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

**Vương Lê Minh Nguyên**

(Sinh viên năm 3, Khoa Công nghệ Thông tin)

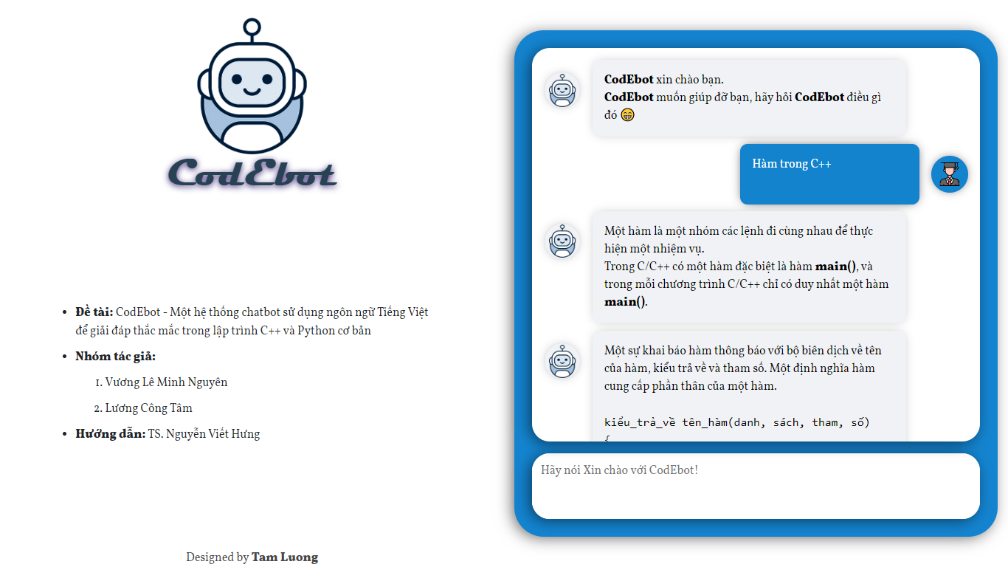
**Lương Công Tâm**

(Sinh viên năm 3, Khoa Công nghệ Thông tin)

***GVHD: TS Nguyễn Viết Hưng***

# **Giới thiệu**

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 kĩ năng lập trình cũng được nhìn nhận là một trong những kĩ 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 Xử lí ngôn ngữ tự nhiên (Natural Language Processing – NLP) và hệ thống tri thức. Dữ liệu huấn luyện mô hình NLP do nhóm tự thu thập và cơ sở dữ liệu các câu trả lởi cũng được tổng hợp và phiên dịch từ các nguồn kiến thức có thể tin cậy.



*Hình 1. CodEbot - Hệ thống chatbot giải đáp thắc mắc trong lập trình cơ bản*

# **Lí do chọn đề tài**

Bản thân người thực hiện đề tài là sinh viên ngành Công nghệ Thông tin, chúng tôi nhận thức được rõ rệt những khó khăn của đa số sinh viên khi mới bắt đầu học lập trình. Sách giáo khoa về lập trình viết bằng tiếng Việt còn rất hạn chế, và đa số được viết bằng ngôn ngữ hàn lâm nên sinh viên chưa biết gì về lập trình sẽ rất khó tiếp cận, tiếp thu. Sách giáo khoa hay tài liệu lập trình tìm kiếm trên Internet đa số viết bằng tiếng Anh, và đối với người mới học lập trình thì khả năng tổng hợp kiến thức và tự giải đáp thắc mắc cho bản thân dựa trên kho kiến thức khổng lồ bằng tiếng Anh này là gần như bất khả thi.

Với sự phát triển của khoa học công nghệ và các tác vụ tự động hóa, khả năng tương tác, giao tiếp giữa người với máy đã và đang thay đổi, thông qua các quá trình xử lí ngôn ngữ tự nhiên, việc giao tiếp với các thiết bị kĩ thuật số hiện đại không còn phải bằng các thao tác trượt, lướt, nhất nút nữa mà thay vào đó là sử dụng những đoạn văn bản bằng ngôn ngữ tự nhiên, điển hình là các công nghệ chatbot trả lời tự động (Følstad, 2017).

Các nghiên cứu và xây dựng hệ thống chatbot ngày càng trở nên phổ biến và thu hút được nhiều sự chú ý từ các chuyên gia trong lĩnh vực Xử lí ngôn ngữ tự nhiên. Thông qua các bài kiểm tra như Turing Test, các hệ thống chatbot hiện đại đã được đánh giá ngày càng trở nên thông minh hơn với lối giao tiếp trở nên gần giống với con người (Shah, 2016). Chúng được sử dụng nhiều trong các lĩnh vực khác nhau, từ tư vấn dịch vụ, chăm sóc khách hàng, hỗ trợ kĩ thuật cho đến các dịch vụ tư vấn sức khỏe, y tế, và lĩnh vực giáo dục cũng không là ngoại lệ (Winkler, 2018). Các hệ thống chatbot lời tự động thường được xây dựng thành các ứng dụng độc lập hoặc được tích hợp vào các chương trình nhắn tin có sẵn, trong đó phổ biến nhất là Facebook Messenger (Smutny, 2020). Các chatbot trong lĩnh vực giáo dục được xây dựng nhằm giúp người học có thể tra cứu thông tin lớp học, xem lại bài giảng trong lớp hay giúp các giảng viên thu thập mức độ hài lòng của người học với lớp học của mình một cách tự động. Tuy nhiên, các chatbot giáo dục trong tiếng Việt lại không được phổ biến hay chưa thực sự hữu dụng.

Cùng những tiến bộ đáng kể trong lĩnh vực phân tích xử lí ngôn ngữ tự nhiên tại Việt Nam, việc xây dựng một hệ thống chatbot để tự động hóa các tác vụ giao tiếp giữa người và máy thông qua ngôn ngữ tiếng Việt không còn quá khó khăn như trước đây. Chúng tôi nhận thấy một trong những cách tiếp cận để giảm bớt những khó khăn cho học sinh, sinh viên khi mới học lập trình là cung cấp một công cụ giao tiếp bằng ngôn ngữ tiếng Việt để hỗ trợ học sinh, sinh viên tìm kiếm một lời giải đáp cho những thắc mắc cơ bản của mình mà việc hỏi thầy cô hay bạn bè có thể không được thuận tiện.

Vì những lí do khách quan và chủ quan đã nêu trên, trong đề tài này, chúng tôi quyết định nghiên cứu và hiện thực hóa một hệ thống chatbot ứng dụng các kĩ thuật máy học chuyên dụng để xử lí ngôn ngữ tự nhiên nhằm giải đáp một nhóm các câu hỏi về lập trình cơ bản dựa trên một hệ cơ sở tri thức được tổng hợp từ những nguồn đáng tin cậy.

# **Mục tiêu**

Các nghiên cứu về xử lí ngôn ngữ tự nhiên, ngôn ngữ Tiếng Việt gần đây được nhóm nghiên cứu quan tâm và tìm hiểu 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 lí ngôn ngữ tự nhiên Tiếng Việt, dùng để phân tích ý định của người dụng 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 không đạt được độ chính xác cao, nhưng nhóm nghiên cứu vẫn hy vọng có thể cải thiện CodEbot hơn trong tương lai.

# **Giới hạn**

## **Giới hạn về kiến thức**

Tuy CodEbot là một hệ thống chatbot giải đáp kiến thức về lập trình nhưng 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:** Hỗ trợ 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…

## **Giới hạn về dữ liệu**

Đề tài này gồm 2 loại dữ liệu là kho ngữ liệu (Corpus) ở dạng văn bản thuần và dữ liệu câu hỏi về lập trình có dán nhãn để huấn luyện các mô hình máy học giải quyết bài toán phân lớp trong NLP.

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.

# **Phương pháp**

## **Thu thập dữ liệu**

Có 2 loại dữ liệu được sử dụng trong đề tài này, với mỗi loại có một cách thức riêng để thu thập.

### Kho ngữ liệu (Corpus)

Đối với kho ngữ liệu (Corpus), 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 5.2.2).

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.

### 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).

## **Hệ thống chatbot giải đáp các vấn đề về lập trình C++ và lập trình Python cơ bản**

### Tổng quan

***Sơ đồ 1*.** *Tổng quan hệ thống*

A screenshot of a cell phone

Description automatically generated

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.

### Tiền xử lí 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.

*Sơ đồ 2. Quá trình tiền xử lí và trích xuất đặc trưng*

A screenshot of a cell phone

Description automatically generated

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 lí 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** (Vu, 2019) 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 lí 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 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 – **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:

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:

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.

### Lựa chọn mô hình phân lớp ngôn ngữ tự nhiên – Support Vector Machine

Trong các thuật toán máy học phân loại dữ liệu không có cấu trúc, các mô hình máy học truyền thống như SVM, CRF cùng với các mô hình học sâu 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 kĩ thuật rút trích dữ liệu phức tạp. Điển hình trong một nghiên cứ về phân loại ý định người dùng trong ngôn ngữ tiếng Việt, các tác giả tiến hành xây dựng mô hình phân loại ý định người dùng trong nhiều lĩnh vực bằng một mạng học sâu LSTM kết hợp mô hình máy học CRF đã đạt được kết quả đánh giá rất cao (Luong, 2017). Trong một nghiên cứu khác cùng năm, các tác giả sử dụng mô hình học sâu CNN với lớp đầu ra Softmax để giải quyết bài toán phân loại ý định người dùng đạt kết quả đánh giá khả quan (Ngo, 2017). Một nghiên cứu khác cũng về chủ đề phân loại ý định người dùng nhưng chỉ trong lĩnh vực bất động sản, các tác giả sử dụng kết hợp mạng học sâu CNN kết hợp với mạng học sâu Bi-GRU để rút trích đặc trưng và phân loại đầu ra bằng mô hình CRF cũng đạt được kết quả đánh giá rất tốt (Quan et al., 2018). Một nghiên cứu mới đây sử dụng kết hợp mạng Bi-LSTM và mạng CNN, đồng thời đầu ra sử dụng CRF để phân loại ý định người dùng trong đối thoại cũng đạt được kết quả rất tốt (Tran, 2020).

Có thể thấy, các mô hình học sâu đã được sử dụng rộng rãi trong các nghiên cứu về phân tích ý định người dùng và ngữ cảnh trong ngôn ngữ tiếng Việt những năm gần đây. Tuy nhiên, trong năm 2017, một nghiên cứu do nhóm tác giả người Việt thực hiện đã tiến hành so sánh thuật toán máy học cổ điển như ME hay CRF với phương pháp học sâu hiện đại là Bi-LSTM – CRF trên tập dữ liệu tin nhắn điện thoại và mạng xã hội cho thấy không có chênh lệch quá lớn trong kết quả đánh giá giữa các phương pháp trên. F1-Score giữa phương pháp máy học truyền thống và phương pháp học sâu hiện đại chỉ chênh lệch nhau không quá 5% (Ngo, 2017). Tất nhiên ở thời điểm hiện tại, tức là 3 năm sau nghiên cứu đó, con số 5% không còn đúng nữa, nhưng nó cũng cho thấy kết quả khả quan của các phương pháp máy học truyền thống trong bài toán phân lớp ngôn ngữ tự nhiên. Do đó, vì kích thước tập dữ liệu phục vụ cho nghiên cứu của chúng tôi là rất nhỏ để sử dụng các mô hình học sâu (kích thước dữ liệu khoảng 1000 câu hỏi), chúng tôi quyết định chọn Support Vector Machine làm thuật toán máy học để giải quyết bài toán phân loại ý định người dùng và ngữ cảnh đối thoại trong nghiên cứu này.

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.

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 cuối cùng.

### 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
* **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.

### 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:

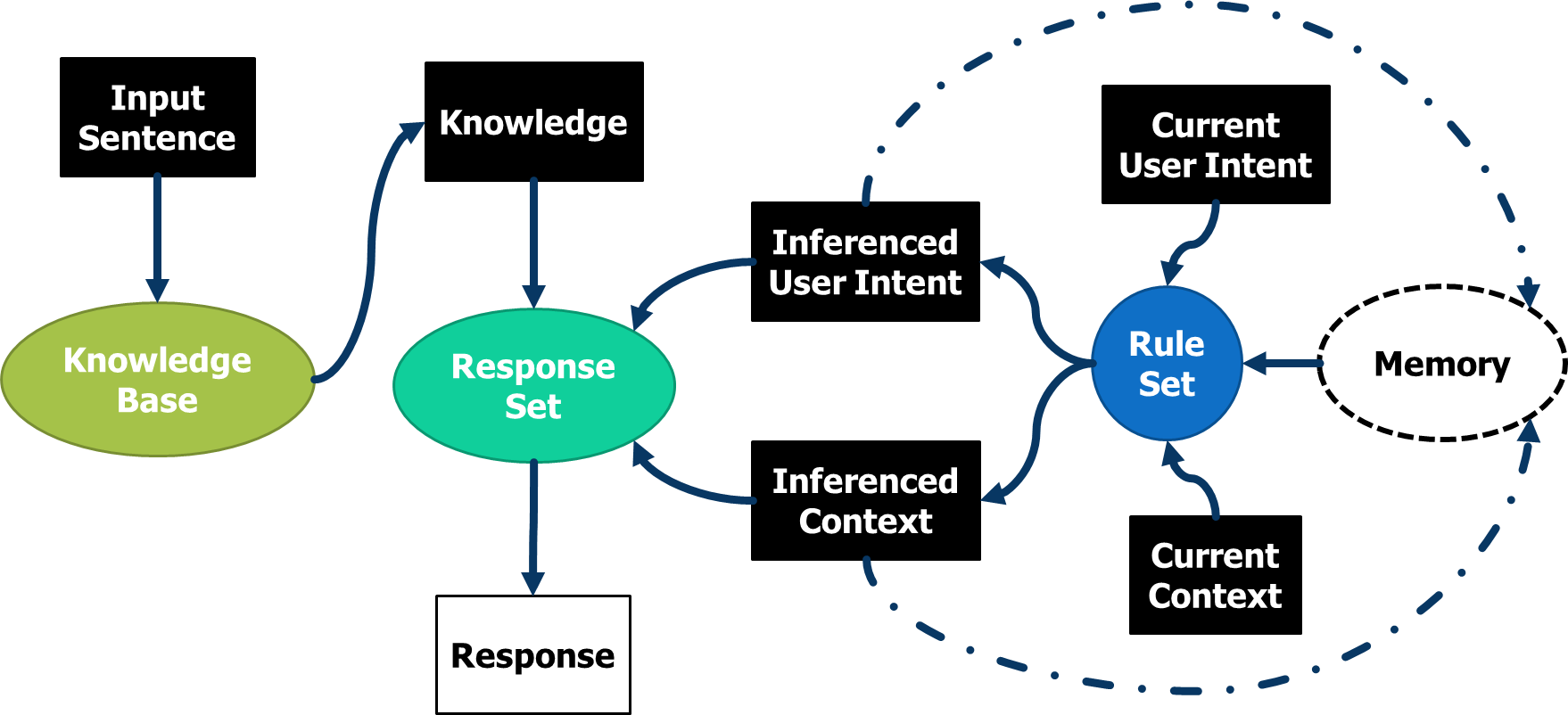
* **General**: câu hỏi về lập trình nhưng không đề cập cụ thể ngôn ngữ nào
* **C++**: câu hỏi thắc mắc về một khái niệm thuộc ngôn ngữ C++
* **Python**: câu hỏi thắc mắc về một khái niệm thuộc ngôn ngữ Python

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.

### Retrieval-based answering

Trong đề tài này, chúng tôi không chú trọng vào việc hoàn thiện một knowledge base cho chatbot vì lí 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.

*Sơ đồ 3. Retrieval-based answering*



Đối với ý định người dùng hay ngữ cảnh đối thoại, chúng tôi xây dựng hệ luật dẫn gồm các bộ theo cấu trúc (I1, I2, Ir) với I1 là ý định hay ngữ cảnh của câu trước đó, I2 là ý định hay ngữ cảnh của câu hiện tại đang xử lí, và Ir là ý định hay ngữ cảnh suy diễn được từ I1 và I2.

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.

## **Ứng dụng CodEbot DEMO**

### Tổng quan

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.

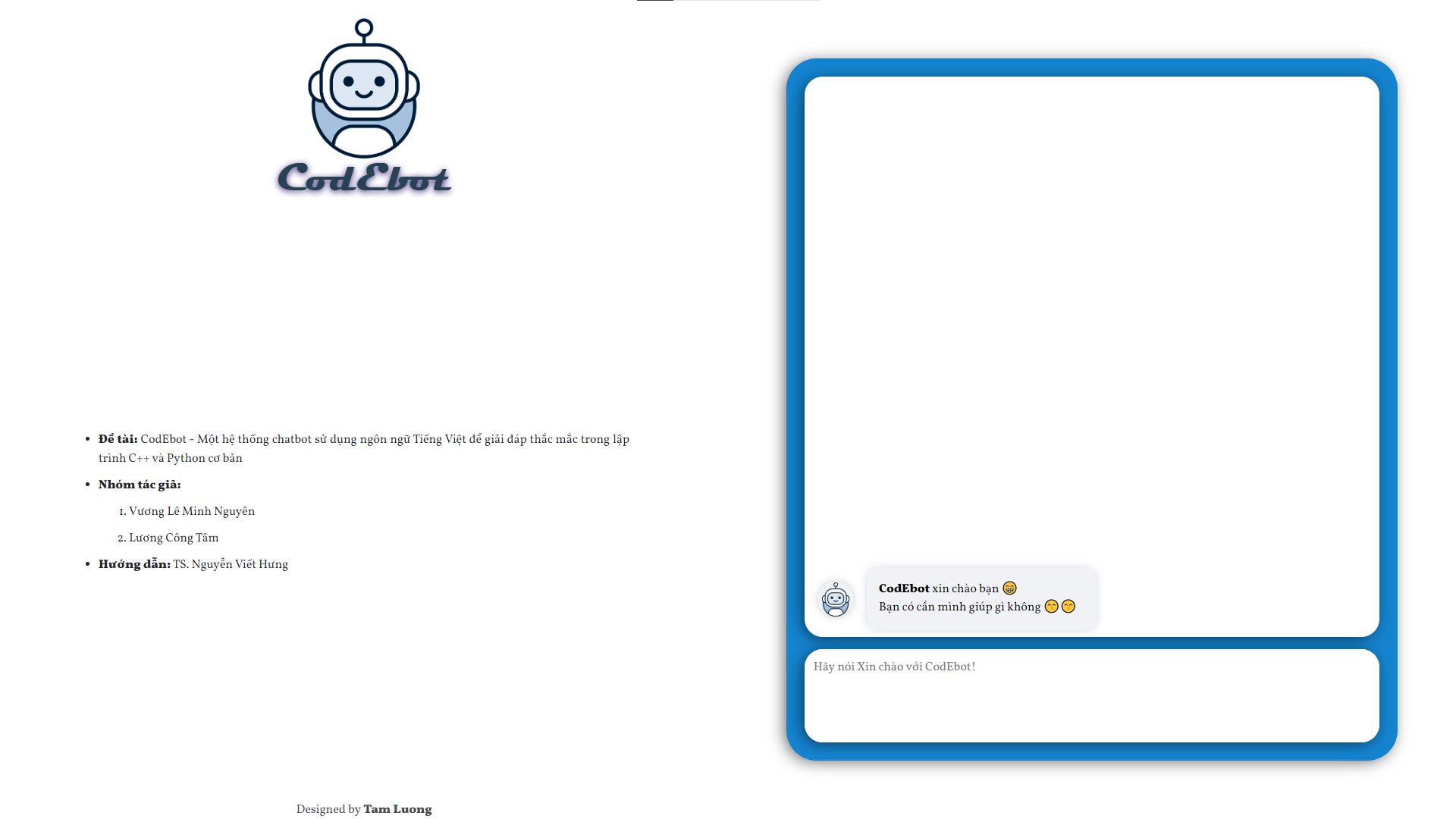
Ngoài ứng dụng để tương tác với CodEbot, chúng tôi cũng xây dựng thêm ứng dụng Knowledge Management để quản lí các tri thức của CodEbot nhằm giúp người dùng có thể dễ dàng đóng góp, cập nhật những kiến thức về lập trình của hệ thống.

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.

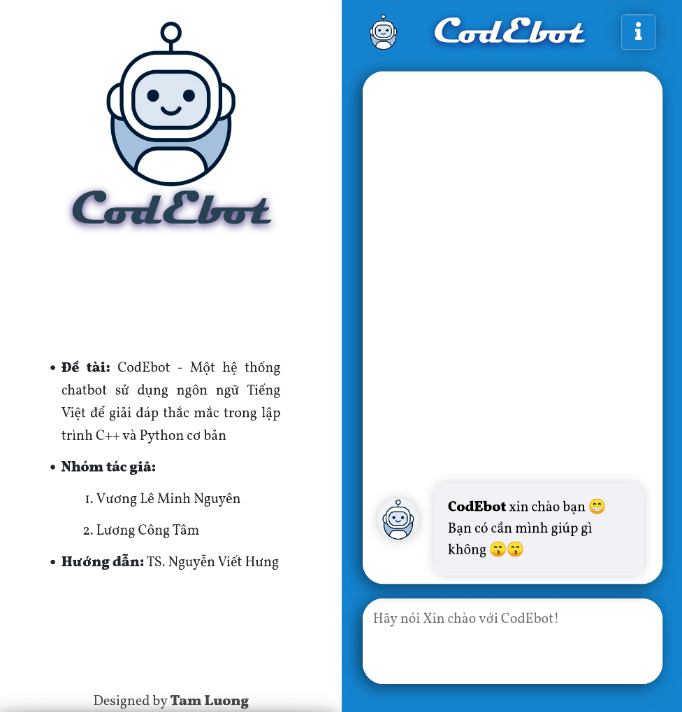
### Giao diện người dùng

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**

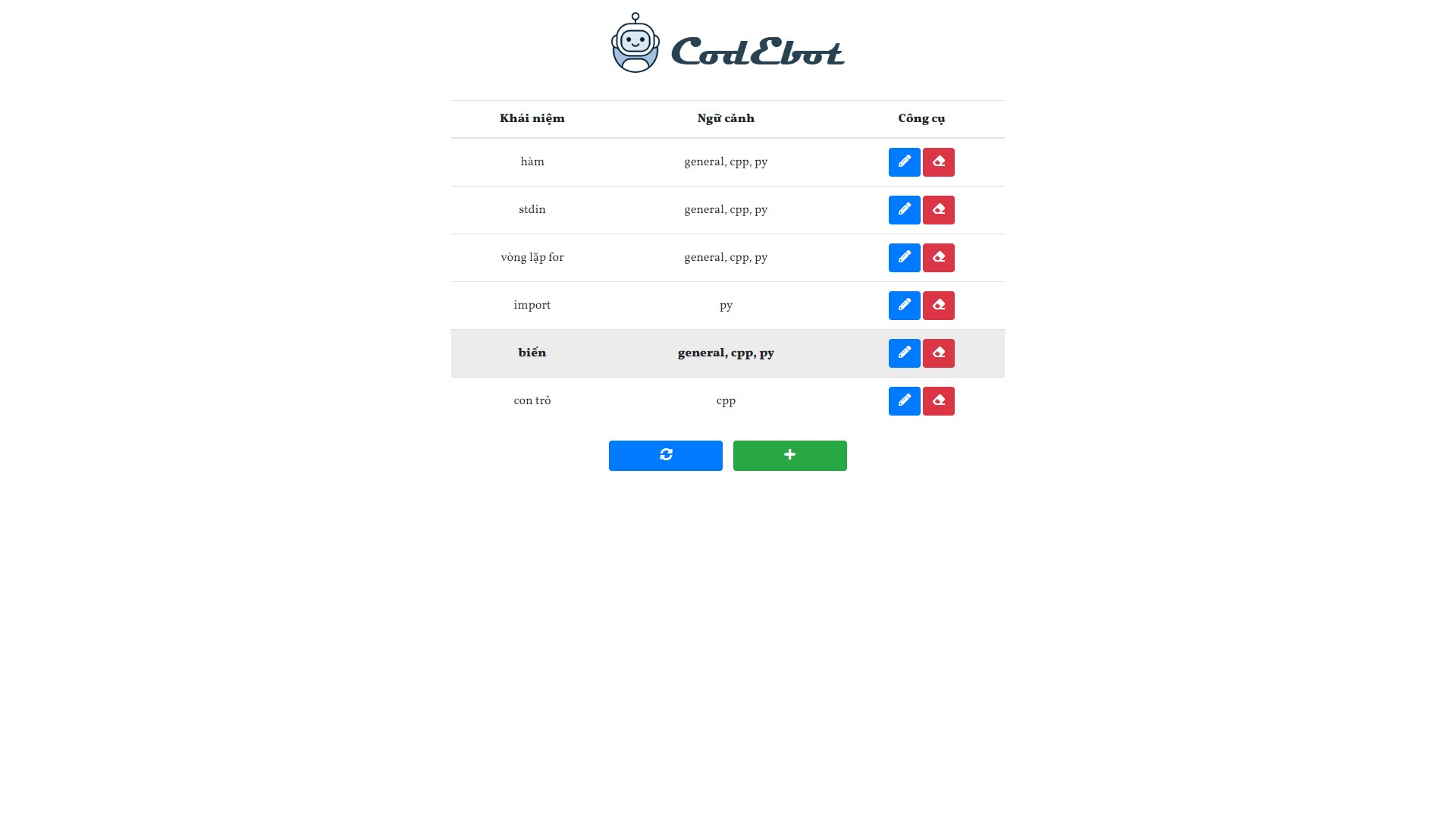


*Hình 2. Giao diện CodEbot Demo trên thiết bị màn hình rộng*

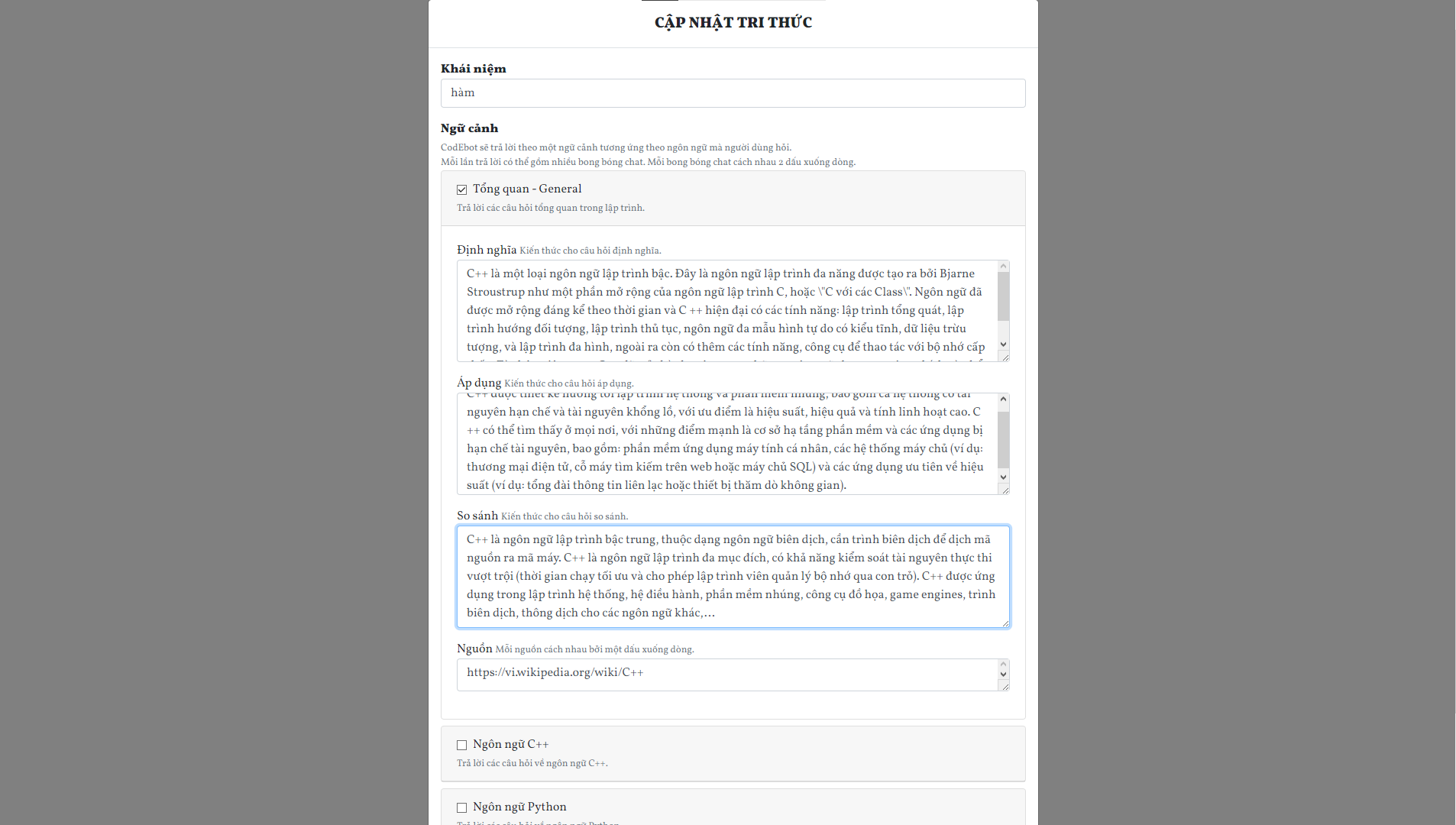


*Hình 3. Giao diện CodEbot Demo trên thiết bị màn hình hẹp*

* Phần giao diện của ứng dụng **Knowledge Management**



*Hình 4. Giao diện danh sách tri thức*



*Hình 5. Giao diện cập nhật tri thức*

# **Kết quả nghiên cứu**

Các mô hình Linear SVM 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.

Trong bài toán phân lớp đa lớp với lượng dữ liệu mỗi lớp là tương đương nhau theo chiến lược one vs all, việc sử dụng phương pháp đánh giá dựa trên Accuracy (độ chính xác) không phản ánh đúng khả năng dự đoán của mô hình phân lớp vì mỗi lớp luôn có lượng dữ liệu ít hơn so với lượng dữ liệu thuộc về các lớp khác. Giả sử một lớp có 50 điểm dữ liệu và các lớp còn lại có 950 điểm dữ liệu thì mô hình chỉ việc dự đoán mọi điểm dữ liệu trong tương lai thuộc các lớp còn lại thì Accuracy đã là 95%. Từ giả sử trên, ta cần một phương pháp đánh giá khả năng phân lớp thực sự của mô hình trên lớp có lượng dữ liệu ít hơn gọi là lớp Positive, phân biện với lớp đa số là Negative. Để khắc phục nhược điểm của Accuracy, phương thức được sử dụng để đánh giá các mô hình trong đề tài này là F1-Score.

**F1-Score** là trung bình điều hòa (harmonic mean) của **Precision** và **Recall**:

**Precision** là độ chính xác của mô hình khi dự đoán lớp Positive, được xác định bằng công thức sau:

**Recall** là độ nhạy của mô hình khi dự đoán lớp Positive, được xác định bằng công thức sau:

*Bảng 1. Kết quả đánh giá của mô hình dự đoán ý định   
(F1-Score trung bình: 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 |

*Bảng 2 Kết quả đánh giá của mô hình dự đoán ngữ cảnh   
(F1-Score trung bình: 0.99)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Context** | **Precision** | **Recall** | **F1-score** |
| cpp | 1.00 | 0.94 | 0.97 |
| py | 1.00 | 1.00 | 1.00 |
| general | 0.98 | 1.00 | 0.99 |

# 

# **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.

# **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 lí 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 lí 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 đa dạng 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**

Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *ACM Interactions,* *24*(4), 38-42. doi:10.1145/3085558

Luong, T., Cao, M., Le, D., & Phan, X. (2017). Intent extraction from social media texts using sequential segmentation and deep learning models. *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*. doi:10.1109/kse.2017.8119461

Ngo, T., Pham, S., Pham, K., Phan, X., & Cao, M. (2017). Dialogue act segmentation for Vietnamese human-human conversational texts. *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*. doi:10.1109/kse.2017.8119459

Ngo, T., Pham, K. L., Takeda, H., Pham, S. B., & Phan, X. H. (2017). On the Identification of Suggestion Intents from Vietnamese Conversational Texts. *Proceedings of the Eighth International Symposium on Information and Communication Technology - SoICT 2017*. doi:10.1145/3155133.3155201

Quan, T., Trinh, T., Ngo, D., Pham, H., Hoang, L., Hoang, H., . . . Mai, T. (2018). Lead Engagement by Automated Real Estate Chatbot. *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*. doi:10.1109/nics.2018.8606862

Shah, H., Warwick, K., Vallverdú, J., & Wu, D. (2016). Can machines talk? Comparison of Eliza with modern dialogue systems. *Computers in Human Behavior,* *58*, 278-295. doi:10.1016/j.chb.2016.01.004

Smutny, P., & Schreiberova, P. (2020). Chatbots for learning: A review of educational chatbots for the Facebook Messenger. *Computers & Education,* *151*, 103862. doi:10.1016/j.compedu.2020.103862

Tran, O. T., & Luong, T. C. (2020). Understanding what the users say in chatbots: A case study for the Vietnamese language. *Engineering Applications of Artificial Intelligence,* *87*, 103322. doi:10.1016/j.engappai.2019.103322

Vu, A., Bui, A. N., & Doan, D. V. (2018). Xây dựng hệ thống tách từ tiếng Việt. Retrieved from https://github.com/undertheseanlp/word\_tokenize

Winkler, R., & Soellner, M. (2018). Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis. *Academy of Management Proceedings,* *2018*(1), 15903. doi:10.5465/ambpp.2018.15903abstract