY DỰNG CHATBOT HỖ TRỢ
HỎI ĐÁP CHO CÁC PHẦN MỀM
KỸ THUẬT TẬP TRUNG VÀO CÂU
HỎI WHAT – HOW**

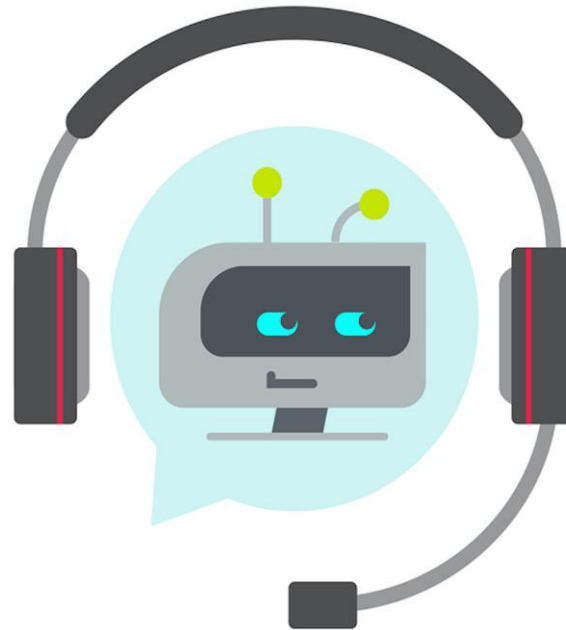
GVHD: TS. PHẠM NGUYỄN CƯỜNG

KHÓA 2015 - 2019





Nhóm thực hiện



❖ **Trần Thị Ly Ly 1512312**

❖ **Trần Hiền Lương 1512308**





Mục lục

1. Tổng quan đề tài

2. Xây dựng mô hình tri thức dựa vào Ontology

3. Xử lí thông tin đầu vào và xây dựng phản hồi với RASA

4. Thực nghiệm – đánh giá

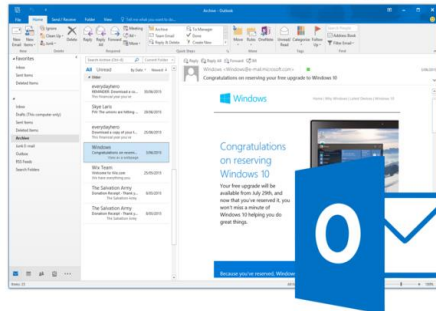
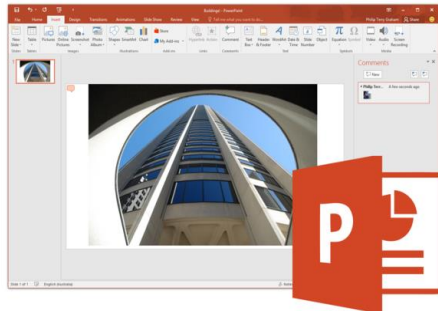
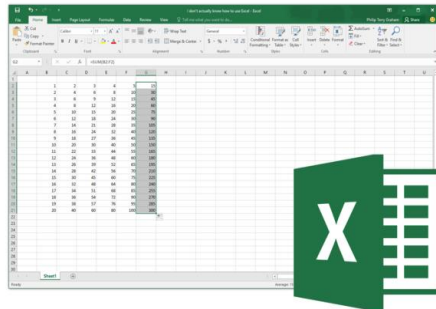
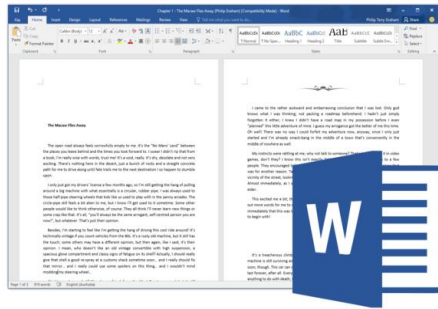
5. Video demo

6. Kết luận và hướng phát triển





1. Tổng quan đề tài



Adobe Photoshop CC



Adobe Illustrator

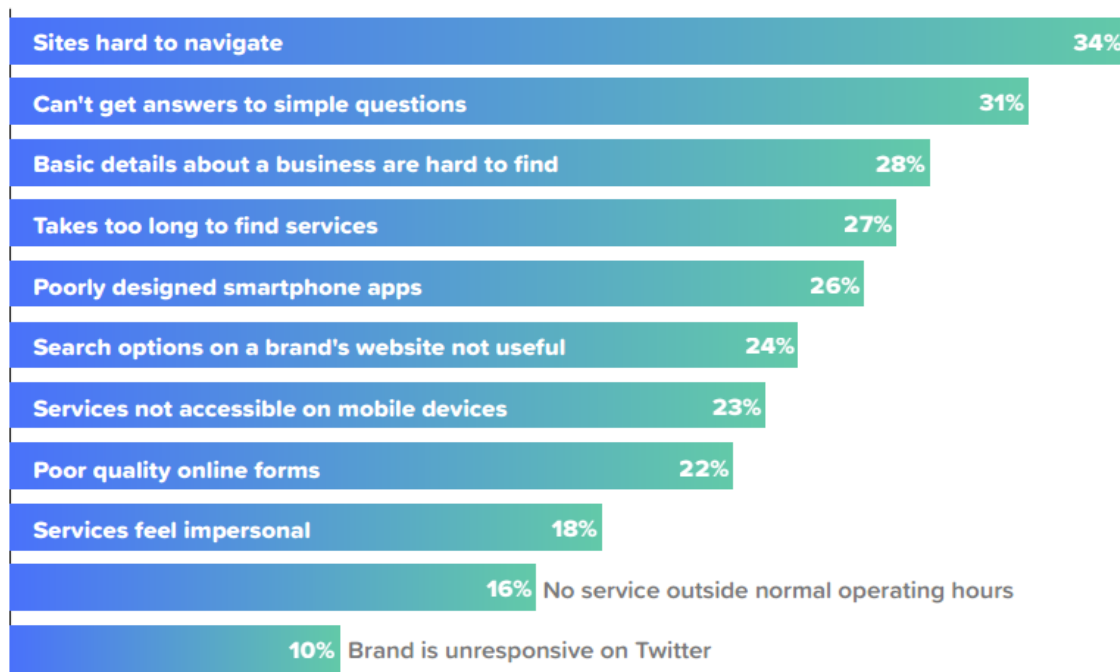




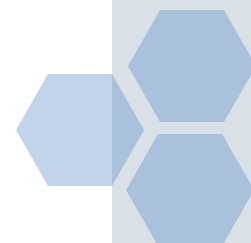
1. Tổng quan đề tài

Problems With Traditional Online Experiences

What frustrations have you experienced in the past month?



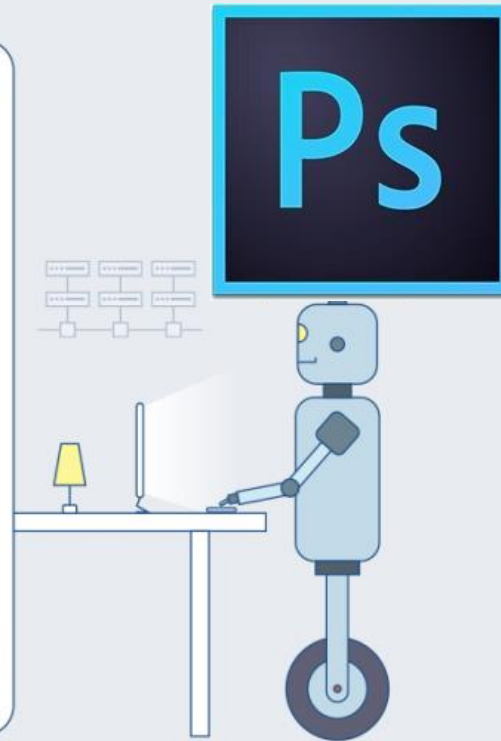
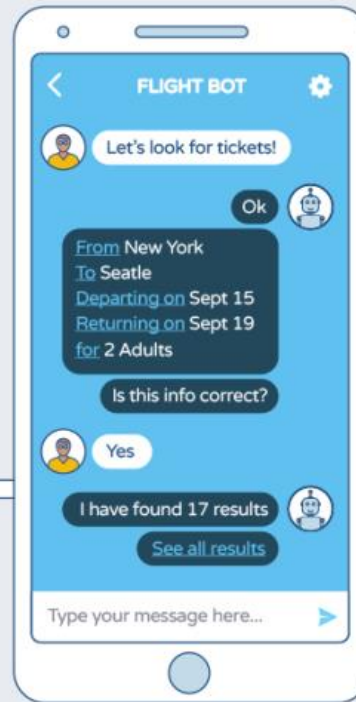
The 2018 State of Chatbots Report.





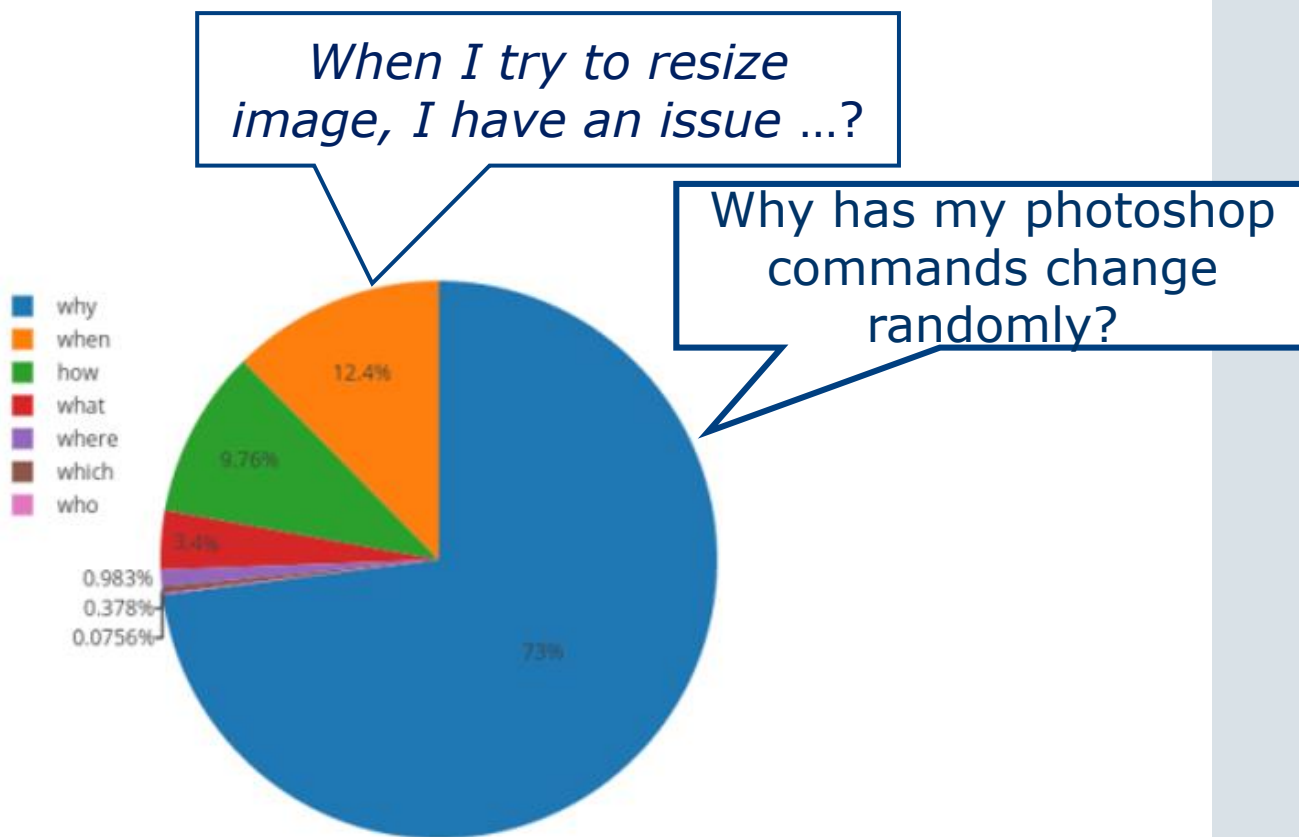
1. Tổng quan đề tài

WHAT, HOW
QUESTION

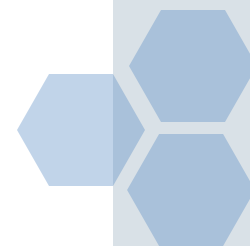




1. Tổng quan đề tài



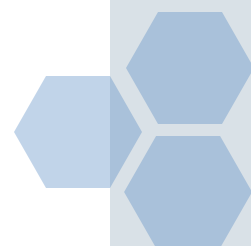
Biểu đồ tròn thống kê các phần trăm theo loại câu hỏi về Adobe Photoshop dựa trên lần xuất hiện đầu tiên của từ để hỏi trong câu hỏi (tổng 1,322 câu hỏi)





1. Tổng quan đề tài

Thành phần tri thức	Số lượng
What	557
How	1334
Tổng cộng	1891





Câu hỏi What?

Thuộc loại câu hỏi về khái niệm, định nghĩa, liên quan đến một đối tượng bất kì của phần mềm kỹ thuật. Nó được cấu thành bằng:

What + Objects + Contexts

What is **Panel**?

What is **Recording Tool** in **CS6**?

Can you tell me about **Camera Raw**?

I don't know what is **Layer** in **Photoshop CC**?





Câu hỏi How?

*Là những câu hỏi nhằm biết được cách thức hoạt động đối với các đối tượng của phần mềm kỹ thuật. Được cấu thành từ **How + Operations + Objects + Contexts***

How to use Recording Tool?

How to install Photoshop CS6 in Mac OS?

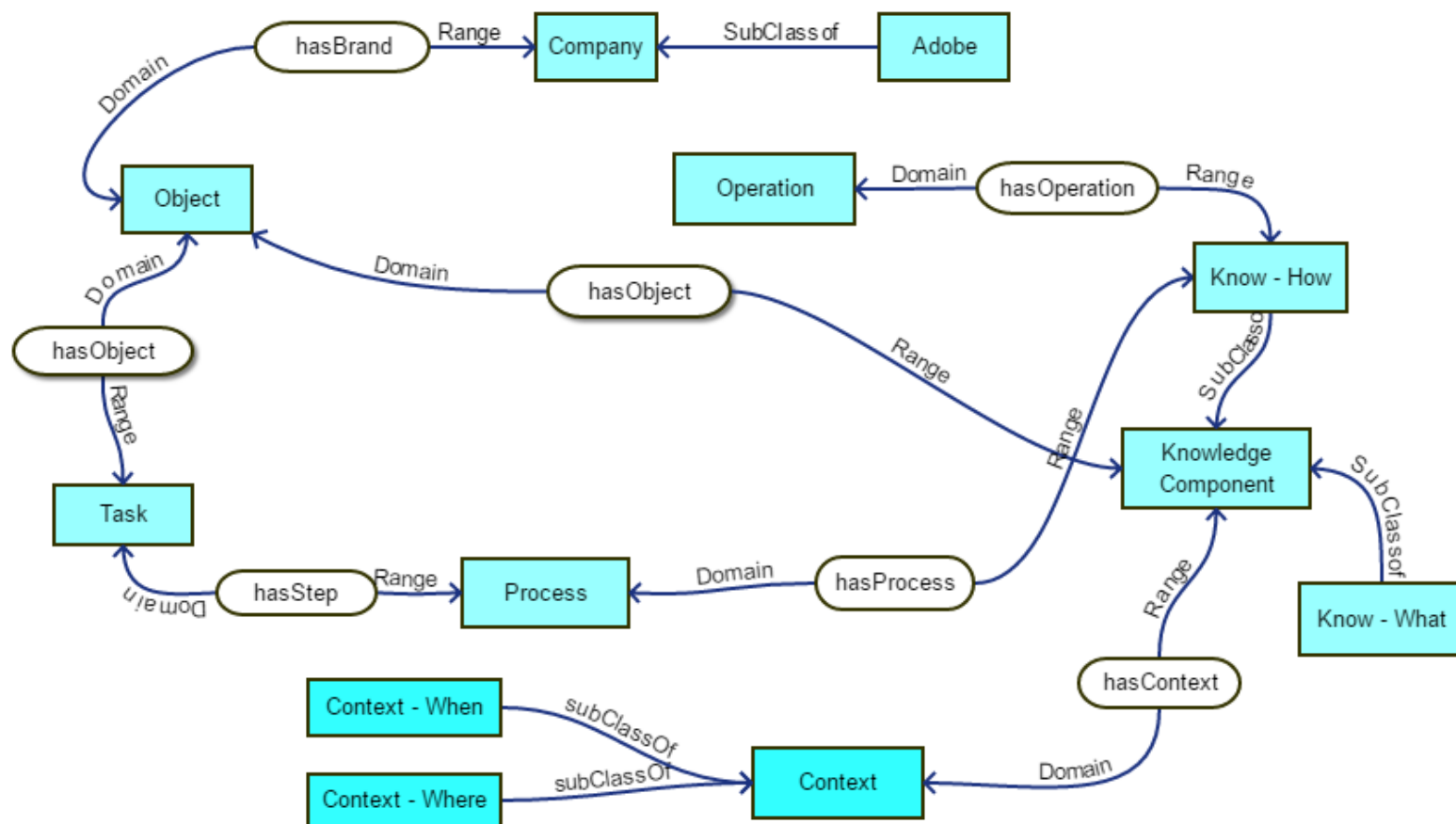
I want to rotate a picture in PS?

Can you tell me a way to resize image?



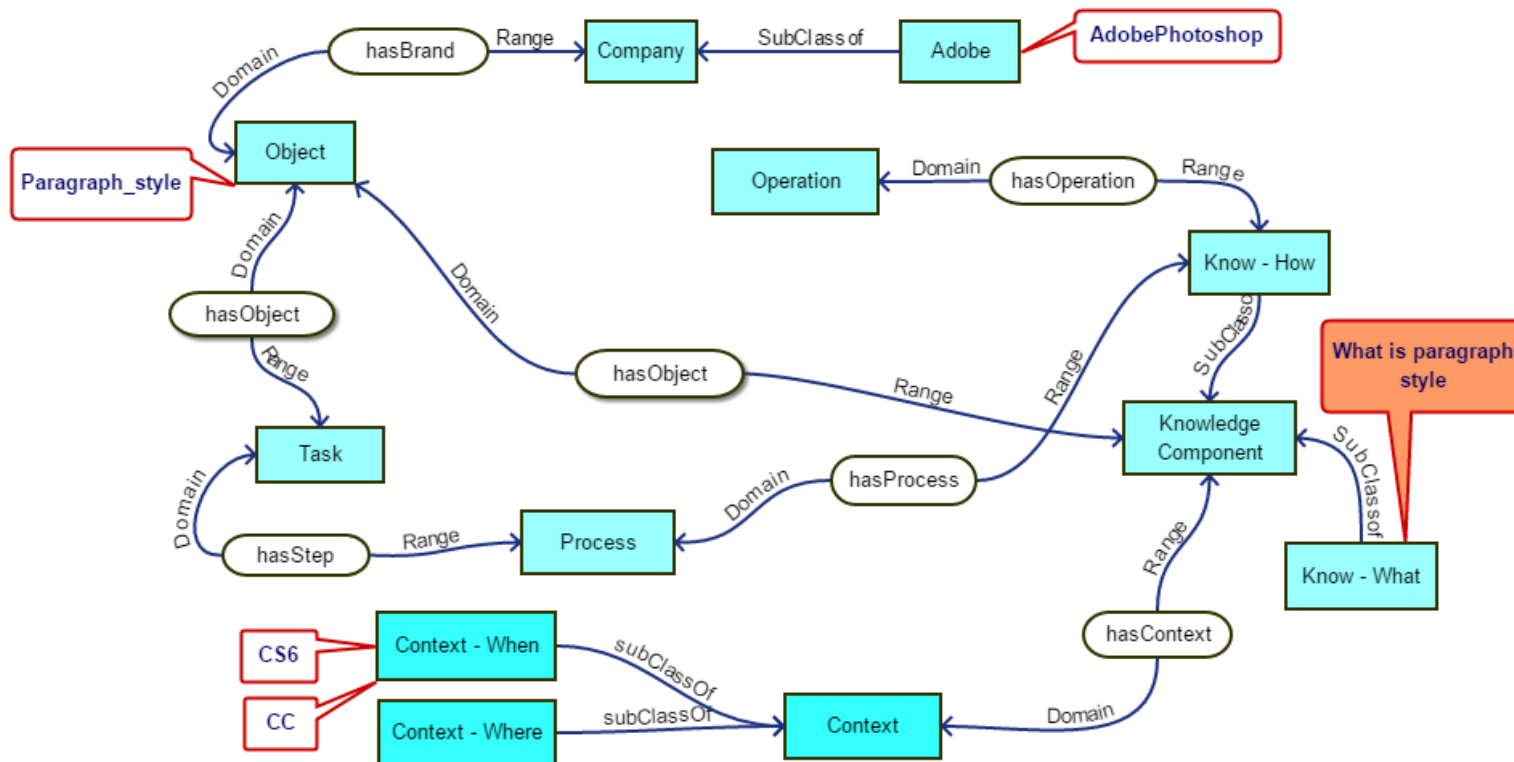


2. Xây dựng mô hình tri thức dựa vào Ontology



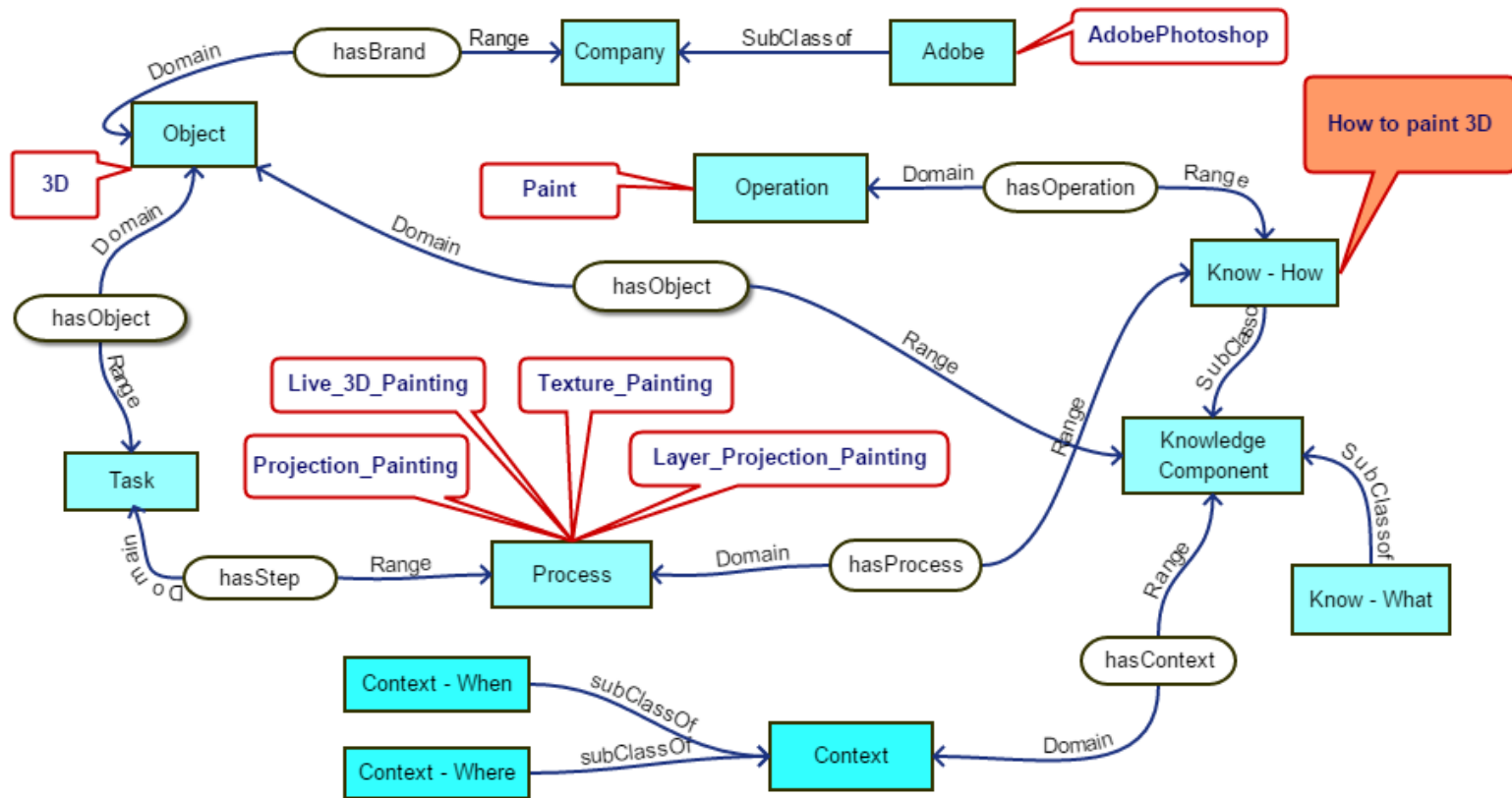


2. Xây dựng mô hình tri thức dựa vào Ontology





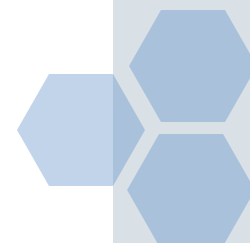
2. Xây dựng mô hình tri thức dựa vào Ontology





2. Xây dựng mô hình tri thức dựa vào Ontology

Class	Số lượng individuals
Object	160
Operation	54
Know - What	63
Know-How	87
Process	81
Talk	260
Company-Adobe	2
Context - When	7
Context - Where - OS	4
Context - Where - equipment	7





3. Xử lý thông tin đầu vào và xây dựng phản hồi với RASA

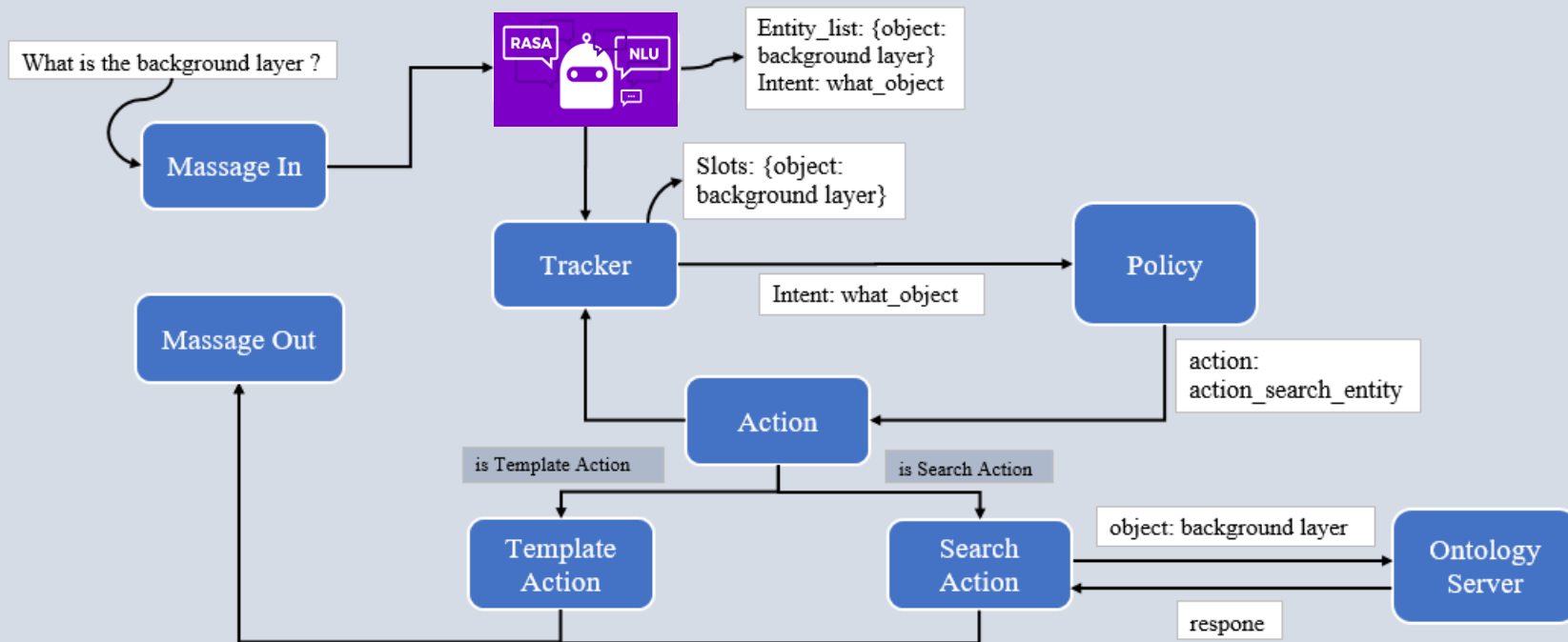


- Rasa là một trong những open source mới nổi khoảng cuối năm 2016 để xây dựng hệ thống tư vấn chatbot.
- Rasa bao gồm 2 thành phần là Rasa NLU và Rasa core
- Rasa NLU dùng để trích xuất thực thể và phân loại ý định
- Rasa core dùng để quản lý cuộc hội thoại





3. Xử lý thông tin đầu vào và xây dựng phản hồi với RASA

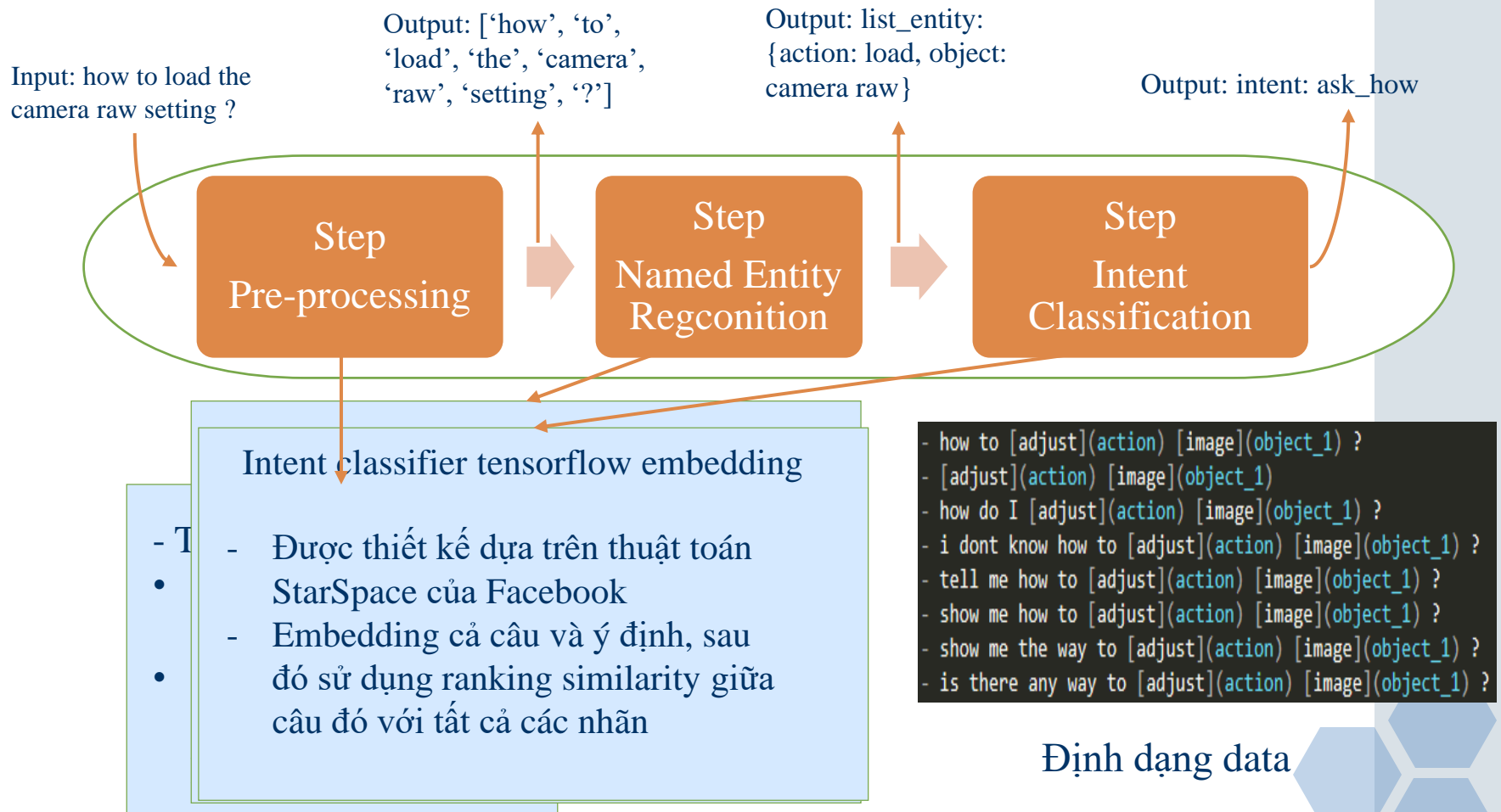


Mô hình hoạt động tổng quát của hệ thống



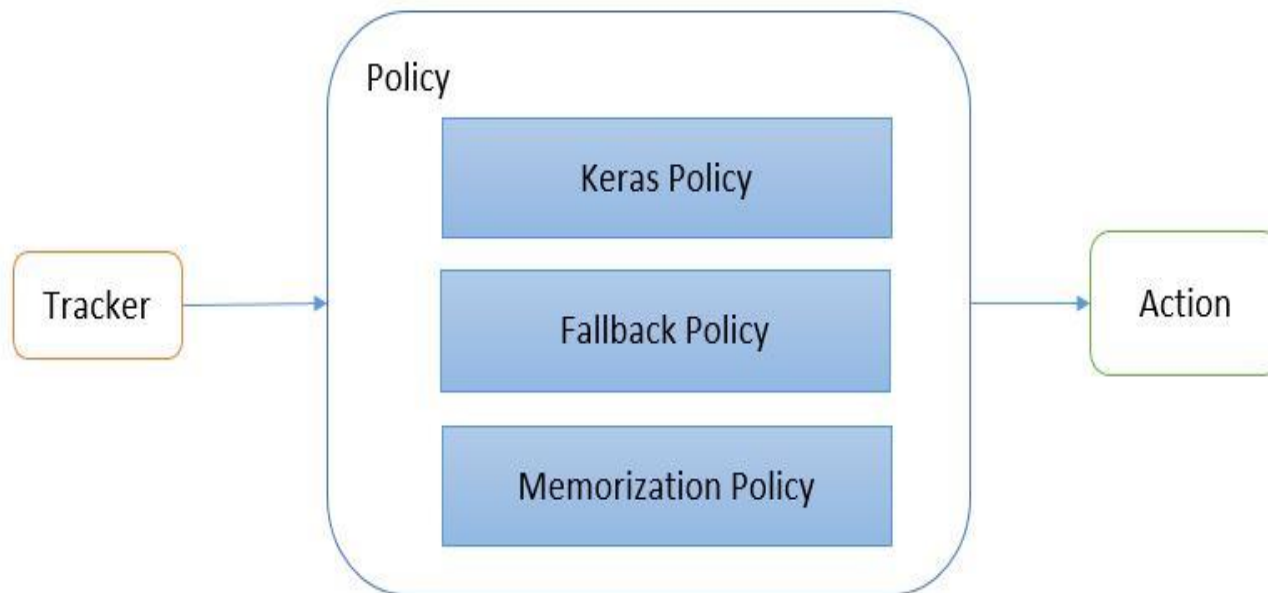


3.1 Rasa NLU (Natural Language Understanding)





3.2 Rasa core





Dữ liệu story

Kịch bản		
Câu hỏi và phản hồi		Ý định
User:	Hi there !	greet
Bot:	Hello, how can I help you ?	
User:	I'm having a problem with the photoshop	ask_how
Bot:	Oh, I'm so sorry, what is your problem ?	
User:	I can not install it on my laptop.	
Bot:	Oh, I see, which is your current version of photoshop ?	
User:	I'm using CS6	
Bot:	Which operating system do you use ?	
User:	Ubuntu	
Bot:	I have found a great video about that: https://youtu.be/QKdALjpM2GI	
User:	that's a great video. Thank bot	thankyou
User:	Your welcome !	
User:	Bye bro, see you next time	goodbye
Bot:	See you next time	

story

* greet

- utter_greet

* ask_how{"object_1":"photoshop"}

- utter_ask_for_help

* ask_how{"action":"install"}

- utter_ask_version

* ask_how{"version":"CS5"}

- utter_ask_os

* ask_how{"OS":"ubuntu"}

- action_search_how_answer

- action_show_process

* thankyou

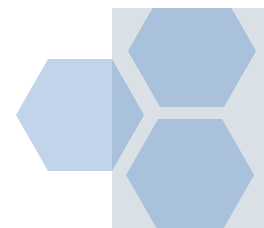
- utter_thank_you

* goodbye

- utter_goodbye

- action_renew

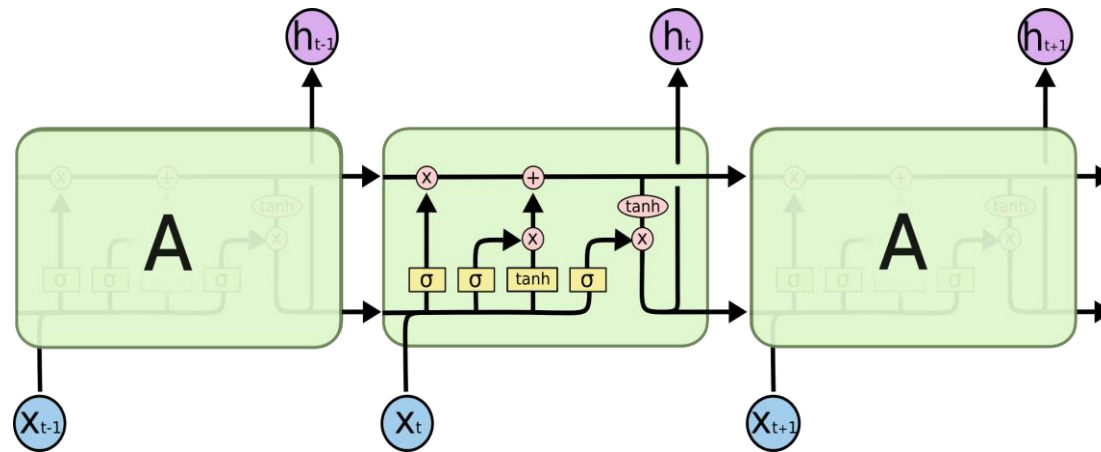
- restart





3.2 Rasa core – Keras Policy

Mạng LSTM



- Keras policy sử dụng neural network được cài đặt để đưa ra lựa chọn next_action.
- LSTM được chọn là Many-to-One
- input là list_intent, output là next_action.





3.2 Rasa core – Fallback & Memorization Policy

Memorization Policy:

- Sử dụng thông tin được ghi trong tracker đưa ra phản hồi. Nếu thông tin trong tracker trùng khớp với dữ liệu huấn luyện => predict next_action với độ chính xác là 1.0 và các action khác 0.0.
- Nếu không thể dự đoán thì sẽ tiếp tục đi đến Keras Policy để dự đoán.

Fallback Policy:

- Thành phần này giúp tự động trả lời 1 action mặc định khi độ tin cậy của việc dự đoán Rasa NLU < một ngưỡng cho trước, hoặc khi độ tin cậy của next_action < một ngưỡng cho trước, thì Fallback Policy sẽ được gọi và trả response mặc định cho user.

```
fallback = FallbackPolicy(fallback_action_name="action_default_fallback"  
                          core_threshold=0.3,  
                          nlu_threshold=0.3)
```



4. Thực nghiệm - đánh giá

- ❖ 4.1 Đánh giá các thuật toán phân loại.
- ❖ 4.2 Đánh giá mức độ hài lòng của người dùng
- ❖ 4.3 Đánh giá thời gian hồi đáp





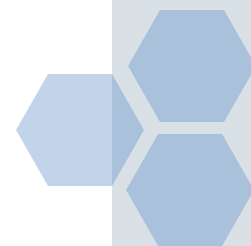
4.1 Đánh giá các thuật toán phân loại.

Huấn luyện
phân loại ý
định

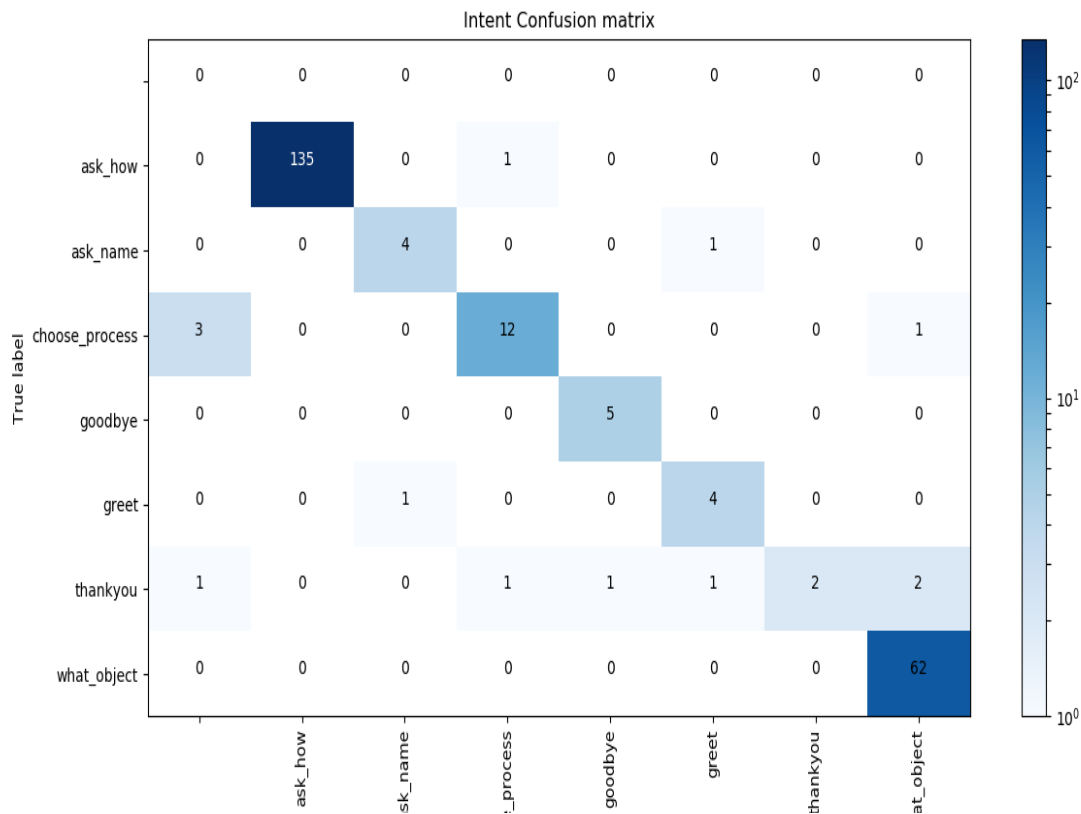
STT	Ý định	Data train	Data test
1	greet	13	5
2	what_object	683	62
3	ask_how	728	136
4	choose_process	26	16
5	thankyou	24	8
6	ask_name	6	5
7	goodbye	8	5
Tổng		1488	237

Huấn luyện
trích xuất
thực thể

STT	Thực thể	Data train	Data test
1	Object	1411	350
2	action	728	134
3	version	113	30
4	OS	30	6
5	equipment	20	6



4.1 Đánh giá các thuật toán phân loại.



Đánh giá phân loại ý định

F1-Score: 0.9460565663891736
Precision: 0.9635137014883851
Recall: 0.9451476793248945

	precision	recall	f1-score	support
ask_how	1.00	0.99	1.00	136
ask_name	0.80	0.80	0.80	5
choose_process	0.86	0.75	0.80	16
goodbye	0.83	1.00	0.91	5
greet	0.67	0.80	0.73	5
thankyou	1.00	0.25	0.40	8
what_object	0.95	1.00	0.98	62

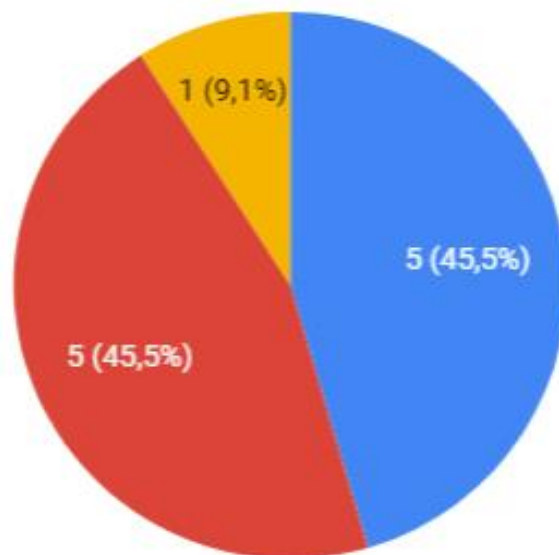
Đánh giá rút trích thực thể

Evaluation for entity extractor: ner_crf
F1-Score: 0.9319385721914843
Precision: 0.9320868803272344
Recall: 0.9326140450501609

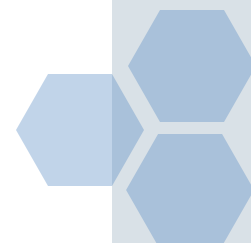


4.2 Đánh giá mức độ hài lòng của người dùng

Câu trả lời có đúng mong đợi của bạn?



- Không phù hợp nhưng có thể hiểu được (weird but understand-able)
- Phù hợp (reasonable)
- Hoàn toàn không phù hợp (nonsensical reply)



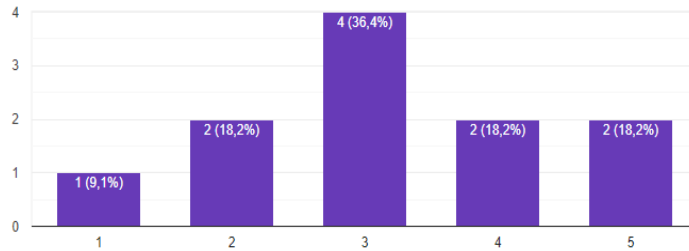


4.2 Đánh giá mức độ hài lòng của người dùng

Độ phù hợp của câu trả lời (câu trả lời có đúng với yêu cầu tìm kiếm?)



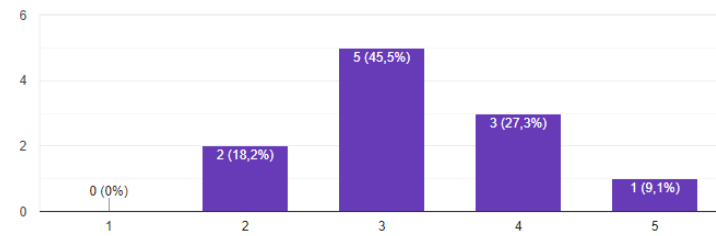
11 câu trả lời



Dễ sử dụng (có khó để bạn tìm ra câu trả lời phù hợp khi sử dụng chatbot?)



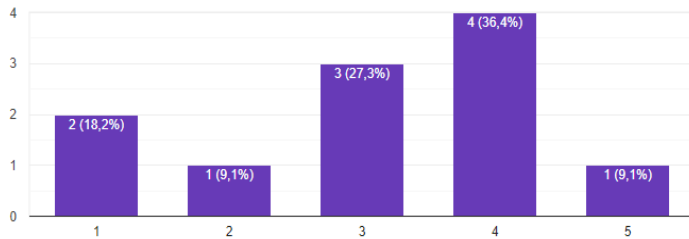
11 câu trả lời



Giao diện

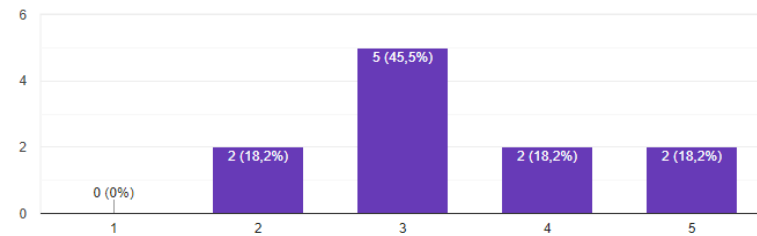


11 câu trả lời



Sự hài lòng (bạn có cảm thấy thoải mái khi sử dụng chatbot?)

11 câu trả lời

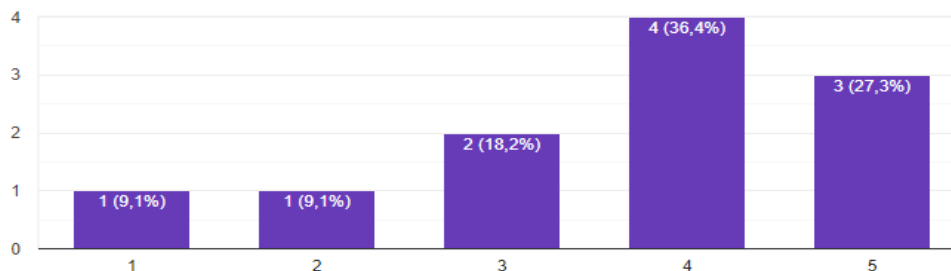




4.3 Đánh giá thời gian hồi đáp

Thời gian hồi đáp

11 câu trả lời



Server Ram 4GB			
	Số lượng entity	thời gian phản hồi (s)	thời gian phản hồi trung bình (s)
tell me about the layer	1	0.234336853	0.212831438
what is the ruler	1	0.156258106	
what is the composite	1	0.156209469	
tell me something about the selection	1	0.140565395	
Layer projection painting	1	0.218521357	
How download Photoshop CC	1	0.312495232	
Create a Photomerge composition	1	0.093657494	0.380135934
tell me the way to cast color	1	0.390607595	
tell me about the background layer	2	0.171800852	
what is the blend mode saturation	2	0.187489986	
is there any way to create a poster	2	0.781116962	0.808515906
how to paint 3d image	3	0.765568733	
tell me the way to install photoshop CC	3	0.875041962	
tell me some how to add text to image	3	1.1248281	
how to correct and check the text on Ubuntu ?	3	0.46862483	0.956290007
how to create 360 degree panorama on CS6	4	0.890580416	
how to install photoshop on window 10 verion CS6	4	0.999990225	
how to create web banner photoshop cs6 on ubuntu	4	0.978299379	



5. Video demo

Link sản phẩm

<https://chatbotui-187b1.wep.app>





6. Kết luận và hướng phát triển

- ❖ **6.1 Kết quả đạt được**
- ❖ **6.2 Hạn chế**
- ❖ **6.3 Hướng phát triển**

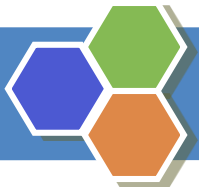




6.1 Kết quả đạt được

LÝ THUYẾT	THỰC HÀNH
<ul style="list-style-type: none">• Hiểu được khái niệm chatbot, cách vận hành của một chatbot.• Hiểu được khái niệm và tầm quan trọng của web ngữ nghĩa đối với tương lai của world wide web.• Hiểu được khái niệm và cách để xây dựng một web API Restful service hoàn chỉnh.• Hiểu được lý thuyết về xử lý ngôn ngữ tự nhiên, các thuật toán phân loại ý định và trích xuất thực thể.	<ul style="list-style-type: none">• Khả năng xây dựng một trang web API Resful service hoàn chỉnh.• Biết cách thiết kế, xây dựng, thêm dữ liệu, truy vấn và công bố ontology.• Biết cách sử dụng RASA như một công cụ để áp dụng các thuật toán xử lý ngôn ngữ tự nhiên nhằm bóc tách ý định người dùng và xây dựng phản hồi phù hợp.• Biết cách công bố, phát hành một ứng dụng.





6.2 Hạn chế

- ❖ Kiến thức trong lĩnh vực xử lý ngôn ngữ tự nhiên có hạn nên việc chuẩn bị dữ liệu còn thực hiện thủ công chứ chưa thể bóc tách dữ liệu từ nguồn có sẵn và tự động thêm vào cơ sở dữ liệu.
- ❖ Dữ liệu chuẩn bị chưa đầy đủ, chỉ ở mức đủ để demo giải pháp, ý tưởng.
- ❖ Phần giao diện làm chưa đủ tốt do chưa có kinh nghiệm trong việc thiết kế giao diện, cũng như chưa thật sự tập trung nhiều thời gian cho phần này.





6.3 Hướng phát triển

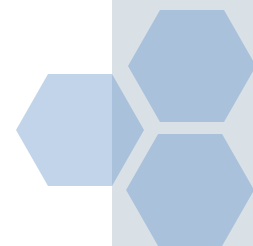
- ❖ Nghiên cứu xử lý tài liệu thô thành dữ liệu có cấu trúc để thêm vào dữ liệu ontology - Information extraction.
- ❖ Cần xây dựng chatbot có thể hiểu được không chỉ là câu hỏi what, how mà còn các câu hỏi phức tạp khác như câu hỏi nguyên nhân kết quả, hay câu hỏi giả thuyết.
- ❖ Nghiên cứu giúp chatbot có khả năng tự đánh giá phản hồi của người dùng để tăng độ xếp hạng của câu trả lời.
- ❖ Thay đổi việc trích xuất thực thể từ học có giám sát (supervised) sang hướng học không giám sát (unsupervised).
- ❖ Nghiên cứu phát triển giúp chatbot có thể nhận diện các lỗi dưới dạng hình ảnh do người dùng đưa vào và đưa ra phản hồi thích hợp.

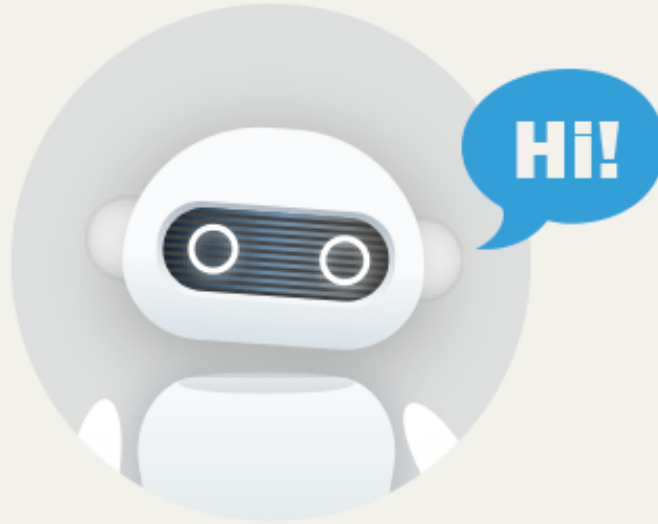




Tài liệu tham khảo

- ❖ T. L. D. W. C. T. B. D. S. K. M. C. V. S. K. Andrei Dulceanu, “PhotoshopQuiA: A Corpus of Non-Factoid Questions and Answers for Why-Question Answering,” Adobe Research, California, United States.
- ❖ A. S. m. Drift, The 2018 State of Chatbots Report.
- ❖ E. A. Bayan Abu Shavar, “Different measurements metrics to evaluate a chatbot system” U.K, 2007.
- ❖ “Mạng ngữ nghĩa,” [Trực tuyến]. Available: https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_ng%E1%BB%AF_ng%C4%A9a.
- ❖ <https://spacy.io>
- ❖ <https://rasa.com>

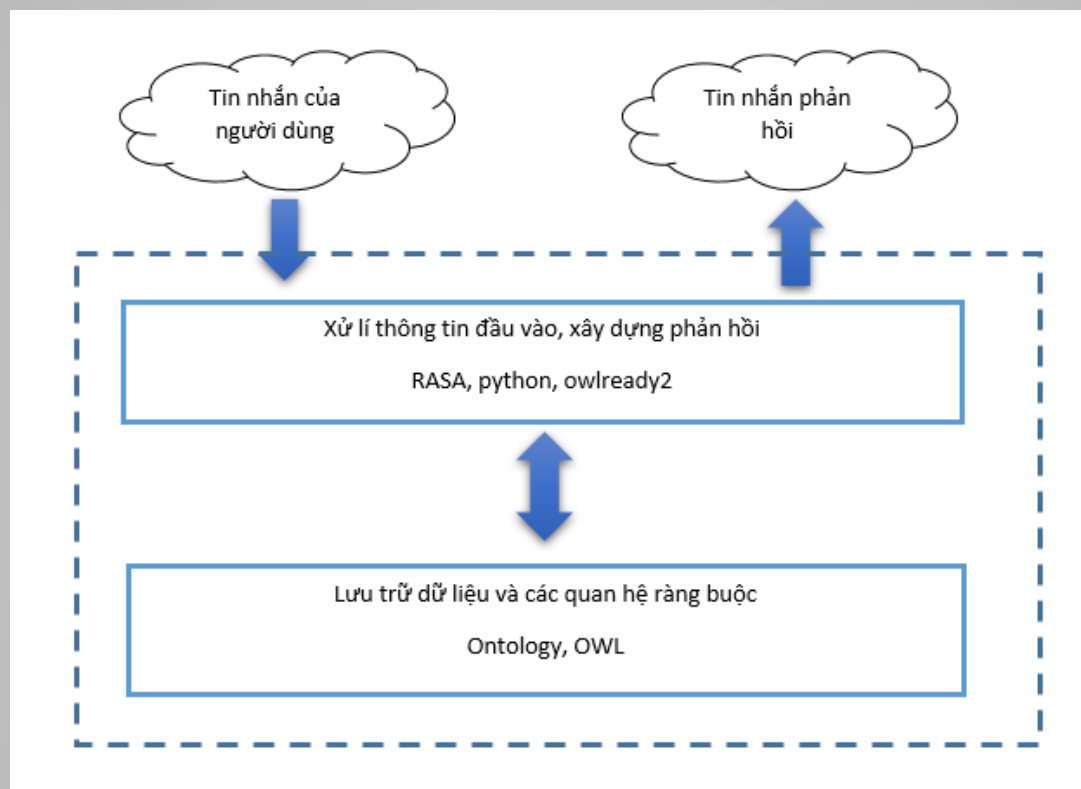




The end
Cảm ơn thầy cô và các bạn đã lắng nghe



1. Tổng quan đề tài



Sơ đồ hoạt động tổng quát





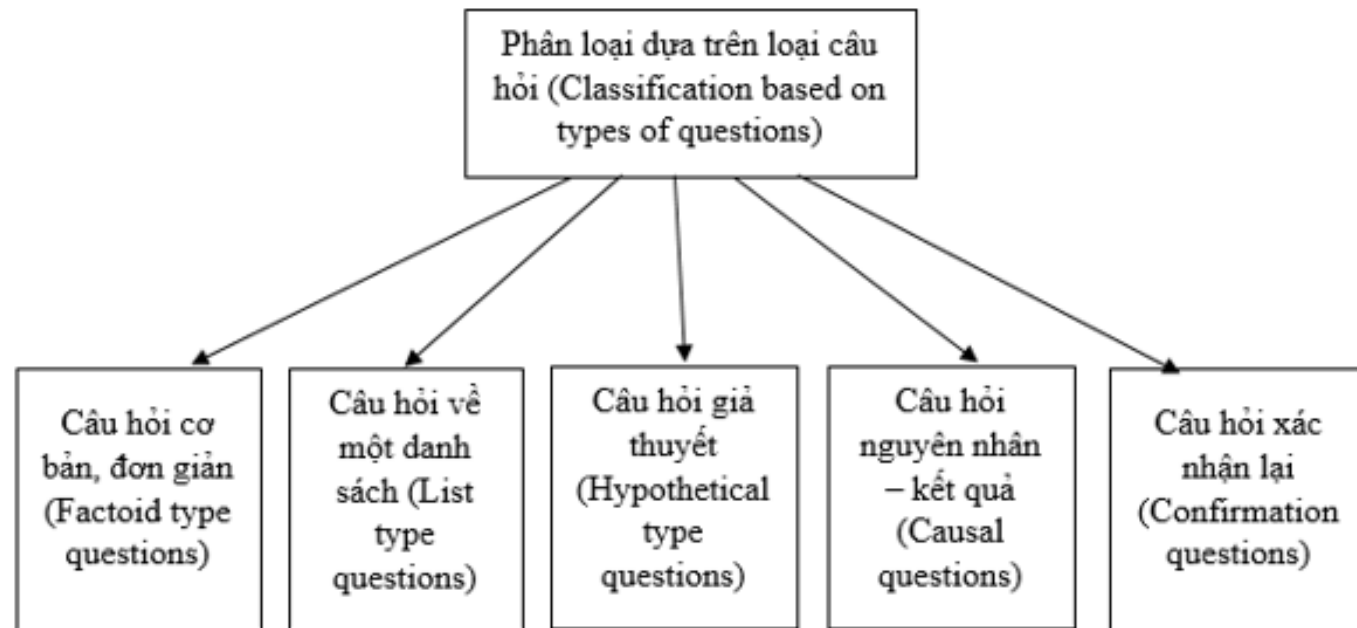
2. Web ngữ nghĩa và mô hình tri thức cho đề tài

- Được dẫn đầu bởi tổ chức W3C.
- Tiêu chuẩn này phát triển các định dạng dữ liệu chung trên World Wide Web.
- Hướng đến việc chuyển đổi các nội dung web hiện tại bao gồm phần lớn các văn bản không có cấu trúc hoặc bán cấu trúc thành dữ liệu có ngữ nghĩa.





1. Tổng quan đề tài

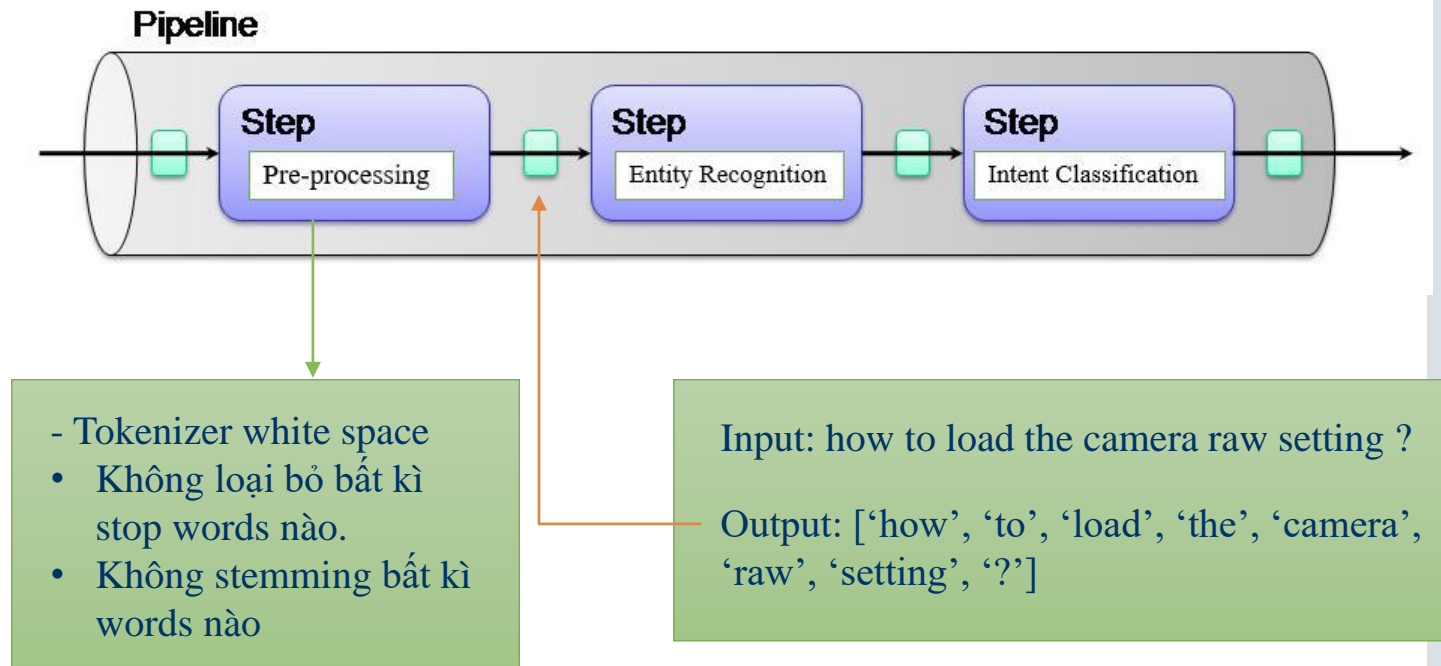


Hình 2. 5 Sơ đồ các phân loại QAS dựa trên loại câu hỏi

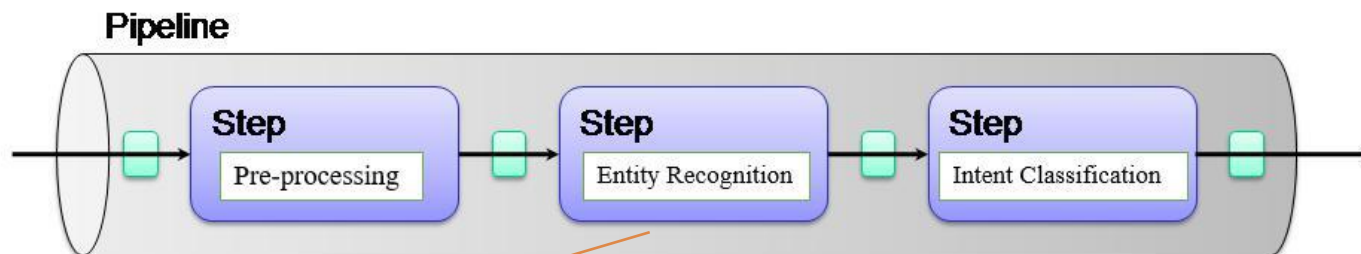




3.1 Rasa NLU (Interpreter)



3.1 Rasa NLU (Interpreter)



Thành phần sử dụng Ner_CRF

- + Convert Entity (start_index, end_index và entity)
- + Đánh nhãn BILOU
- + Huấn luyện dữ liệu
 - X_train: feature trước và sau token
 - Y_train: label thẻ BILOU

`[(7, 11, 'load'), (12, 22, 'camera raw')]`

tokens :: [how, to, load, the, camera, raw, setting]
BILOU :: [O, O, 'U-action', O, 'B-object', 'L-object', O]

```
crf = sklearn_crfsuite.CRF(algorithm='lbfgs',  
                           **component_config)  
crf.fit(X_train, y_train)
```



3.1 Rasa NLU (Interpreter)



Optimize hàm loss

- Mặc định tham số thay cho μ trong công thức thuần.
- + μ_{pos} : 0.8 độ tương đồng tối đa để thuật toán tạo ra vector embedding cho nhãn ý định đúng
- + μ_{neg} : -0.4 độ tương đồng tối đa cho các nhãn ý định sai.

```
+ sim_op = tf.reduce_sum(tf.expand_dims(a, 1) * b, -1)
```

```
# loss for maximizing similarity with correct action  
loss = tf.maximum(0., mu_pos - sim_op[:, 0])
```

```
# minimize only maximum similarity over incorrect actions  
max_sim_neg = tf.reduce_max(sim_op[:, 1:], -1)  
loss += tf.maximum(0., mu_neg + max_sim_neg)
```

```
# average the loss over the batch and add regularization losses  
loss = (tf.reduce_mean(loss) + tf.losses.get_regularization_loss())
```

4. Drop out 0.2, activation = ReLU

5. Output layer dim = 20