TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

❖❖❖❖❖❖❖❖❖❖❖❖

BÁO CÁO ĐỒ ÁN CUỐI KỲ

ỨNG DỤNG CHATBOT

CHATTERBOT

by Gunthercox

MÔN XỬ LÍ NGÔN NGỮ TỰ NHIÊN ỨNG DỤNG

LỚP CQ19-22

Ngô Nguyễn Nhật Hạ – 19120498

Phạm Văn Nam – 19120597

GIẢNG VIÊN LÝ THUYẾT

Thầy Nguyễn Hồng Bửu Long

Tp. Hồ Chí Minh, tháng 05/2022

Mục lục

1. Giới thiệu

1.1 Khái niệm chatbot

Chatbot là một phần mềm trí tuệ nhân tạo và là sự tương tác giữa người và máy (Human-computer Interaction) [1]. Chatbot là một chương trình máy tính được thiết kế để mô phỏng cuộc đối thoại với người dùng [2]. Chatbot sử dụng NLP và phân tích cảm xúc để giao tiếp với con người hoặc chatbot khác bằng ngôn ngữ của con người thông qua văn bản hoặc giọng nói [3].

Chatbot được ứng dụng vào nhiều lĩnh vực: giáo dục, kinh doanh, thương mại điện tử, y tế và giải trí. Trong kinh doanh, chatbot làm việc không mệt mỏi, giảm chi phí thuê nhân viên và có thể phục vụ nhiều người cùng một lúc.

1.2 Lịch sử chatbot

- Năm 1966, chatbot đầu tiên tên ELIZA ra đời. Bot này đóng vai một nhà tâm lý học, nhận ra các từ khóa từ input của người dùng và trả lời bằng các tập lệnh viết sẵn. Mặc dù, ELIZA có nhiều điểm yếu nhưng nó là nền tảng phát triển cho nhiều chatbot.
- Năm 1972, chatbot PARRY - bệnh nhân tâm thần phân liệt xuất hiện. Nó xác định câu phản hồi dựa trên hệ thống các giả định và sự thay đổi trong lời nói của người dùng sẽ kích hoạt phản hồi cảm xúc. Tuy nhiên, PARRY lại được xem là chatbot với khả năng hiểu ngôn ngữ kém, tốc độ phản hồi chậm và không thể học từ các cuộc đối thoại.
- Năm 1988, chatbot Jabberwacky kết hợp trí tuệ nhân tạo ra đời. Chatbot này được viết bằng ngôn ngữ CleverScript – ngôn ngữ dựa trên bảng tính (spreadsheet), đã tạo điều kiện phát triển cho nhiều chatbot khác.
- Năm 1955, chatbot ALICE được lấy cảm hứng từ ELIZA ra đời. ALICE hoạt động dựa trên so khớp mẫu (pattern-matching) mà không cần biết về toàn bộ cuộc hội thoại. ALICE sử dụng lược đồ XML được gọi là ngôn ngữ đánh dấu trí thông minh nhân tạo (AIML). Câu phản hồi của ALICE không biểu đạt cảm xúc con người.
- Năm 2001, công nghệ phát triển chatbot cải tiến mạnh mẽ với sự phát triển của SmarterChild có mặt trên các hệ thống Messengers như AOL, MSN. Lần đầu tiên, một chatbot có thể giúp con người với các công việc hằng ngày như thu thập thông tin từ cơ

sở dữ liệu về thời gian chiếu phim, kết quả thi đấu thể thao, giá tiền của các sản phẩm, tin tức và thông tin thời tiết.

- Trong những năm gần đây, sự phát triển của Trí tuệ nhân tạo đưa chatbot lên một tầm cao mới với sự ra đời của trợ lý ảo bằng giọng nói nằm trong các thiết bị thông minh như Apple Siri (2010), IBM Watson (2011), Google Assistant (2012), Microsoft Cortana (2014), Amazon Alexa (2014). Những Chatbot này có thể kết nối Interrnet, đưa ra phản hồi có nghĩa, thực hiện các công việc không quá phức tạp.

1.3 Loại chatbot

Knowledge Domain:

- Generic: trả lời bất kỳ câu hỏi trong bất kỳ miền/lĩnh vực
- Open Domain: trả lời câu hỏi thuộc về nhiều hơn một miền/lĩnh vực
- Closed Domain: trả lời câu hỏi chỉ trong một miền/ lĩnh vực

Service provided:

- Interpersonal: phục vụ trong đặt vé khách sạn, nhà hàng, máy bay; trả lời FAQ, …
- Intrapersonal: "một người bạn đồng hành", theo dõi và hiểu rõ nhu cầu người dùng.
- Inter-agent: có khả năng giao tiếp với các chatbot khác

Goal:

- Informative: thu thập thông tin cụ thể tại một nguồn cố định
- Conversational: giao tiếp tự nhiên với người dùng
- Task based: thực hiện các task, các chức năng như đặt phòng, …

Response Generation method:

- Rule based: so khớp input người dùng với một luật khuôn mẫu (rule pattern) và chọn một câu trả lời đã được định trước trong tập những phản hồi bằng thuật toán Pattern matching. Phương pháp này có tính lặp lại, không tạo ra câu trả lời mới, phụ thuộc vào độ đa dạng của bộ luật, khó khăn khi xử lý các lỗi ngữ pháp, cú pháp.
- Retrieval based: sử dụng các bộ logic, kho ngữ liệu có cấu trúc và các thuật toán học máy để lấy câu phản hồi từ một tập có sẵn.
- Generative based: Tự sinh ra câu trả lời từ input (theo cơ chế encoder-decoder).

Human-aid:

- Human mediated: sử dụng sự trợ giúp tính toán từ con người
- Fully automonous: hoàn toàn tự động

Phương pháp tiếp cận:

- Pattern matching: đa số sử dụng các phương pháp như rule based, classification algorithm và searching algorithm
- Machine learning: sử dụng các mô hình deep learning như RNN, LSTM, Seq2Seq, Transformer, BERT, GPT, …

1.4 Về ChatterBot

- Là thư viện giúp tự động trả lời câu hỏi từ người dùng
- Sử dụng thuật toán học máy để xuất ra các câu trả lời khác nhau

1.4.1 Các yêu cầu cài đặt và thư viện bắt buộc (sử dụng tại các IDE)

- Python 3.8
- chatterbot==1.0.8 [4]
- chatterbot_corpus
- coveralls
- flake8
- nltk>=3.2,<4.0
- nose
- pint>=0.8.1
- pymongo>=3.3,<4.0
- twine
- twython
- spacy>=2.1,<2.2
- sphinx>=3.0,<3.1
- sphinx_rtd_theme

- pyyaml>=5.3,<5.4
- https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-2.1.0/en_core_web_sm-2.1.0.tar.gz#egg=en_core_web_sm
- https://github.com/explosion/spacy-models/releases/download/de_core_news_sm-2.1.0/de_core_news_sm-2.1.0.tar.gz#egg=de_core_news_sm

1.4.2 Một số phiên bản Release

- 1.0.8 (stable, lately updated) on Aug 23, 2020
- 1.0.7 on Aug 16, 2020
- 1.1.0a7 on May 2, 2020
- 1.1.0a6 on May 2, 2020

1.4.3 Ưu – nhược điểm

Ưu điểm:

- Dễ dàng cài đặt
- Thời gian thực thi nhanh
- Mô hình dễ tiếp cận
- Tương thích với mọi ngôn ngữ
- Mã nguồn mở cho phép tuỳ chỉnh theo mục đích sử dụng không thương mại hoá

Nhược điểm:

- Thuật toán chọn mặc định chỉ tốt về thời gian nhưng chưa tốt về kết quả
- Cần lượng dữ liệu sẵn nhiều và ‘tốt’ để đưa ra câu trả lời đang dạng và ‘tốt’
- Việc cài đặt thư viện chưa được hỗ trợ tốt vì tác giả ít nâng cấp công nghệ mới

1.4.4 Về khả năng tương thích đa dạng ngôn ngữ

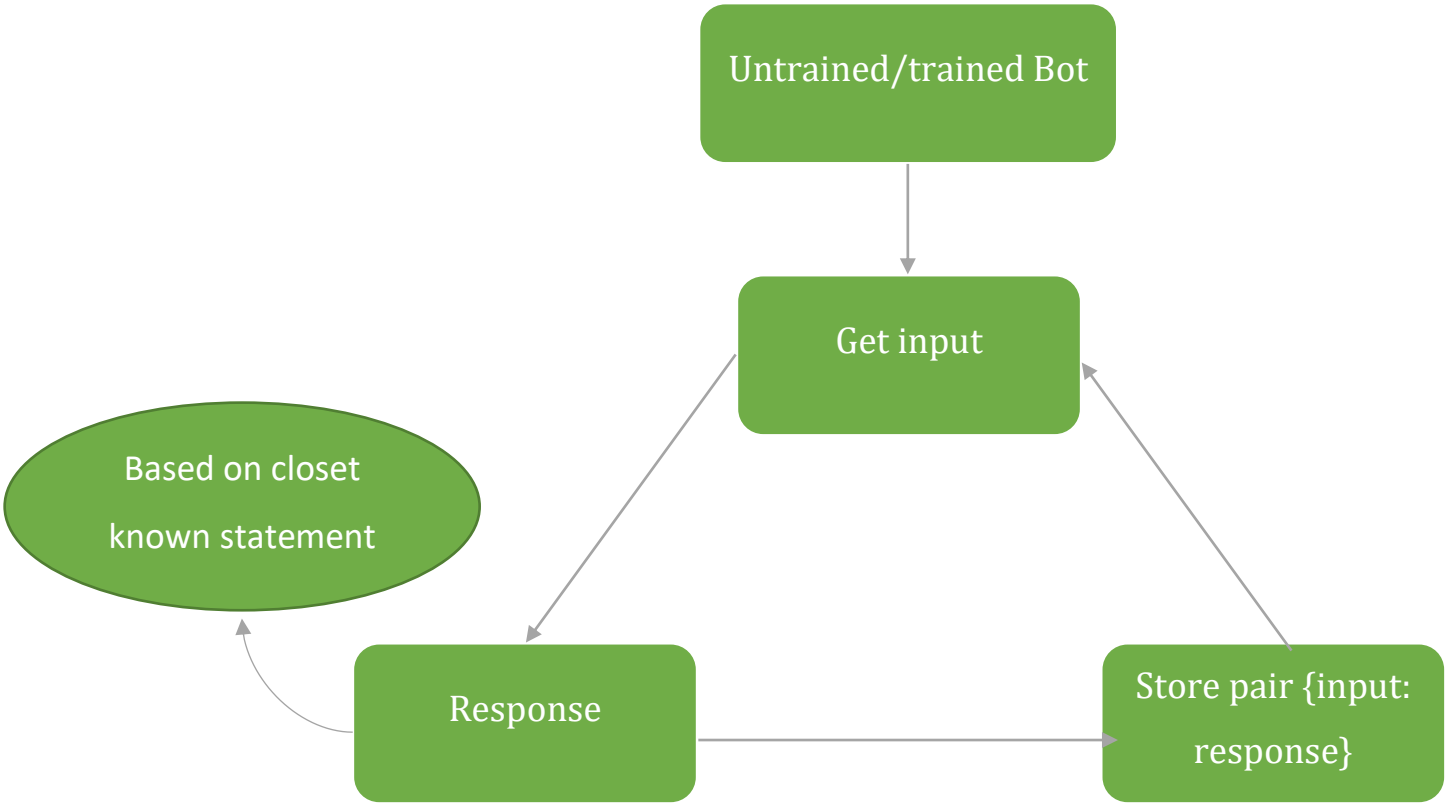
ChatterBot có thể xử lý các giá trị unicode một cách chính xác. Bạn có thể chuyển cho nó dữ liệu không được mã hóa và nó sẽ có thể xử lý đúng cách (bạn sẽ cần đảm bảo rằng bạn giải mã đầu ra được trả về). [5]

2. Quy trình hoạt động

2.1 Quy trình

Một Chatbot khi chưa được train sẽ không có kiến thức về cách giao tiếp. Mỗi khi người dùng nhập vào một statement, thư viện lưu văn bản vừa nhập và văn bản mà chatbot phản hồi lại. Khi nhận được càng nhiều input thì số lượng các phản hồi và độ chính xác liên quan đến input đó sẽ tăng theo.

Chương trình lựa chọn phản hồi phù hợp nhất bằng cách tìm kiếm statement đã biết gần nghĩa nhất với input, sau đó chọn một phản hồi từ các phản hồi đã biết của statement đó.



Bảng 2.1: Process flow diagram

2.2 Các thuật toán máy học

Theo chatterbot document, ChatterBot sử dụng một vài thuật toán máy học khác nhau để đưa ra phản hồi thích hợp. Sau đây là một số thuật toán được dùng cơ bản trong code:

- Search algorithms: thuật toán tìm kiếm cần thiết để bot nhanh chóng tìm kiếm hiệu quả và trả về các "ứng cử viên cho phản hồi" (candidate statement). Việc tìm kiếm sẽ dựa trên:

- Sự tương đồng giữa input statement và statement đã biết
- Tần số xuất hiện của các câu phản hồi tương tự mà đã biết
- Likeliness (sự phù hợp) của một input statement thuộc về thể loại của statement đã biết

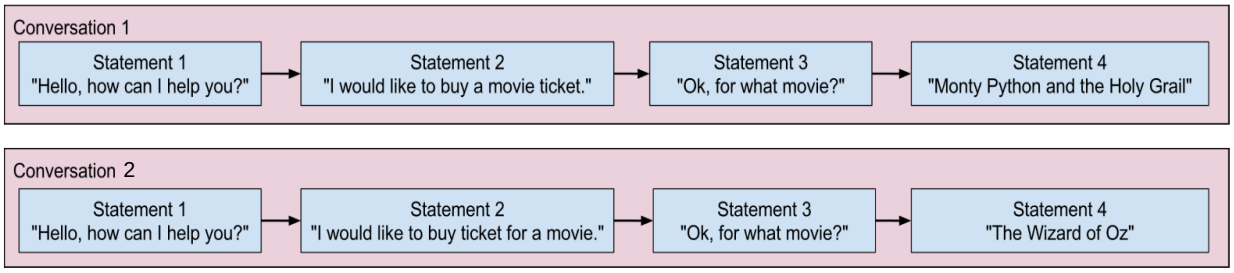
o Classification algorithms: một vài logic adapter sử dụng thuật toán phân lớp Naïve Bayesian để xác định xem một input statement có thỏa các tiêu chí của một câu phản hồi để từ đó logic adapter có thể sinh ra câu phản hồi phù hợp

3. Ngữ liệu – dữ liệu huấn luyện

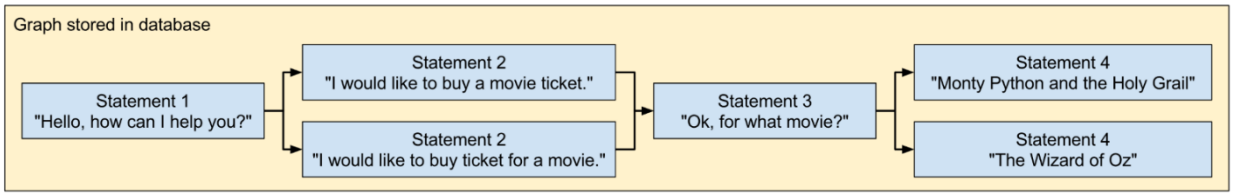
3.1 Huấn luyện

Quá trình huấn luyện chatbot liên quan đến việc lưu các cuộc đối thoại mẫu vào chatbot database. Khi đó, dữ liệu sẽ được tạo mới hoặc thêm vào cấu trúc dữ liệu đồ thị (graph data structure) của các câu phản hồi đã biết.

Khi cung cấp cho chatbot trainer một data set, nó sẽ tạo ra các entry trong đồ thị tri thức (knowledge graph) để biểu diễn input statement và response một cách hợp lý.



Hình 3.1: Graph data structure



Hình 3.2: Knowledge graph

Có nhiều built-in class trong ChatterBot cho phép cập nhật dữ liệu của chatbot (database knowledge graph) bằng list các statement tương tự như một đoạn đối thoại, hoặc huấn luyện bot bằng các corpus có sẵn.

3.2 Huấn lyện trên ngữ liệu/ dữ liệu

o Huấn luyện bằng list các statement

Truyền vào dãy các câu, thứ tự của câu sẽ dựa vào vị trí của câu đó trong một cuộc hội thoại

VD: huấn luyện bot phản hồi “Hello” khi câu input là “Hi there!” hoặc “Greetings!”

chatbot.py

```
chatbot = ChatBot('Training Example')
```

train.py

```
from chatbot import chatbot
from chatterbot.trainers import ListTrainer

trainer = ListTrainer(chatbot)

trainer.train([
    "Hi there!",
    "Hello",
])

trainer.train([
    "Greetings!",
    "Hello",
])
```

Lưu ý: khi khởi tạo một chatbot, cần đặt tên cho bot

- Huấn luyện bằng corpus

```
from chatbot import chatbot
from chatterbot.trainers import ChatterBotCorpusTrainer

trainer = ChatterBotCorpusTrainer(chatbot)

trainer.train(
    "chatterbot.corpus.english"
)
```

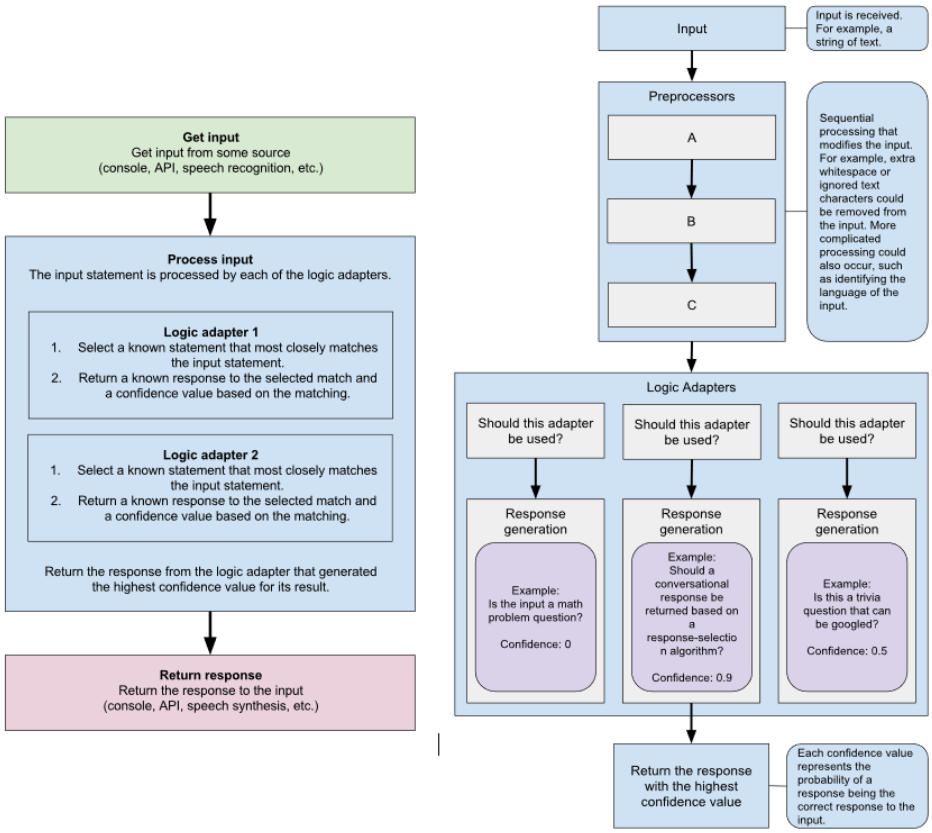
Corpus có sẵn trong source code: bengali, chinese, english, french, german, Hebrew, hindi, indonesian, italian, japanese, korean, marathi, oriya, persian, portuguese, russian, spanish, telugu, thai, traditionalchinese, turkish.

- Huấn luyện trong miền nhất định, ví dụ như huấn luyện bot chỉ ở ngôn ngữ là tiếng anh, trong phạm vi ngữ liệu xoay quanh chủ đề những câu chào hỏi, những cuộc đối thoại.

```
trainer.train(
    "chatterbot.corpus.english.greetings",
    "chatterbot.corpus.english.conversations"
)
```

4. Các bộ chuyển đổi logic

Vai trò của bộ chuyển đổi logic: quyết định cách chatterbot lựa chọn một phản hồi dựa trên input statement

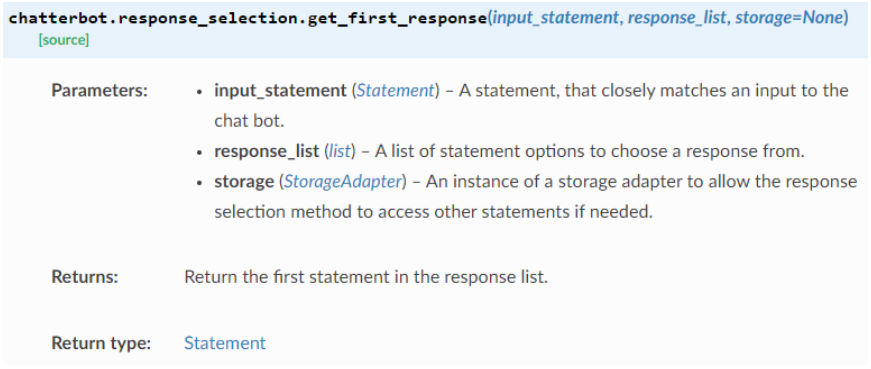


Hình 4.1: Decision flow diagram

Nếu có thiết lập nhiều logic adapter thì chatbot sẽ chọn response từ logic adapter có confidence (nằm giữa 0 và 1) cao nhất để trả về kết quả cho người dùng. Nếu có nhiều logic adapter có cùng mức confidence thì chatbot sẽ chọn logic adapter đầu tiên.

Response selection method: xác định phản hồi nào nên chọn trong các phản hồi trong một logic adapter

Các hàm selection method:



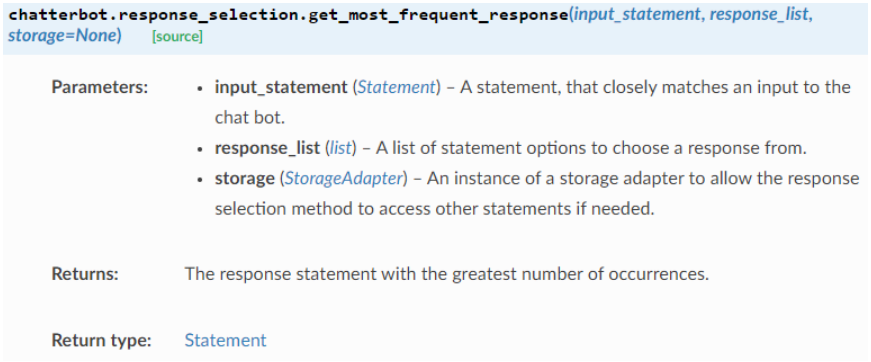
chatterbot.response_selection.get_first_response(input_statement, response_list, storage=None) [source]

Parameters:
- input_statement (Statement) – A statement, that closely matches an input to the chat bot.
- response_list (list) – A list of statement options to choose a response from.
- storage (StorageAdapter) – An instance of a storage adapter to allow the response selection method to access other statements if needed.

Returns: Return the first statement in the response list.

Return type: Statement

Hình 4.2: Get first response (default)



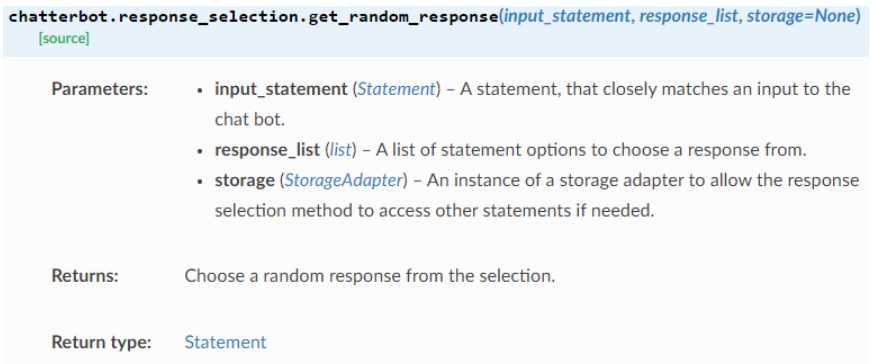
chatterbot.response_selection.get_most_frequent_response(input_statement, response_list, storage=None) [source]

Parameters:
- input_statement (Statement) – A statement, that closely matches an input to the chat bot.
- response_list (list) – A list of statement options to choose a response from.
- storage (StorageAdapter) – An instance of a storage adapter to allow the response selection method to access other statements if needed.

Returns: The response statement with the greatest number of occurrences.

Return type: Statement

Hình 4.3: Get most frequent response



chatterbot.response_selection.get_random_response(input_statement, response_list, storage=None) [source]

Parameters:
- input_statement (Statement) – A statement, that closely matches an input to the chat bot.
- response_list (list) – A list of statement options to choose a response from.
- storage (StorageAdapter) – An instance of a storage adapter to allow the response selection method to access other statements if needed.

Returns: Choose a random response from the selection.

Return type: Statement

Hình 4.4: Get random response

Statement comparison: chatbot sử dụng Object Statement để nắm giữ thông tin statement. Khi lựa chọn một phản hồi, bot sẽ so sánh 2 statement với nhau.

Comparison fuction trong source code:

- LevenshteinDistance (default)
- SpacySimilarity (use SpacyModel)
- JaccarSimilarity (Jaccard index)

Trong source code, có 5 built in logic adapter:

- Best match (default): trả về phản hồi dựa trên các phản hồi đã biết mà gần giống nhất với input statement

  Parameter excluded_words: danh sách các từ mà chatbot sẽ không trả về phản hồi chứa chúng. Điều này hữu ích nếu muốn ngăn chatbot "swear"
- Mathematical evaluation: xác định xem người dùng có đang hỏi câu hỏi liên quan đến toán không, nếu có thì trích xuất các phương trình, biến, … từ input và trả về kết quả đã được tính toán. Adapter này xử lý ký hiệu toán học dạng chữ cái và dạng chữ số.
- Specific response: trả về một phản hồi cụ thể dựa trên một input cụ thể
- Time adapter: trả về thời gian hiện tại

  + Positive: câu hỏi liên quan đến thời gian để xác định thời gian (what time is it)

  Negative: câu hỏi không liên quan đến thời gian để xác định thời gian (it is time to go to sleep)
- Unit conversion: phân tích input và chuyển đổi các đại lượng.

5. Các bộ chuyển đổi kho chứa

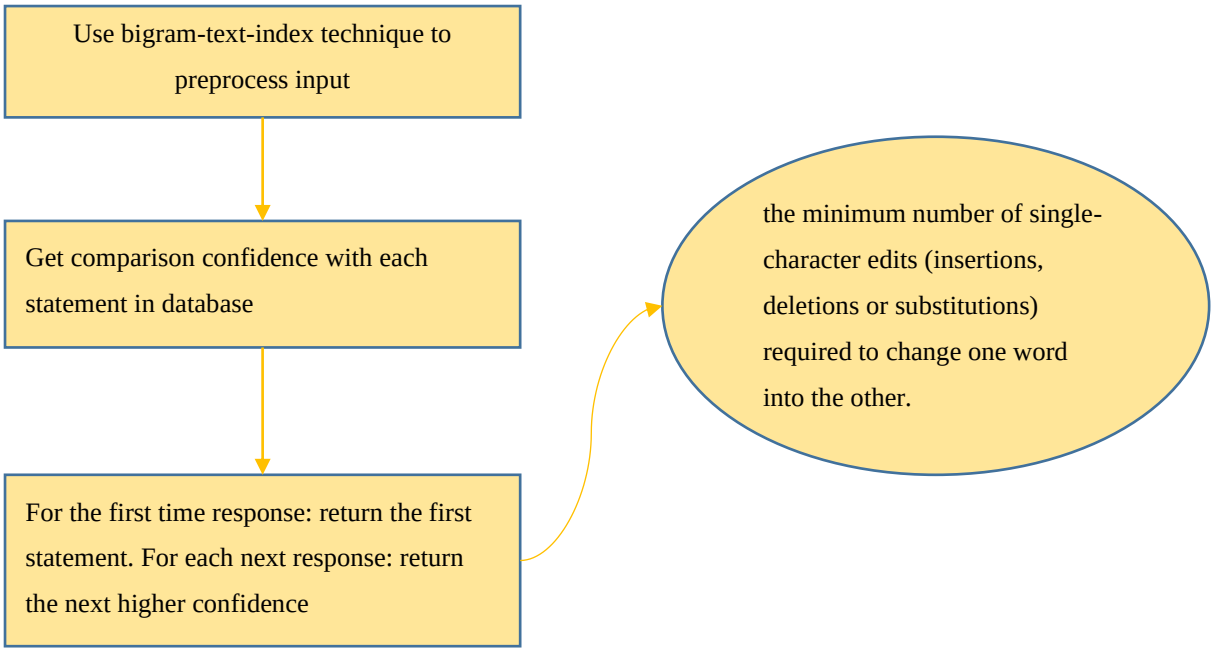
Các loại database được hỗ trợ tương thích cho việc lưu trữ và truy xuất data.

Các loại database được hỗ trợ:

- SQL storage (SQLAlchemy)
- Django storage
- MongoDB

Hỗ trợ hàm Text Search:

- Kiểm tra câu hỏi đầu vào với các 'câu hỏi' trong database, sử dụng kỹ thuật bigram-text-index để giảm số lượng kết quả có thể trùng nhau và giảm thời gian thực thi.
- Sử dụng thuật toán Levenshtein distance để đưa ra mức độ tương đồng giữa 2 từ



Hình 5.1: TextSearch flow diagram

Hàm tạo bigram-text-index (theo document):

- Tiền xử lí input (generalize, remove stopwords...)
- Tokenize input
- Loại bỏ prefix – suffix (đưa về dạng gốc của token)
- Tạo bigram với mỗi cặp token

Hàm tạo bigram-text-index hay get_index_text_string (theo mã nguồn):

- Sử dụng thư viện spacy để document câu input
- Tiền xử lí các token (generalize, remove stopwords...)
- Gắn cặp POS-lemma lần lượt các token sau khi xử lí và thêm vào list
- Tạo câu từ list với các cặp cách nhau bằng khoảng trắng

6. Các lớp được phép tuỳ chỉnh theo nhu cầu của người dùng

Preprocessors

Các bộ tiền xử lí sẽ định dạng input statement trước khi đưa vào các bộ chuyển đổi logic

Có thể thêm các tuỳ chỉnh tiền xử lí vào mã nguồn theo đường dẫn ‘ChatterBot/chatterbot/preprocessors.py’

Training class

Có thể tạo một huấn luyện viên mới để huấn luyện bot trò chuyện của mình từ các tệp dữ liệu của riêng bạn, từ nguồn dữ liệu ở định dạng không được ChatterBot hỗ trợ trực tiếp.

Training class nên kế thừa lớp ‘chatterbot.trainers.Trainer’. Trainer sẽ cần phải có một phương thức có tên là train, có thể lấy bất kỳ thông số nào do người lập trình chọn.

Có thể thêm các tuỳ chỉnh tiền xử lí vào mã nguồn theo đường dẫn ‘ChatterBot/chatterbot/trainers.py’

Logic adapters

Có thể viết bộ điều hợp logic của riêng bằng cách tạo một lớp mới kế thừa từ LogicAdapter và ghi đè các phương thức cần thiết được thiết lập trong lớp cơ sở LogicAdapter

Có thể thêm các tuỳ chỉnh tiền xử lí vào mã nguồn theo đường dẫn ‘ChatterBot/chatterbot/logic/logic_adapter.py’

Filters

Bộ lọc là một cách hiệu quả để tạo các truy vấn có thể được chuyển tới bộ điều hợp lưu trữ của ChatterBot. Bộ lọc sẽ giảm số lượng câu lệnh mà bot trò chuyện phải xử lý khi nó đang chọn một phản hồi.

Có thể thêm các tuỳ chỉnh tiền xử lí vào mã nguồn theo đường dẫn ‘ChatterBot/chatterbot/filters.py’

Storage adapters

Có thể viết bộ điều hợp lưu trữ của riêng mình bằng cách tạo một lớp mới kế thừa từ StorageAdapter và ghi đè các phương thức cần thiết được thiết lập trong lớp StorageAdapter cơ sở. Cần cài đặt giao diện được thiết lập bởi lớp StorageAdapter.

Có thể thêm các tuỳ chỉnh tiền xử lí vào mã nguồn theo đường dẫn ‘ChatterBot/chatterbot/storage/storage_adapter.py’

7. Đánh giá mô hình

Tác giả của mô hình chatterbot không đề cập đến đánh giá chatterbot nên nhóm sẽ có một số đề xuất:

Để tránh tình trạng chatbot không thể trả lời người dùng hoặc đưa ra câu trả lời không hề chính xác: Sử dụng các threshold để xác định ngưỡng mà chatbot không thể tìm ra câu phản hồi, sau đó trả về response được định sẵn như “Sorry, I don’t know”.

Áp dụng nhiều độ đo để nhận xét đánh giá, từ đó có thể rút ra kết luận đáng tin cậy và tinh chỉnh mô hình cho phù hợp.

7.1 Trong giai đoạn phát triển chatbot:

Sử dụng các độ đo confusion matrix, accuracy, precision và recall để đánh giá mức độ nhận dạng thể loại/ ý định từ câu input của người dùng.

VD: “What is good to eat” thuộc về category: food

Thiết kế một bộ test riêng gồm các câu hỏi và câu trả lời mong muốn, sau đó đưa vào chatbot để kiểm tra mức độ chính xác

Sử dụng phương pháp k-fold cross validation, chia bộ ngữ liệu thành k phần với phân bố dữ liệu là ngẫu nhiên và đều (không bị lệch, các câu hỏi trải đều). Dùng một phần để kiểm tra và các phần còn lại để huấn luyện. Lặp lại để mỗi phần đóng vai trò tập test một lần. Cuối cùng tính toán độ chính xác trung bình làm kết quả.

7.2 Trong giai đoạn production:

User metrics:

- Engaged users: mức độ sẵn sàng sử dụng lại chatbot của người dùng.
- Chat volume: mức độ những lần tương tác thành công, cuộc đối thoại càng dài và tần suất sử dụng chatbot càng nhiều thì chat volume càng cao
- Completion rate: độ đo Goal completion rate đánh giá mức độ thành công của hành động mà chatbot thực hiện để giải quyết một công việc
- Bounce rate: đo số lượng người xem và số người thực sự tương tác với chatbot

Conversation metrics:

- Conversation duration: độ dài cuộc đối thoại ở mức hợp lý, đủ ngắn để không gây chán nản, đủ dài để giải quyết yêu cầu của người dùng.
- Interaction rate: đo độ tương tác, số lần trao đổi tin nhắn

- Fallback rate: số lần mà chatbot không hiểu yêu cầu của người dùng và thực hiện giải pháp thay thế (phản hồi: “Sorry, I don’t know”)

Customer satisfaction metrics:

- Retention rate: Phần trăm số lượng người dùng sử dụng chatbot trở lại sau một thời gian
- Satisfaction score: Độ hài lòng, có thể là “nhờ” người dùng đánh giá hoặc dùng các đánh giá như cho số sao (star rating), bày tỏ cảm xúc (emoji rating)

8. Demo

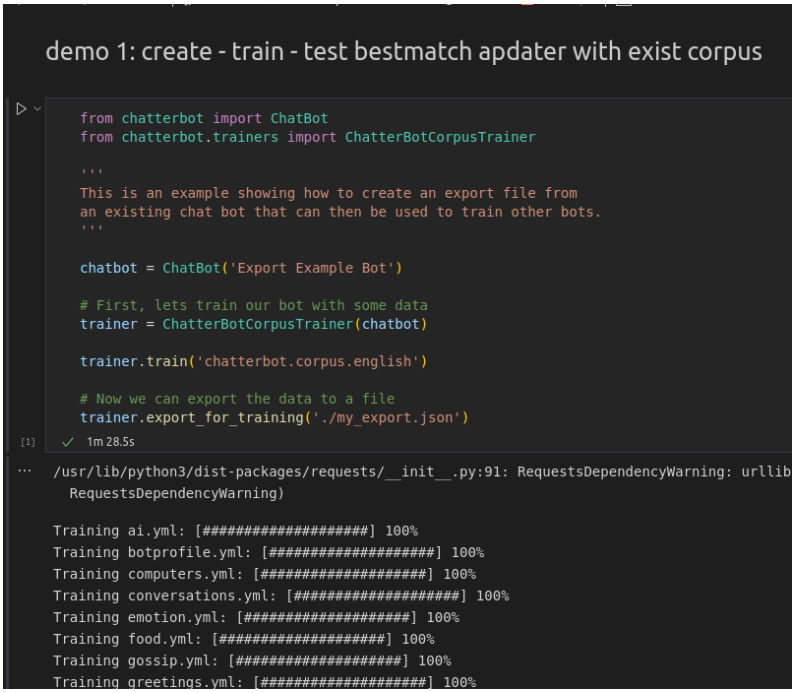
8.1 Django web app

Link github: https://github.com/PhmNm/django_chatterbot_final_nlp_application_hcmus

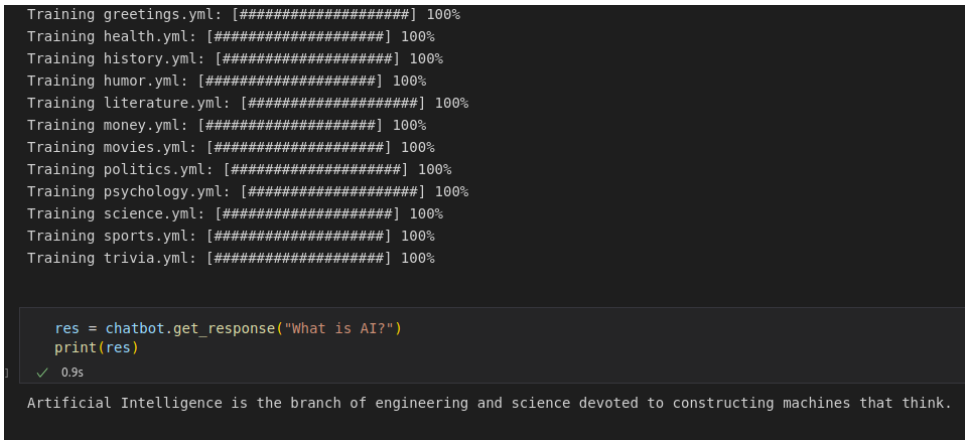
8.2 Demo python code

Xem file test.ipynb trong đường dẫn trên

8.2.1 Demo 1: create – train – test BestMatch adapter with exist corpus

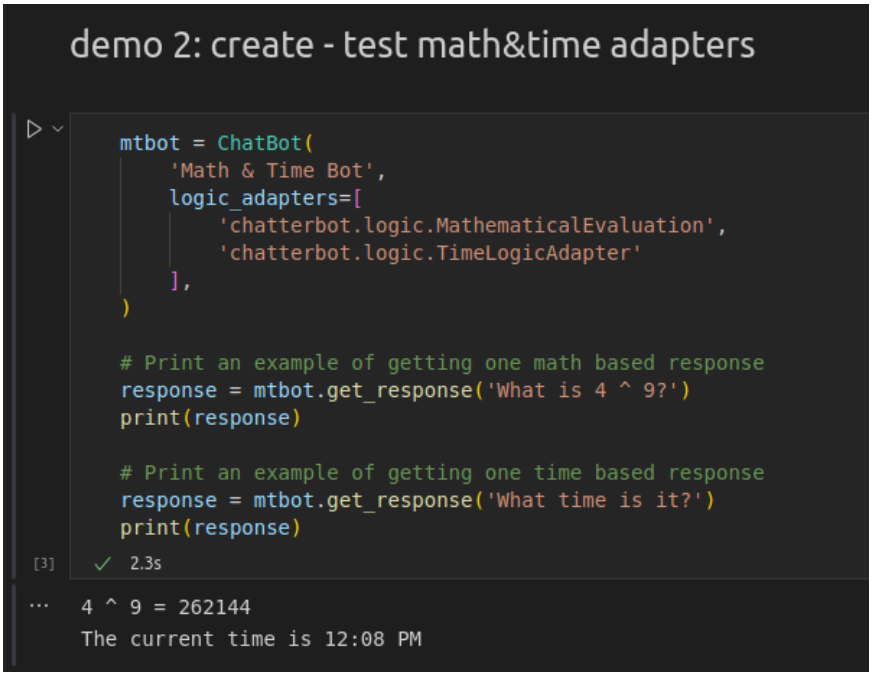


Hình 8.1: Create – train – test demo1



Hình 8.2: Test demo1

8.2.2 Demo 2: create – test math & time adapter

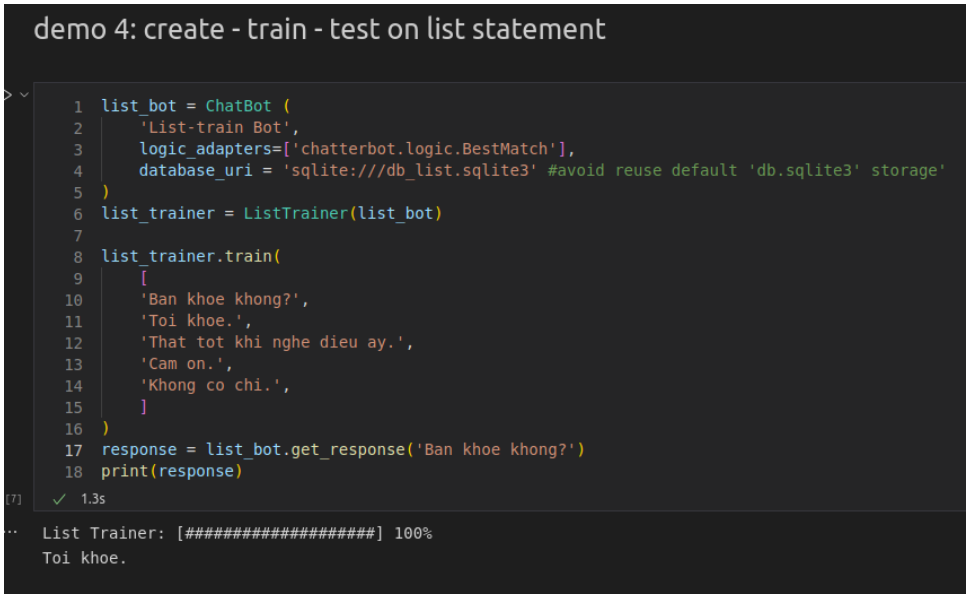


Hình 8.3: Create & test demo2

8.2.3 Demo 3: create – test with exist database

Hình 8.4: Create & test demo3

8.2.4 Demo 4: train on list



Hình 8.5: Create & train & test demo4

9. References

[1] "A Review Paper on Human Computer Interaction," *International Journal of Emerging Technologies and Innovative Research*, 2019.

[2] L. Dictionaries, Definition of chatbot in english, 2019.

[3] A. Khanna, B. Pandey, K. Vashishta, K. Kalia, P. Bhale and T. Das, "A study of today's A.I. through chatbots and rediscovery of machine intelligence," *International Journal of U- and e-Service, Science and Technology*, 2015.

[4] Gunthercox, "Release 1.0.8 · gunthercox/ChatterBot," 23 Aug 2020. [Online]. Available: https://github.com/gunthercox/ChatterBot/releases/tag/1.0.8.

[5] Gunthercox, "Python String Encoding," 9 Nov 2018. [Online]. Available: https://chatterbot.readthedocs.io/en/stable/encoding.html#does-chatterbot-handle-non-ascii-characters.

END