

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP. HỒ CHÍ MINH**

**CÔNG TRÌNH DỰ THI
GIẢI THƯỞNG “SINH VIÊN NGHIÊN CỨU KHOA HỌC”
NĂM 2020**

Tên công trình
**CodEbot – Một hệ thống chatbot sử dụng ngôn ngữ Tiếng
Việt để giải đáp thắc mắc trong lập trình cơ bản
với C++ và Python 3**

**Nhóm ngành Công Nghệ Thông Tin
Sinh viên thực hiện
Sinh viên 1: Vương Lê Minh Nguyên
MSSV: 43.01.104.117
Sinh viên 2: Lương Công Tâm
MSSV: 43.01.104.157
Khóa 43
Khoa Công Nghệ Thông Tin**

Giảng viên hướng dẫn: TS. Nguyễn Viết Hưng

TP.HCM, 4/2020

Trường Đại học Sư Phạm Thành phố Hồ Chí Minh

Khoa Công Nghệ Thông Tin



BÁO CÁO ĐỀ TÀI NGHIÊN CỨU KHOA HỌC SINH VIÊN

CodEbot – Một hệ thống chatbot sử dụng ngôn ngữ Tiếng Việt để giải đáp thắc mắc trong lập trình cơ bản với C++ và Python 3

Nhóm sinh viên thực hiện

VƯƠNG LÊ MINH NGUYỄN - 43.01.104.117

LƯƠNG CÔNG TÂM - 43.01.104.157

Giảng viên hướng dẫn

TS. NGUYỄN VIỆT HÙNG

MỤC LỤC

MỤC LỤC.....	3
SƠ ĐỒ	5
HÌNH ẢNH	5
SƠ LƯỢC.....	6
CHƯƠNG 1 - GIỚI THIỆU ĐỀ TÀI.....	7
1.1 - CodEbot – Trợ thủ học lập trình	7
1.2 - Mục tiêu đề tài.....	10
1.3 - Giới hạn đề tài	11
a - Giới hạn về kiến thức	11
b - Giới hạn về dữ liệu.....	12
CHƯƠNG 2 - TÌNH HÌNH NGHIÊN CỨU	13
2.1 - Tổng quan.....	13
2.2 - Tình hình nghiên cứu hệ thống chatbot Tiếng Việt	14
CHƯƠNG 3 - PHƯƠNG PHÁP THỰC HIỆN.....	15
3.1 - Thu thập dữ liệu	15
a - Kho ngữ liệu (Corpus)	15
b - Dữ liệu câu hỏi về lập trình.....	15
3.2 - Hệ thống Chatbot giải đáp các vấn đề về lập trình C++ và lập trình Python cơ bản .	16
a - Tổng quan hệ thống chatbot.....	16
b - Tiền xử lý và trích xuất đặc trưng.....	16
c - Support Vector Machine	18
d - Mô hình Linear SVM dự đoán ý định người dùng (user intent classification)	19
e - Mô hình Linear SVM dự đoán ngữ cảnh đối thoại (context classification)	20
f - Retrieval-based answering.....	20
3.3 - Ứng dụng CodEbot.....	21
a - Tổng quan ứng dụng	21

b - Giao diện chương trình	21
CHƯƠNG 4 - KẾT QUẢ - ĐÁNH GIÁ.....	24
CHƯƠNG 5 - TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN	25
5.1 - Tổng kết.....	25
5.2 - Hướng phát triển.....	25
TÀI LIỆU THAM KHẢO.....	26

SƠ ĐỒ

Sơ đồ 1 Tổng quan hệ thống	16
Sơ đồ 2 Quá trình tiền xử lý và trích xuất đặc trưng.....	16

HÌNH ẢNH

Hình 1.1 Kỷ nguyên Công nghiệp 4.0	7
Hình 1.2 Sân chơi công nghệ dành cho trẻ em được tổ chức bởi Học viện sáng tạo công nghệ Teky.....	8
Hình 1.3 Ngôn ngữ lập trình C++ và một số ứng dụng	9
Hình 1.4 Ngôn ngữ lập trình Python và một số ứng dụng	9
Hình 2.1 Trợ lý ảo Google Assistant	13
Hình 3.1 Một đoạn ngữ liệu về lập trình Python được lấy từ Wikipedia	15
Hình 3.4 Giao diện PC – CodEbot DEMO	21
Hình 3.5 Giao diện trên di động – CodEbot DEMO.....	22
Hình 3.6 Danh sách tri thức - Knowledge Management.....	22
Hình 3.7 Giao diện cập nhật chi thức - Knowledge Management.....	23

SƠ LƯỢC

Việt Nam đang hướng tới mục tiêu trở thành một nước công nghiệp để cùng hòa nhập với thế giới trong thời đại Công nghiệp 4.0, chính vì thế kỹ năng lập trình cũng được nhìn nhận là một trong những kỹ năng thiết yếu của giới trẻ trong thời đại này, và kế hoạch giáo dục cũng đang dần được thay đổi, chú trọng hơn trong bộ môn tin học và lập trình từ cấp tiểu học và trung học cơ sở. Đối với học sinh, thời gian học tập một môn trên trường lớp không thực sự nhiều mà đặc biệt trong lĩnh vực lập trình thì kiến thức lại rất sâu và rộng, sách giáo khoa chỉ có thể tóm tắt sơ lược, và tài liệu trên mạng thì thông thường lại là Tiếng Anh, hay Tiếng Việt nhưng nội dung thì lan man, khó hiểu đối với học sinh. Nắm bắt được thực trạng và thấu hiểu được khó khăn của những người mới học lập trình, đề tài này hướng đến mục tiêu thiết kế và xây dựng thử nghiệm một hệ thống Chatbot Tiếng Việt để giải thắc mắc trong lập trình C++ và lập trình Python cơ bản sử dụng kết hợp công nghệ Web và công nghệ Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP). Dữ liệu huấn luyện mô hình NLP do nhóm tự thu thập và cơ sở dữ liệu các câu trả lời cũng được tổng hợp và phiên dịch từ các nguồn kiến thức có thể tin cậy.



- **Đề tài:** CodEbot - Một hệ thống chatbot sử dụng ngôn ngữ Tiếng Việt để giải đáp thắc mắc trong lập trình C++ và Python cơ bản
- **Nhóm tác giả:**
 1. Vương Lê Minh Nguyên
 2. Lương Công Tâm
- **Hướng dẫn:** TS. Nguyễn Việt Hưng

Designed by **Tam Luong**



CHƯƠNG 1 - GIỚI THIỆU ĐỀ TÀI

1.1 - CodEbot – Trợ thủ học lập trình

Trong kỷ nguyên Công nghiệp 4.0, các công việc của con người đang dần được chuyển đổi từ sức người sang sử dụng máy móc qua đó gia tăng năng suất và giảm thiểu các rủi ro, tai nạn lao động không mong muốn. Quá trình chuyển đổi đó có tác động rất lớn đến công ăn việc làm của con người, điển hình là rất nhiều ngành nghề đã loại bớt sức lao động của con người bằng các cỗ máy tự động có thể thực hiện công việc đó với năng suất vượt trội hơn. Tuy nhiên, điều đó không có nghĩa là con người sẽ mất đi công việc của mình, mà thay vào đó, vai trò của con người sẽ trở thành người giám sát và vận hành những cỗ máy đó, đặc biệt là vai trò nghiên cứu, chế tạo và phát triển các thiết bị tự động hóa đó. Để đảm nhiệm tốt vai trò trên, con người ở thời đại này cần phải có tư duy giải quyết vấn đề, kiến thức về máy tính và kỹ năng lập trình là rất cần thiết.



Hình 1.1 Kỷ nguyên Công nghiệp 4.0

Việt Nam đang từng bước thực hiện “công nghiệp hóa, hiện đại hóa” với mục tiêu trở thành một quốc gia công nghiệp. Mặc dù là một đất nước vừa thoát khỏi chiến tranh không lâu, nhưng Việt Nam vẫn luôn giữ vững đà phát triển trong mọi lĩnh vực và không ngần ngại tham gia vào cuộc Cách mạng Công nghiệp 4.0 cùng với các nước trên thế giới. Điển hình là những năm gần đây, Việt Nam ra sức vận động, thúc đẩy phát triển các lĩnh vực Công nghệ thông tin và truyền thông, hay trong lĩnh vực giáo dục đã có sự thay đổi rất lớn về chương trình học, đặc

biệt quan tâm đến bộ môn tin học và lập trình. Cụ thể trong chương trình giáo dục phổ thông bộ môn tin học mới, học sinh sẽ được tiếp xúc và làm quen với máy tính từ cấp tiểu học, được rèn luyện tư duy và lập trình ở các cấp trung học cơ sở và phổ thông. Qua đó có thể thấy được rằng Việt Nam đang rất chú trọng trong việc đào tạo kỹ năng công nghệ thông tin, tư duy giải quyết vấn đề và kỹ năng lập trình cho giới trẻ của thời đại mới.

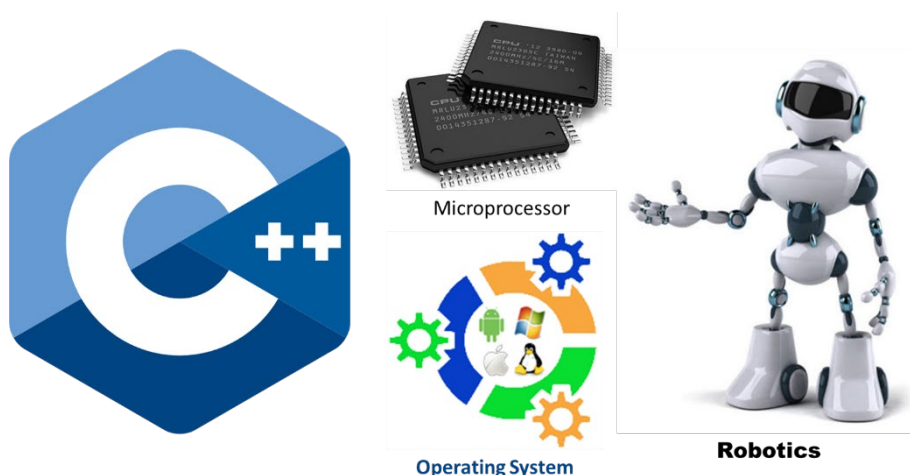


Hình 1.2 Sân chơi công nghệ dành cho trẻ em được tổ chức bởi Học viện sáng tạo công nghệ Teky

Tư duy giải quyết vấn đề và kỹ năng lập trình là 2 tố chất quan trọng của con người trong thời đại mới. Trong đó, lập trình là phương thức cơ bản nhất để con người giao tiếp với máy tính, từ đó chỉ định, hướng dẫn máy tính thực hiện theo từng bước cụ thể để hoàn thành một công việc cụ thể. Máy tính chỉ có thể hiểu và thực thi ngôn ngữ máy, tức một dãy số chỉ gồm 2 chữ số 0 và 1, rất khó để con người có thể học và giao tiếp trực tiếp bằng loại ngôn ngữ này. Qua thời gian, việc lập trình đã trở nên đơn giản hơn rất nhiều sau khi các ngôn ngữ lập trình sử dụng các ký tự latin từ ngôn ngữ tự nhiên để tạo thành các câu lệnh mà con người có thể hiểu được đã được ra đời, thời gian đầu là các ngôn ngữ lập trình bậc thấp hay hợp ngữ (Assembly, MIP, ...), cú pháp của các ngôn ngữ lập trình này vẫn còn rất gần với ngôn ngữ máy, rất khó học và sử dụng các ngôn ngữ này để lập trình. Tuy nhiên, chúng vẫn được sử dụng rất nhiều, đặc biệt là trong lĩnh vực vi điều khiển. Tiếp tục với sự phát triển của các ngôn ngữ lập trình, các ngôn ngữ lập trình bậc cao ra đời với cú pháp gần với ngôn ngữ tự nhiên nhất và được biên dịch hoặc thông dịch sang ngôn ngữ máy để máy tính có thể hiểu được. Điển hình trong số đó là ngôn ngữ C++ và Python.

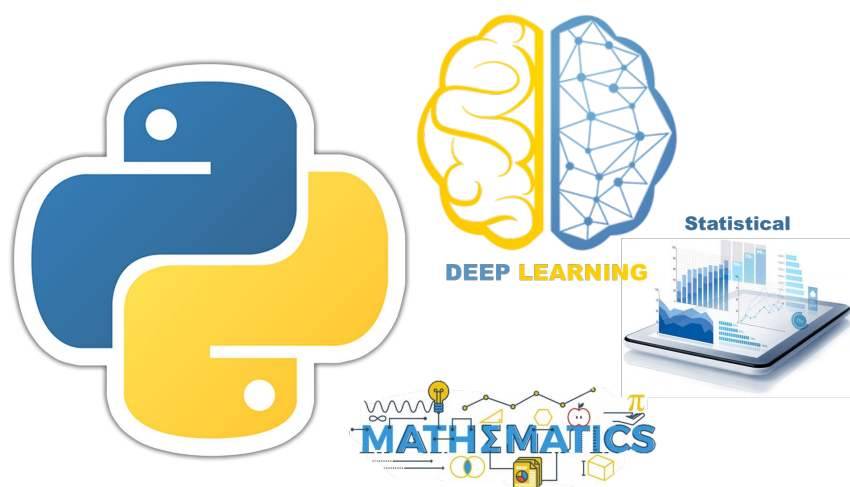
Ngôn ngữ lập trình C++ ra mắt vào năm 1985, thiết kế bởi Bjarne Stroustrup, là ngôn ngữ được phát triển dựa trên ngôn ngữ C, hay có thể nói là một phần mở rộng của C. Với khả năng lập trình tổng quát, lập trình thủ tục, lập trình hướng đối tượng và đặc biệt đi kèm với các công cụ để thao tác với bộ nhớ cấp thấp (low-level memory), C++ thường được sử dụng trong

lập trình hệ thống, lập trình vi điều khiển, các ứng dụng cần tối ưu bộ nhớ, ... C/C++ thường được sử dụng trong giảng dạy nhằm giúp cho người học có thể hiểu rõ hơn về các kiểu dữ liệu, kiến trúc hệ thống hay hiểu rõ hơn về cách hoạt động của các thuật toán bằng việc tự mình cài đặt các thuật toán.



Hình 1.3 Ngôn ngữ lập trình C++ và một số ứng dụng

Ngôn ngữ lập trình Python được tạo ra bởi Guido van Rossum và ra mắt vào năm 1991. Python được biết đến nhờ cú pháp đơn giản của nó, một chương trình được viết bằng Python sẽ ngắn hơn rất nhiều so với viết bằng các ngôn ngữ khác. Python được sử dụng nhiều trong lĩnh vực khoa học máy tính nhờ vào cú pháp đơn giản và nhiều gói công cụ có sẵn để làm việc với các phép toán thống kê, phép toán ma trận hay các công cụ biểu diễn dữ liệu trên đồ thị, ... Trong các lĩnh vực ngoài công nghệ thông tin, Python cũng được nhiều nhà khoa học dữ liệu lựa chọn làm công cụ hỗ trợ nghiên cứu.



Hình 1.4 Ngôn ngữ lập trình Python và một số ứng dụng

Số lượng kiến thức trong lĩnh vực lập trình có thể nói là rất lớn, ngay cả với những kiến thức cơ bản của lập trình. Thế nhưng số tiết học tin học tại trường của học sinh trung học cơ sở và trung học phổ thông chỉ bao gồm 70 tiết/lớp/năm theo kế hoạch đào tạo phổ thông mới, ngoài ra các em còn phải học thêm các kiến thức khác không liên quan đến lập trình, vì thế mà lượng kiến thức và thời gian thực hành của học sinh trong trường rất ít, điều này cũng là một trở ngại rất lớn đối với những học sinh có đam mê và định hướng theo lập trình. Có thể nói ngoài việc học tập trên trường, học sinh có thể tìm kiếm thêm tài liệu về lập trình qua Internet, thế nhưng kiến thức trên Internet lại quá nhiều nhưng lại không được tổng hợp, các tài liệu chủ yếu bằng Tiếng Anh, thường nói chuyên sâu, khó hiểu hay nói lan man gây khó khăn cho những người mới học lập trình.

CodEbot – Code Education chatBot – là sáng kiến từ sự nắm bắt tình hình và sự đồng cảm nhờ vào kinh nghiệm và những trải nghiệm khó khăn của nhóm tác giả trong khoảng thời gian mới bắt đầu học lập trình. CodEbot là đề tài với mục tiêu thiết kế và xây dựng thử nghiệm một hệ thống chatbot sử dụng Tiếng Việt giúp giải đáp các vấn đề cơ bản trong lập trình với ngôn ngữ C++ và ngôn ngữ Python. Hệ thống sẽ bao gồm một chatbot ứng dụng công nghệ Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) để phân tích đoạn hội thoại Tiếng Việt nhằm đưa ra phản hồi tương ứng, và phần giao diện ứng dụng để giao tiếp với chatbot.

Dữ liệu huấn luyện mô hình NLP do nhóm tự thu thập và cơ sở dữ liệu các câu trả lời cũng được tổng hợp và phiên dịch từ các nguồn kiến thức có thể tin cậy.

CodEbot hướng tới các bạn học sinh ở các cấp trung học cơ sở và trung học phổ thông của Việt Nam trong thời đại mới này, hay những người có niềm đam mê với lập trình và mới bắt đầu học lập trình. CodEbot mong muốn sẽ trở thành hành trang, thành công cụ học tập cho các bạn học sinh trong quá trình chinh phục môn lập trình của các bạn.

1.2 - Mục tiêu đề tài

Các nghiên cứu về xử lý ngôn ngữ tự nhiên, ngôn ngữ Tiếng Việt gần đây được nhóm nghiên cứu quan tâm và tham gia, tuy nhiên vẫn còn rất nhiều hạn chế và chưa hoàn thiện cùng với lượng kiến thức khổng lồ trong lĩnh vực lập trình là những trở ngại chính của CodEbot, các trở ngại đó khó có thể giải quyết tất cả chỉ trong một đề tài nghiên cứu, vì vậy mà nhóm

nghiên cứu đã giới hạn các mục tiêu cần đạt được trong đề tài này, và trong tương lai sẽ tiếp tục giải quyết các khó khăn còn tồn đọng để hoàn thiện CodEbot.

Những mục tiêu mà nhóm nghiên cứu muốn đạt được trong đề tài này bao gồm:

- Một mô hình máy học xử lý ngôn ngữ tự nhiên Tiếng Việt, dùng để phân tích ý định của người dùng qua đoạn đối thoại.
- Một hệ thống giải đáp thắc mắc về các kiến thức liên quan đến lập trình cơ bản, dựa trên ý định của người dùng và ngữ cảnh của đối thoại.
- Kết hợp 2 hệ thống trên thành một hệ thống chatbot Tiếng Việt để giải đáp các vấn đề liên quan đến lập trình cơ bản trong 2 ngôn ngữ lập trình là C++ và Python.
- Xây dựng một ứng dụng giao diện trên nền tảng Web để người dùng có thể tương tác với hệ thống chatbot trên mọi thiết bị.

Về cơ bản, các mục tiêu trên có thể sẽ không đạt được độ chính xác cao, nhưng nhóm nghiên cứu vẫn hy vọng có thể cải thiện CodEbot hơn trong tương lai.

1.3 - Giới hạn đề tài

a - Giới hạn về kiến thức

CodEbot là một chatbot giải đáp kiến thức về lập trình, tuy nhiên lập trình nói chung là một lĩnh vực vô cùng rộng lớn, ngoài kiến thức về các ngôn ngữ lập trình và cú pháp ngôn ngữ mà lập trình còn bao gồm một số kiến thức về toán học và kiến thức về kiến trúc máy tính, ... Đề tài này không thể bao quát hết lượng kiến thức khổng lồ này, nên nhóm nghiên cứu đã giới hạn lại lượng kiến thức mà CodEbot có thể trả lời. Cụ thể:

- **Về ngôn ngữ lập trình:** 2 ngôn ngữ là C++ và Python bởi vì đây là 2 ngôn ngữ phổ biến, cả trong nền công nghiệp lẫn trong lĩnh vực giáo dục. Ở các cấp trung học cơ sở và trung học phổ thông tại Việt Nam trước đây vẫn dạy ngôn ngữ Turbo Pascal, tuy nhiên nhóm nhận thấy ngôn ngữ này đã không còn phổ biến, và cũng có thể sẽ bị loại bỏ trong chương trình giáo dục phổ thông mới nên đã không lựa chọn ngôn ngữ này.
- **Về các dạng kiến thức:** Do giới hạn về thời gian và chi phí, cùng với lượng dữ liệu mà nhóm tổng hợp được tương đối nhỏ, nên trong đề tài này CodEbot chỉ có thể trả lời các dạng câu hỏi định nghĩa, so sánh và ứng dụng.

- **Tương tác ngoài lề:** Ngoài trả lời các câu hỏi về lập trình, người dùng còn có thể tương tác với CodEbot về các vấn đề không liên quan như chào hỏi, hỏi xin tài liệu lập trình hoặc mẹo lập trình, ...

b - Giới hạn về dữ liệu

Đề tài này gồm 2 loại dữ liệu là dữ liệu huấn luyện mô hình NLP (Corpus) và dữ liệu câu hỏi về lập trình.

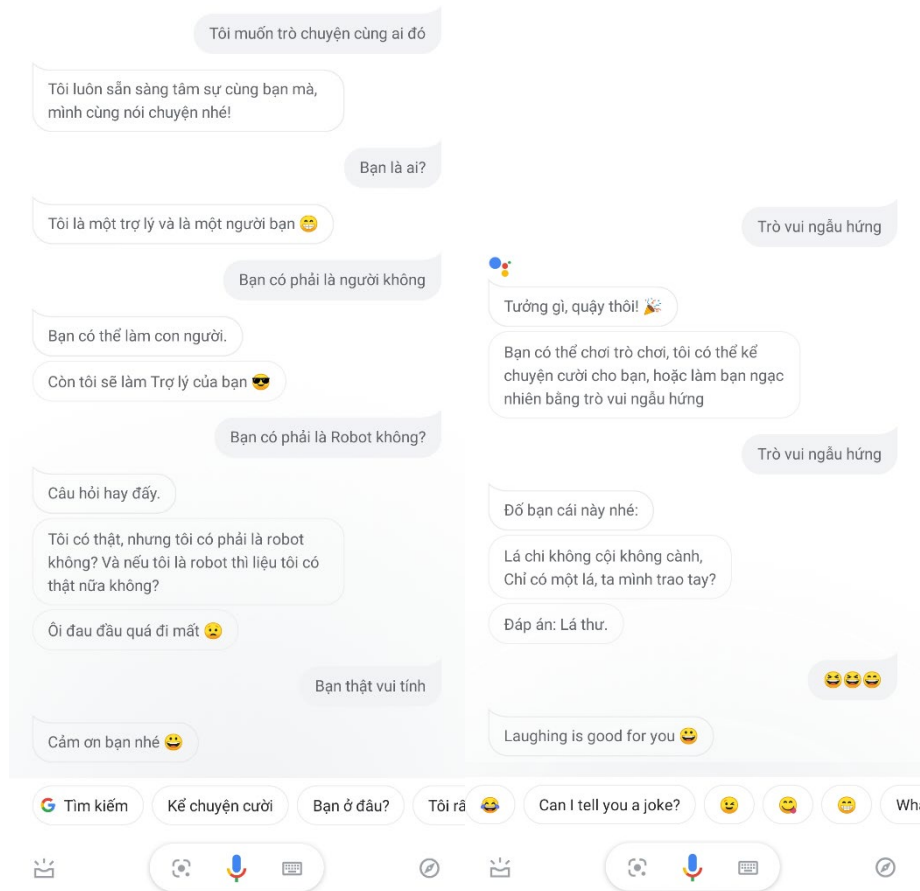
Trong các tập Corpus của các công trình nghiên cứu NLP ngôn ngữ Tiếng Việt trước đây không hề có các thông tin, kiến thức về lập trình nên những tập (Corpus) này hoàn toàn không phù hợp với bài toán đặt ra của đề tài này. Chính vì vậy nhóm nghiên cứu đã tự đi thu thập dữ liệu cho bài toán của đề tài, và do hạn chế về thời gian thực hiện nên lượng dữ liệu mà nhóm thu thập được không nhiều, chủ yếu là các dữ liệu văn bản về lập trình, tuy nhiên vẫn đảm bảo được độ chính xác cao của mô hình NLP trong phạm vi giới hạn kiến thức của đề tài.

Dữ liệu hỏi đáp về lập trình là các câu trả lời, là kiến thức mà CodEbot sẽ trả về cho người dùng tương ứng với thắc mắc của người dùng. Để tổng hợp các kiến thức về lập trình, nhóm nghiên cứu phải tổng hợp, phiên dịch và tóm tắt lại các kiến thức từ các nguồn có thể tin cậy như sách giáo khoa, giáo trình lập trình cơ bản, tài liệu gốc tùy của ngôn ngữ lập trình, ... Việc tổng hợp và tóm tắt lại các kiến thức cũng dựa phụ thuộc rất nhiều vào kiến thức của nhóm nghiên cứu và thời gian thực hiện đề tài có giới hạn nên dữ liệu thu thập được không nhiều, nên CodEbot sẽ không trả về kiến thức mong muốn.

CHƯƠNG 2 - TÌNH HÌNH NGHIÊN CỨU

2.1 - Tổng quan

Cùng với sự gia tăng tương tác người - máy (HCI), sự ra đời và phát triển của các thiết bị phần cứng hay phần mềm có khả năng giao tiếp với con người thông qua ngôn ngữ tự nhiên là một điều tất yếu về công nghệ hiện nay. Những thiết bị hay phần mềm có giao diện người dùng hỗ trợ giao tiếp qua ngôn ngữ tự nhiên như vậy được gọi chung là chatbot [1] [2].



Hình 2.1 Trợ lý ảo Google Assistant

Trong hội nghị Spring 2016, Facebook và Microsoft đã chính thức cung cấp các tài nguyên để xây dựng chatbots tích hợp vào nền tảng nhắn tin của họ là Messenger và Skype. Theo một công bố chính thức từ bài phỏng vấn của David Marcus, người đứng đầu mảng tin nhắn của Facebook, chỉ 6 tháng sau khi Facebook cung cấp tài nguyên để xây dựng chatbots trên nền tảng Messenger đã có đến khoảng 33000 nhà phát triển xây dựng khoảng 34000 chatbots dù chất lượng của chúng chưa thực sự tốt. Ngoài ra những nền tảng tin nhắn khác như Slack, Kik, Viber, ... cũng đã hỗ trợ sự phát triển của chatbots.

Để có thể giao tiếp được với con người, chatbots cần phải trải qua quá trình xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) để có thể đưa ra phản hồi tương ứng và có nghĩa cho con người. NLP xuất hiện từ rất sớm trong ngành khoa học máy tính. Từ những năm 1960, Weizenbaum đã xuất bản một nghiên cứu sáng tạo về tương tác ngôn ngữ tự nhiên với ELIZA, một chương trình máy tính được phát triển để bắt chước các phản ứng của nhà trị liệu tâm lý trong một buổi trị liệu tâm lý [3]. Những tiến bộ vượt bậc những năm gần đây của Machine Learning hứa hẹn những cải tiến đáng kể trong khả năng diễn giải và dự đoán ngôn ngữ tự nhiên [4]. Các mô hình dự đoán trong NLP như Recurrent Neural Network (RNN) và Sequence-to-sequence ngày càng trở nên vượt trội hơn so với các mô hình luật dẫn (rule-based) truyền thống [5]. Tổng thể những tiến bộ về NLP trong thời gian gần đây đã nâng hiệu quả và chất lượng của chatbots lên một tầm cao mới. Ngoài ra, giải thưởng Loebner trong lĩnh vực trí tuệ nhân tạo cũng đóng góp không nhỏ thúc đẩy sự phát triển của NLP.

Chatbots đang ngày càng hiện diện trong nhiều lĩnh vực như y tế, các ngành sản xuất, dịch vụ và trong lĩnh vực giáo dục [10][11]. Trong lĩnh vực giáo dục, chatbot được xây dựng nhằm hỗ trợ học sinh, sinh viên truy cập các tài liệu về bài giảng, hay giúp giảng viên thu thập ý kiến của sinh viên về buổi học. Bên cạnh đó, các hệ trả lời câu hỏi dựa trên cơ sở tri thức (Knowledge-based Question Answering – KBQA) cũng đạt được nhiều bước tiến gần đây [6] [7] [8] [9]. Những bước tiến này góp phần hoàn thiện các chatbots giải đáp trả lời câu hỏi đặc biệt trong lĩnh vực giáo dục.

2.2 - Tình hình nghiên cứu hệ thống chatbot Tiếng Việt

Lĩnh vực nghiên cứu về NLP Tiếng Việt hiện đang là một lĩnh vực rất được quan tâm tại Việt Nam. Nhờ vào sự đóng góp của cộng đồng hay các tổ chức trong việc xây dựng các bộ dữ liệu Tiếng Việt, mà quá trình nghiên cứu được tiến triển mạnh mẽ hơn. Ngoài ra còn có các cuộc thi về xây dựng các mô hình xử lý ngôn ngữ Tiếng Việt nhằm tiếp thêm động lực cho các nhà nghiên cứu. Tuy nhiên, các nghiên cứu về xây dựng chatbot hỗ trợ Tiếng Việt vẫn còn hạn chế, đặc biệt là chatbots trong lĩnh vực giáo dục. Hầu hết chatbots sử dụng Tiếng Việt đều được thiết kế cho mục đích thương mại điện tử, marketing, chăm sóc khách hàng, ... Chính vì sự khan hiếm của những nghiên cứu đi trước về một hệ trả lời câu hỏi bằng Tiếng Việt nên đây cũng là một khó khăn lớn cần phải vượt qua của đề tài này.

CHƯƠNG 3 - PHƯƠNG PHÁP THỰC HIỆN

3.1 - Thu thập dữ liệu

a - Kho ngữ liệu (Corpus)

Trong đề tài này, chúng tôi tiến hành thu thập các văn bản ngôn ngữ tự nhiên về đề tài lập trình C++ và Python trên các trang Wikipedia để tổng hợp thành một kho ngữ liệu lập trình C++ và Python. Việc xây dựng kho ngữ liệu này nhằm mục đích phục vụ cho thuật toán TF-IDF, một thuật toán thống kê thường được áp dụng như là một phương pháp word-embedding

để ánh xạ từ ngữ, câu, văn bản trong ngôn ngữ tự nhiên sang một vector các số thực để có thể tính toán được bằng các mô hình máy học (chi tiết tại mục 3.2.b).

Ngữ liệu được tổ chức dưới dạng các văn bản thuần túy để khi đưa vào xây dựng bảng IDF trong thuật toán TF-IDF sẽ được đọc theo từng câu, mỗi câu ứng với một đơn vị *document* trong bảng IDF.

Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

Lịch sử
Sự phát triển Python đến nay có thể chia làm các giai đoạn:

Python 1: bao gồm các bản phát hành 1.x. Giai đoạn này, kéo dài từ đầu đến cuối thập niên 1990. Từ năm 1990 đến 1995. Guido làm việc tại CWI (Centrum voor

Hình 3.1 Một đoạn ngữ liệu về lập trình Python được lấy từ Wikipedia

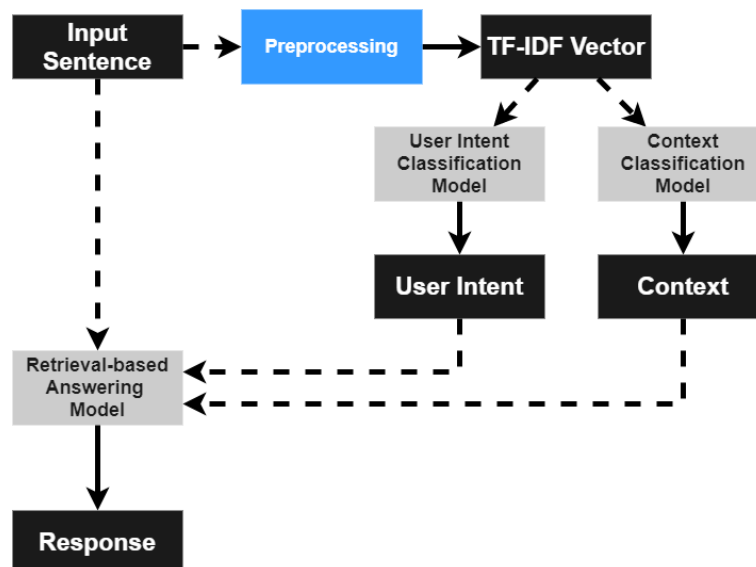
b - Dữ liệu câu hỏi về lập trình

Bên cạnh việc xây dựng kho ngữ liệu, chúng tôi còn tiến hành tự xây dựng một tập dữ liệu có cấu trúc để phục vụ cho mô hình máy học nhằm phân loại ý định của người dùng (User Intent Classification) và phân loại ngữ cảnh (Context Classification). Phần lớn dữ liệu được thu thập từ các sinh viên khoa Công Nghệ Thông Tin trường Đại học Sư phạm TP.HCM cũng như từ một số diễn đàn lập trình và nhóm học lập trình trên Facebook. Một phần nhỏ của dữ liệu do chúng tôi tự thêm vào để tinh chỉnh dữ liệu học của các mô hình máy học theo ý đồ.

Dữ liệu được tổ chức thành một tập các bộ gồm 1000 câu hỏi hoặc câu nói cùng với ngữ cảnh và ý định tương ứng của người dùng. Câu hỏi hoặc câu nói chính là dữ liệu đầu vào cho các mô hình máy học. Ngữ cảnh hay ý định tương ứng với mỗi câu sẽ là nhãn cho 2 mô hình máy học (mô hình dự đoán ngữ cảnh và mô hình dự đoán ý định).

3.2 - Hệ thống Chatbot giải đáp các vấn đề về lập trình C++ và lập trình Python cơ bản

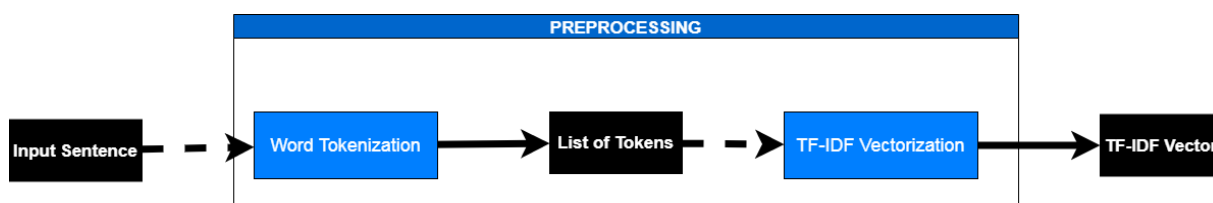
a - Tổng quan hệ thống chatbot



Sơ đồ 1 Tổng quan hệ thống

Dữ liệu đầu vào của hệ thống chatbot là một đoạn hội thoại bằng tiếng Việt, và dữ liệu đầu ra của hệ thống là một phản hồi tới người dùng trong ngôn ngữ tự nhiên. Câu hỏi (câu nói) đầu vào sẽ được rút trích đặc trưng và phân loại ý định cũng như ngữ cảnh bằng hai mô hình máy học, sau đó cả câu gốc (ở dạng văn bản thuần) và ý định cũng như ngữ cảnh mà hai mô hình máy học trả về sẽ được cung cấp cho một mô hình trả lời câu hỏi retrieval-based để tìm ra phản hồi tương ứng cho người dùng.

b - Tiền xử lý và trích xuất đặc trưng



Sơ đồ 2 Quá trình tiền xử lý và trích xuất đặc trưng

Dữ liệu đầu vào ở dạng ngôn ngữ tự nhiên không thể trực tiếp đưa vào mô hình máy học, mà phải thông qua quá trình word-embedding để biến văn bản trong ngôn ngữ tự nhiên thành một vector các số thực.

Trước khi đưa dữ liệu ngôn ngữ tự nhiên vào mô hình word-embedding, chúng tôi tiến hành quá trình phân đoạn từ (word segmentation – word tokenization) để nâng cao kết quả huấn luyện và dự đoán của các mô hình máy học. Đối với các nghiên cứu về ngôn ngữ tự nhiên trước đây dành cho ngôn ngữ Anh, quá trình phân đoạn từ đối với Tiếng Anh thực sự không quá khó, vì Tiếng Anh có rất ít từ ghép, và các thành phần của từ ghép đều được đặt cách nhau bởi dấu gạch nối, giúp cho công đoạn tokenization chỉ cần dựa vào khoảng trắng hoặc dấu câu là đủ đạt độ chính xác rất cao. Tuy nhiên, trong Tiếng Việt, một từ có thể được cấu thành từ nhiều tiếng và vẫn cách nhau bằng khoảng trắng thông thường. Nếu áp dụng phương pháp phân đoạn từ tương tự như trong Tiếng Anh, nhiều từ sẽ bị xé nhỏ thành những tiếng vô nghĩa, không có giá trị sử dụng trong mô hình máy học. Do đó, bài toán phân đoạn từ trong Tiếng Việt là một bài toán khó cần được giải quyết nghiêm túc để có thể nâng cao hiệu quả của các mô hình xử lý ngôn ngữ Tiếng Việt. Trong đề tài này, chúng tôi sử dụng mô hình word tokenization dành cho Tiếng Việt được xây dựng sẵn của nhóm **underthesea** [38] để giải quyết bài toán tách từ Tiếng Việt. Mô hình này sử dụng thuật toán học thống kê Conditional Random Field được huấn luyện sẵn trên các kho ngữ liệu có dán nhãn của **underthesea**.

Cùng với sự phát triển vượt bậc trong lĩnh vực xử lý ngôn ngữ tự nhiên, ngày nay, có rất nhiều mô hình word-embedding từ đơn giản đến phức tạp như Bag of Words, TF-IDF, word2vec, doc2vec, GloVe,... Mỗi mô hình đều có ưu nhược điểm riêng, và tất nhiên, những mô hình ra đời sau như word2vec, doc2vec hay GloVe đều dựa trên các mạng học sâu (Deep Learning Networks) nên có khả năng số hóa các từ hoặc câu một cách chuẩn xác, thể hiện được sự tương đồng hoặc đối lập về mặt ngữ nghĩa của các từ trong không gian đích ở dạng số thực. Tuy nhiên, khuyết điểm của những mô hình này là cần lượng dữ liệu rất lớn để có thể đạt được độ chính xác kỳ vọng trong một hệ thống chatbot. Do đó, trong đề tài này, nhận thấy dữ liệu thu thập được là quá ít cho các mô hình học sâu, chúng tôi quyết định sử dụng TF-IDF, một mô hình word-embedding đơn giản nhưng hiệu quả, làm mô hình word-embedding cho hệ thống chatbot này.

Mô hình TF-IDF gồm hai phần quan trọng cần phải xác định là TF – Term Frequency và IDF – Inverse Document Frequency.

TF của một token t trong văn bản d – $\mathbf{TF}(t, d)$ là tần số xuất hiện của token t trong văn bản đang xét d . TF của một token trong văn bản có nhiều phương pháp tính. Ở đây chúng tôi

chọn cách tính tương tự như Bag of Words, chỉ đếm số lần xuất hiện của token trong văn bản chứ không chia kết quả đếm cho tổng số từ của văn bản.

IDF của một token t – **IDF(t)** là nghịch đảo tần số xuất hiện của token đó trên toàn bộ các văn bản. IDF của một token cũng có nhiều cách tính. Ở đây chúng tôi sử dụng công thức sau để tính IDF cho mỗi token:

$$\text{IDF}(t) = \log \frac{N}{\text{DF}(t)} + 1$$

Với N là tổng số văn bản trong bộ dữ liệu, **DF(t)** là số văn bản có chứa token t trong bộ dữ liệu.

Sau khi có được TF và IDF của mỗi token trong văn bản, ta tiến hành xây dựng TF-IDF vector bằng cách tính giá trị TF-IDF cho từng token trong văn bản đang xét theo công thức sau:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

Với **TF(t, d)** và **IDF(t)** đã được giải thích ở trên.

Vậy, mỗi một câu nói (văn bản) trong ngôn ngữ tự nhiên có thể được chuyển thành một vector số thực với số tọa độ là số token trong từ điển dùng để xây dựng bảng IDF, và mỗi tọa độ chính là giá trị TF-IDF của token tương ứng.

c - Support Vector Machine

Support Vector Machine (SVM) là một thuật toán phân lớp nhị phân (binary classification), trong đó, tập dữ liệu ban đầu sẽ được chia thành hai lớp dữ liệu thông qua một siêu phẳng. Mô hình SVM sẽ tìm cách biểu diễn các điểm dữ liệu trong không gian và tìm kiếm hồi quy một siêu phẳng giữa hai lớp dữ liệu sao cho khoảng cách từ các điểm dữ liệu của tập huấn luyện tới siêu phẳng là xa nhất có thể. Các điểm dữ liệu của tập đánh giá cũng sẽ nằm trong cùng không gian với tập huấn luyện, và được SVM dự đoán thuộc một trong hai lớp dữ liệu tùy thuộc điểm đó nằm ở bên nào của siêu phẳng. Trong các thuật toán máy học phân loại dữ liệu không có cấu trúc, SVM cùng với các mô hình học sâu (Deep Learning models) được đánh giá khá cao về hiệu quả dự đoán, tức là dễ dàng đạt điểm cao trong nhiều evaluation metric mà không cần sử dụng quá nhiều kỹ thuật rút trích dữ liệu phức tạp. Tuy nhiên, vì kích

thước tập dữ liệu sử dụng cho nghiên cứu này là rất nhỏ để sử dụng các mô hình học sâu (kích thước dữ liệu khoảng 1000 bộ), SVM chính là lựa chọn thích hợp nhất về mọi mặt.

SVM vốn là một thuật toán phân lớp nhị phân, nhưng có thể được áp dụng để phân loại nhiều hơn hai lớp (multi-class classification). Để sử dụng mô hình SVM bài toán multi-class, người ta thường áp dụng một trong hai chiến thuật cơ bản One vs One (OVO) hoặc One vs the Rest (OVR). Ở đây chúng tôi chọn chiến lược One vs the Rest để phân loại ý định của người dùng hay ngữ cảnh của đối thoại từ câu hỏi hoặc câu nói của người dùng. Với chiến lược OVR, SVM sẽ tối ưu tham số từng siêu phẳng tương ứng chia cắt mỗi lớp cảm xúc với phần còn lại thông qua quá trình học trên tập huấn luyện. Dữ liệu cần dự đoán sẽ được lần lượt đưa vào các phương trình siêu phẳng và phân lớp dựa theo số điểm cao nhất của nó (voting) có được từ các phương trình siêu phẳng, từ đó đi đến một kết quả phân loại.

d - Mô hình Linear SVM dự đoán ý định người dùng (user intent classification)

Trong đề tài này, chúng tôi đề xuất chia ý định người dùng ra thành 11 lớp:

- Agree: người dùng đồng ý với câu hỏi xác nhận của chatbot
- Disagree: người không đồng ý với câu hỏi xác nhận của chatbot
- Credit Info: người dùng hỏi về thông tin cơ bản của chatbot (tác giả, ...)
- Greeting: người dùng chào hỏi chatbot
- Help: người dùng cần chatbot giúp đỡ
- References: người dùng muốn xin tài liệu học tập tham khảo từ chatbot
- Tip: người dùng muốn xin mẹo học lập trình từ chatbot
- Define: người dùng muốn hỏi chatbot một câu hỏi định nghĩa khái niệm
- Compare: người dùng muốn hỏi chatbot một câu hỏi so sánh giữa 2 khái niệm
- Apply: người dùng muốn hỏi chatbot một câu hỏi về ứng dụng của một khái niệm
- Other: người dùng đang hỏi hoặc nói về một chủ đề nằm ngoài phạm vi đề tài nghiên cứu này.

Chúng tôi tiến hành huấn luyện mô hình Linear SVM dựa trên tập dữ liệu đã được rút trích đặc trưng với nhãn là ý định người dùng. Mô hình này sau đó sẽ được dùng để dự đoán ý định người dùng một cách linh hoạt.

e - Mô hình Linear SVM dự đoán ngữ cảnh đối thoại (context classification)

Yếu tố ngữ cảnh của đối thoại mà chúng tôi sử dụng được đề cập đến trong đề tài này là ngôn ngữ lập trình được nhắc đến trong đoạn đối thoại. Quá trình dự đoán và suy diễn dựa trên ngữ cảnh (ngôn ngữ lập trình đang được đề cập) giúp chatbot phản ứng linh hoạt với câu hỏi của người dùng và bắt đúng khái niệm trong tập tri thức để trả lời câu hỏi chuẩn xác.

Chúng tôi chia ngữ cảnh thành 3 lớp như sau:

- General: câu hỏi về lập trình nhưng không đề cập cụ thể ngôn ngữ nào
- C++: câu hỏi thắc mắc về một khái niệm thuộc ngôn ngữ C++
- Python: câu hỏi thắc mắc về một khái niệm thuộc ngôn ngữ Python

Chúng tôi tiến hành huấn luyện mô hình Linear SVM dựa trên tập dữ liệu đã được rút trích đặc trưng với nhãn là ngữ cảnh đối thoại. Mô hình này sau đó sẽ được dùng để dự đoán ngữ cảnh của đối thoại một cách linh hoạt.

f - Retrieval-based answering

Trong đề tài này, chúng tôi không chú trọng vào việc hoàn thiện một knowledge base cho chatbot vì lý do hạn hẹp về mặt thời gian cũng như không nằm trong trọng tâm nghiên cứu của đề tài. Do đó, chúng tôi chỉ đề xuất một số ý tưởng cơ bản để từ câu hỏi ban đầu, cùng với ý định và ngữ cảnh do hai mô hình máy học cung cấp, chúng tôi xây dựng hệ luật dẫn và một tập tri thức để rút ra được câu trả lời tương ứng trong một tập các câu trả lời có sẵn.

Đối với ý định người dùng hay ngữ cảnh đối thoại, chúng tôi xây dựng hệ luật dẫn gồm các bộ theo cấu trúc (I_1, I_2, I_r) với I_1 là ý định hay ngữ cảnh của câu trước đó, I_2 là ý định hay ngữ cảnh của câu hiện tại đang xử lý, và I_r là ý định hay ngữ cảnh suy diễn được từ I_1 và I_2 .

Tập tri thức được tổ chức thành các bộ theo cấu trúc (T, C, Q, K) , với T là thuật ngữ lập trình, C là ngữ cảnh, Q là dạng câu truy vấn (Apply, Compare, Define), và K là tri thức tương ứng.

Tập câu trả lời có sẵn được tổ chức thành các bộ theo cấu trúc (I, A) hoặc (I, C, A) với I là ý định người dùng, C là ngữ cảnh và A là một danh sách các mẫu câu trả lời có sẵn để lấy ngẫu nhiên.

Từ hệ luật dẫn, tập tri thức và tập câu trả lời có sẵn, chatbot sẽ phản hồi lại với người dùng theo cách tốt nhất có thể để giải đáp thắc mắc cho người dùng hoặc định hướng cuộc đối thoại đúng theo chủ đề về lập trình cơ bản C++ hay Python.

3.3 - Ứng dụng CodEbot

a - Tổng quan ứng dụng

Ứng dụng CodEbot DEMO là ứng dụng thuộc đề tài cho phép người dùng tương tác với CodEbot, chatbot được chúng tôi xây dựng ở mục 3.2, qua một giao diện Web trực quan và đơn giản.

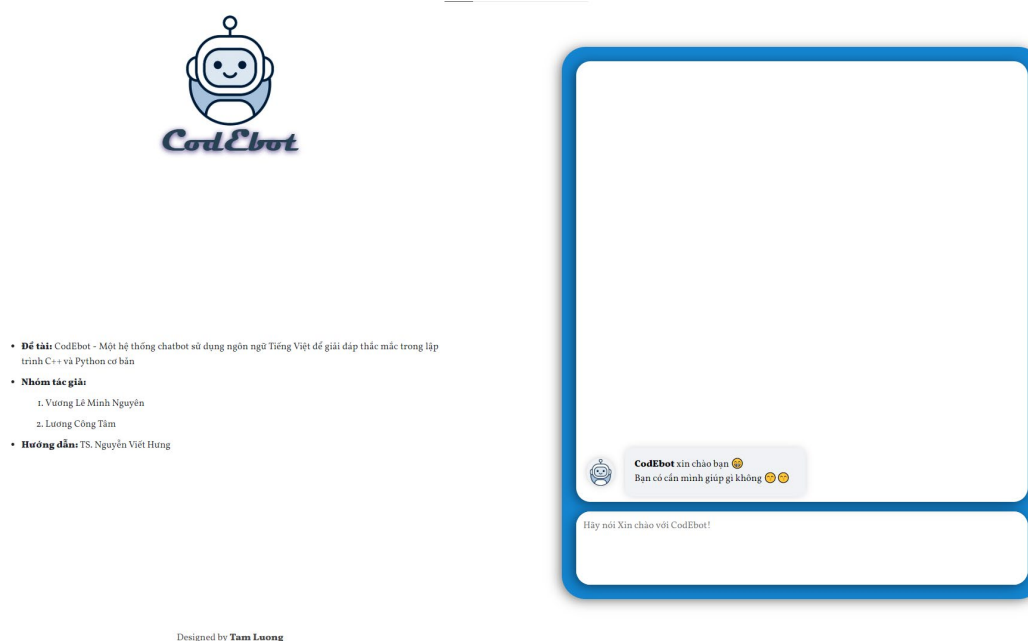
Ngoài ứng dụng để tương tác với CodEbot, chúng tôi cũng xây dựng thêm ứng dụng Knowledge Management để quản lý các tri thức của CodEbot nhằm giúp người dùng có thể dễ dàng đóng góp, cập nhật những kiến thức về lập trình của hệ thống.

Ứng dụng được xây dựng dựa trên công nghệ Web, với khả năng tương thích trên nhiều loại thiết bị, cho phép người dùng có thể truy cập và sử dụng CodEbot mọi lúc mọi nơi.

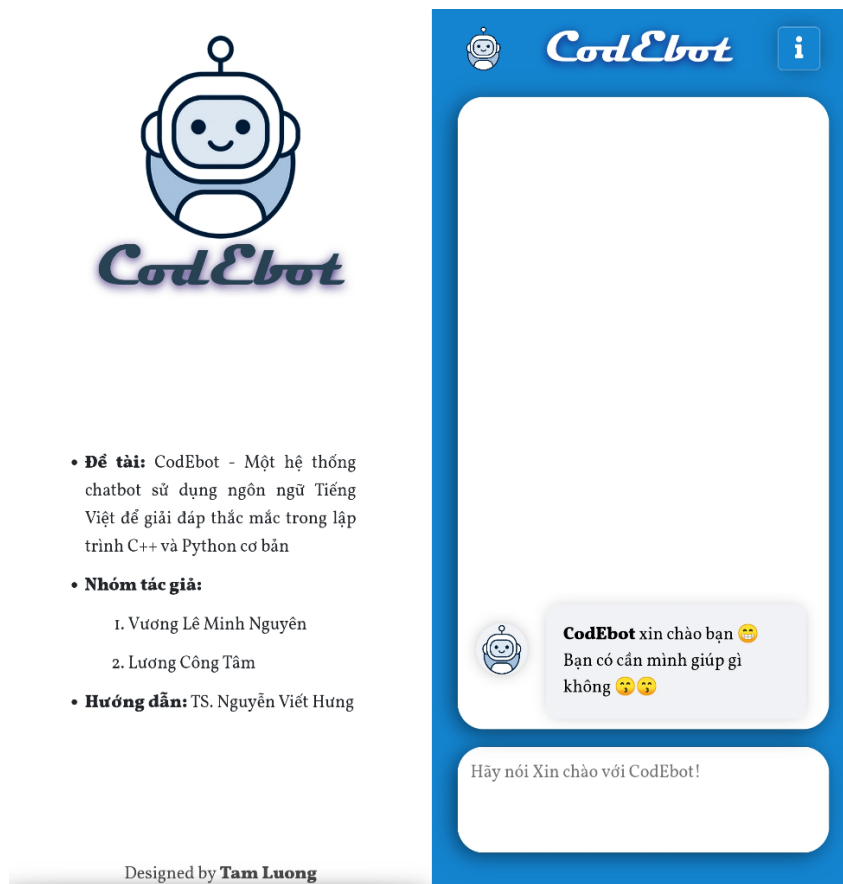
b - Giao diện chương trình

Phần giao diện ứng dụng được xây dựng với Bootstrap Framework, hỗ trợ tương thích với nhiều kích cỡ màn hình (Máy tính, laptop, điện thoại thông minh, máy tính bảng...).

Phần giao diện của ứng dụng CodEbot DEMO















Hình 3.2 Giao diện PC – CodEbot DEMO





Hình 3.3 Giao diện trên di động – CodEbot DEMO

Phần giao diện của ứng dụng Knowledge Management

CodEbot

Khái niệm	Ngữ cảnh	Công cụ
hàm	general, cpp, py	 
stdin	general, cpp, py	 
vòng lặp for	general, cpp, py	 
import	py	 
biến	general, cpp, py	 
con trỏ	cpp	 

Hình 3.4 Danh sách tri thức - Knowledge Management

CẬP NHẬT TRI THỨC

Khái niệm

hàm

Ngữ cảnh

CodEbot sẽ trả lời theo một ngữ cảnh tương ứng theo ngôn ngữ mà người dùng hỏi.
Mỗi lần trả lời có thể gồm nhiều bong bóng chat. Mỗi bong bóng chat cách nhau 2 dấu xuống dòng.

☒ Tổng quan - General

Trả lời các câu hỏi tổng quan trong lập trình.

Định nghĩa

Kiến thức cho câu hỏi định nghĩa.

C++ là một loại ngôn ngữ lập trình bậc. Đây là ngôn ngữ lập trình đa năng được tạo ra bởi Bjarne Stroustrup như một phần mở rộng của ngôn ngữ lập trình C, hoặc VC với các Class\'. Ngôn ngữ đã được mở rộng đáng kể theo thời gian và C++ hiện đại có các tính năng: lập trình tổng quát, lập trình hướng đối tượng, lập trình thủ tục, ngôn ngữ đa mẫu hình tự do có kiểu tĩnh, dữ liệu trừu tượng, và lập trình đa hình, ngoài ra còn có thêm các tính năng, công cụ để thao tác với bộ nhớ cấp.

Áp dụng

Kiến thức cho câu hỏi áp dụng.

C++ được thiết kế tương tự lập trình hệ thống và phần mềm nhưng, bao gồm cả hệ thống cơ sở nguyên nhân chế và tài nguyên khổng lồ, với ưu điểm là hiệu suất, hiệu quả và tính linh hoạt cao. C++ có thể tìm thấy ở mọi nơi, với những điểm mạnh là cơ sở hạ tầng phần mềm và các ứng dụng bị hạn chế tài nguyên, bao gồm: phần mềm ứng dụng máy tính cá nhân, các hệ thống máy chủ (ví dụ: thương mại điện tử, cỗ máy tìm kiếm trên web hoặc máy chủ SQL) và các ứng dụng ưu tiên về hiệu suất (ví dụ: tổng đài thông tin liên lạc hoặc thiết bị bị thâm dò không gian).

So sánh

Kiến thức cho câu hỏi so sánh.

C++ là ngôn ngữ lập trình bậc trung, thuộc dạng ngôn ngữ biên dịch, cần trình biên dịch để dịch mã nguồn ra mã máy. C++ là ngôn ngữ lập trình đa mục đích, có khả năng kiểm soát tài nguyên thực thi vượt trội (thời gian chạy tối ưu và cho phép lập trình viên quản lý bộ nhớ qua con trỏ). C++ được ứng dụng trong lập trình hệ thống, hệ điều hành, phần mềm nhúng, công cụ đồ họa, game engines, trình biên dịch, thông dịch cho các ngôn ngữ khác,...

Nguồn

Mỗi nguồn cách nhau bởi một dấu xuống dòng.

[https://vi.wikipedia.org/wiki/C++](https://vi.wikipedia.org/wiki/C%2B%2B)

☐ Ngôn ngữ C++

Trả lời các câu hỏi về ngôn ngữ C++.

☐ Ngôn ngữ Python

Mỗi lần trả lời có thể gồm nhiều bong bóng chat. Mỗi bong bóng chat cách nhau 2 dấu xuống dòng.

Hình 3.5 Giao diện cập nhật chi thức - Knowledge Management

23

CHƯƠNG 4 - KẾT QUẢ - ĐÁNH GIÁ

Mô hình được huấn luyện và đánh giá với bộ dữ liệu do nhóm tự thu thập. Trong quá trình đánh giá, bộ dữ liệu được chia làm 2 phần:

- Phần huấn luyện chiếm 80% bộ dữ liệu
- Phần đánh giá chiếm 20% bộ dữ liệu

Phương thức đánh giá được sử dụng là F1-Score.

Kết quả đánh giá dựa trên mô hình dự đoán ý định (Overall F1-Score: 0.96):

Intent	Precision	Recall	F1-score
agree	1.00	0.86	0.92
apply	1.00	1.00	1.00
compare	1.00	1.00	1.00
credit_info	1.00	1.00	1.00
define	0.80	1.00	0.89
disagree	1.00	1.00	1.00
greeting	1.00	1.00	1.00
help	1.00	0.80	0.89
other	0.90	1.00	0.95
references	0.86	1.00	0.92
tip	1.00	1.00	1.00

Kết quả đánh giá của mô hình dự đoán ngữ cảnh (Overall F1-Score: 0.99):

Context	Precision	Recall	F1-score
c++	1.00	0.94	0.97
py	1.00	1.00	1.00
general	0.98	1.00	0.99

CHƯƠNG 5 - TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

5.1 - Tổng kết

Tuy lượng dữ liệu thu thập được và các dạng câu hỏi còn hạn chế, mô hình máy học vẫn còn bị thiên kiến (high bias) mà nguyên nhân là thiếu ngôn ngữ địa phương hay từ lóng trong tập dữ liệu, nhưng kết quả từ mô hình xác định ý định của người dùng vẫn rất chính xác (f1-score đạt trên 0.9 với cả hai mô hình dự đoán ý định và dự đoán ngữ cảnh) trong mức giới hạn của đề tài.

Bên cạnh các hạn chế nêu trên, kết quả này có thể được xem là tiền đề để phát triển, mở rộng các dạng câu hỏi, hành vi hay chức năng của CodEbot trong các nghiên cứu, dự án sau.

5.2 - Hướng phát triển

Mặc dù đề tài đã đạt được những kết quả khả quan, nhưng CodEbot vẫn cần được phát triển thêm để có thể đáp ứng được cho nhu cầu học lập trình đang ngày càng gia tăng. Những tiến triển tiếp theo mà nhóm dự tính là:

- **Xây dựng bộ dữ liệu cho mô hình NLP:** Trong lĩnh vực Trí tuệ nhân tạo, dữ liệu luôn là thành phần quan trọng nhất. Các mô hình máy học, học sâu, ... không thể học trực tiếp từ con người mà phải học qua các dữ liệu đó. Chính vì thế, xây dựng bộ dữ liệu kích thước lớn cũng là bước quan trọng đầu tiên cần làm để phát triển đề tài.
- **Cải thiện ứng dụng quản lý cơ sở dữ liệu:** Khi tổng hợp, thêm, sửa đổi các dữ liệu kiến thức về lập trình trong cơ sở dữ liệu, không thể cứ thay đổi trực tiếp trong cơ sở dữ liệu được, mà cần phải quản lý nó thông qua một giao diện trực quan.
- **Mở rộng các dạng câu hỏi, chức năng:** Số lượng các dạng câu hỏi trong đề tài này còn rất ít, chưa đủ để cung cấp các kiến thức trong lập trình. Ngoài ra, nhóm còn muốn phát triển thêm các chức năng truy vấn thông tin như tài liệu học tập, lịch học và bài tập, ...

TÀI LIỆU THAM KHẢO

- [1] Dale, Robert, “The return of the chatbots,” in *Natural Language Engineering*, vol. 22, pp. 811-817, 2016.
- [2] Asbjørn Følstad, Petter Bae Brandtzæg, “Chatbots – the new world of HCI,” in *ACM Interactions*, vol.4, no.4, June 2017.
- [3] Joseph Weizenbaum, “ELIZA – a computer program for the study of natural language communication between man and machine,” in *Communications of the ACM*, vol. 9, no. 1, Jan 1966.
- [4] H. Shah, K. Warwick, J. Vallverdú, D. Wu, “Can machines talk? Comparison of ELIZA with modern dialogue systems,” in *Computers in Human Behavior*, vol. 58, no. C, pp. 278–295, May 2016.
- [5] Oriol Vinyals, Quoc Le, “A Neural Conversational Model,” 2015.
- [6] Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, Bowen Zhou, “Improved Neural Relation Detection for Knowledge Base Question Answering,” 2017.
- [7] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, Wei Wang, “KBQA: Learning Question Answering over QA Corpora and Knowledge Bases,” in *Proceedings of the VLDB Endowment*, vol. 10, pp.565-576, Jan 2017.
- [8] K. Lei, Y. Deng, B. Zhang and Y. Shen, "Open Domain Question Answering with Character-Level Deep Learning Models," 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, pp. 30-33, 2017.
- [9] Wei Emma Zhang, Quan Z. Sheng, Jey Han Lau, Ermyas Abebe, “Detecting Duplicate Posts in Programming QA Communities via Latent Semantics and Association Rules,” April 2017.
- [10] Rainer Winkler, Matthias Söllner, “Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis,” 78th annual meeting of the academy of management, Chicago, Illinois, March 2018.
- [11] Pavel Smutny, Petra Schreiberova, “Chatbots for learning: A review of educational chatbots for the Facebook Messenger,” in *Computers & Education*, vol. 151, 2020.
- [12] D. A. Ferrucci, “Introduction to “This is Watson”,” in *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 1:1-1:15, May-June 2012.

- [13] A. Lally et al., “Question analysis: How Watson reads a clue,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 2:1-2:14, May-June 2012.
- [14] M. C. McCord, J. W. Murdock and B. K. Boguraev, “Deep parsing in Watson,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 3:1-3:15, May-June 2012.
- [15] J. Chu-Carroll, J. Fan, N. Schlaefer and W. Zadrozny, “Textual resource acquisition and engineering,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 4:1-4:11, May-June 2012.
- [16] J. Fan, A. Kalyanpur, D. C. Gondek and D. A. Ferrucci, “Automatic knowledge extraction from documents,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 5:1-5:10, May-June 2012.
- [17] J. Chu-Carroll, J. Fan, B. K. Boguraev, D. Carmel, D. Sheinwald and C. Welty, “Finding needles in the haystack: Search and candidate generation,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 6:1-6:12, May-June 2012.
- [18] J. W. Murdock et al., “Typing candidate answers using type coercion,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 7:1-7:13, May-June 2012.
- [19] J. W. Murdock, J. Fan, A. Lally, H. Shima and B. K. Boguraev, “Textual evidence gathering and analysis,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 8:1-8:14, May-June 2012.
- [20] C. Wang, A. Kalyanpur, J. Fan, B. K. Boguraev and D. C. Gondek, “Relation extraction and scoring in DeepQA,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 9:1-9:12, May-June 2012.
- [21] A. Kalyanpur et al., “Structured data and inference in DeepQA,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 10:1-10:14, May-June 2012.
- [22] J. M. Prager, E. W. Brown and J. Chu-Carroll, “Special Questions and techniques,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 11:1-11:13, May-June 2012.
- [23] J. Chu-Carroll, E. W. Brown, A. Lally and J. W. Murdock, “Identifying implicit relationships,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 12:1-12:10, May-June 2012.
- [24] A. Kalyanpur, S. Patwardhan, B. K. Boguraev, A. Lally and J. Chu-Carroll, “Fact-based question decomposition in DeepQA,” in IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 13:1-13:11, May-June 2012.

- [25] D. C. Gondek et al., “A framework for merging and ranking of answers in DeepQA,” in *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 14:1-14:12, May-June 2012.
- [26] E. A. Epstein, M. I. Schor, B. S. Iyer, A. Lally, E. W. Brown and J. Cwiklik, “Making Watson fast,” in *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 15:1-15:12, May-June 2012.
- [27] G. Tesauro, D. C. Gondek, J. Lenchner, J. Fan and J. M. Prager, “Simulation, learning, and optimization techniques in Watson's game strategies,” in *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 16:1-16:11, May-June 2012.
- [28] Oanh Tran, Tho Luong, “Understanding what the users say in chatbots: A case study for the Vietnamese language,” in *Engineering Applications of Artificial Intelligence*, vol.87, Jan 2020
- [29] Lan Ngo, Linh Pham, Hideaki Takeda, Bao Pham, Hieu Phan, “On the Identification of Suggestion Intents from Vietnamese Conversational Texts,” in *SoICT 2017: Proceedings of the Eighth International Symposium on Information and Communication Technology*, pp. 417-424, Dec 2017
- [30] T. Luong, M. Cao, D. Le and X. Phan, “Intent extraction from social media texts using sequential segmentation and deep learning models,” *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 215-220, Hue, Oct 2017.
- [31] Lan Ngo, Linh Pham, Son Cao. Bao Pham, Hieu Phan, “Dialogue act segmentation for Vietnamese human-human conversational texts,” in *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 203-208, Hue, Oct 2017
- [32] T. Quan et al., “Lead Engagement by Automated Real Estate Chatbot,” *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, pp. 357-359, Ho Chi Minh City, Nov 2018.
- [33] Chunxi Liu, Puyang Xu, Ruhi Sarikaya, “Deep contextual language understanding in spoken dialogue systems,” In *INTERSPEECH*, pp. 120-124, 2015.
- [34] Dinoj Surendran, Gina Anne Levow, “Dialog act tagging with support vector machines and hidden markov models,” in *In Proceedings of Interspeech/ICSLP*, pp. 1–28, 2006.
- [35] Simon Keizer, “Dialogue act modelling using bayesian networks,” 2001.

- [36] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, M. Meteer, “Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech,” in Computational Linguistics, pp. 339-373, Sep 2000.
- [37] S.A. Ali, N. Sulaiman, A. Mustapha, N. Mustapha, “K-Mean Clustering to Improve the Accuracy of Decision Tree Response Classification,” in Information Technology Journal, vol. 8, no. 8, pp. 1256-1262, 2009.
- [38] Vu Anh et al., Underthesea - Vietnamese NLP Toolkit, (2019), Github Repository, <https://github.com/undertheseanlp/underthesea>