TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH

HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE

Tập 18, Số 9 (2021): 1711-1723

Vol. 18, No. 9 (2021): 1711-1723

ISSN: 2734-9918

Website: *http://journal.hcmue.edu.vn*

Research Article

CODEBOT – A VIETNAMESE CHATBOT SYSTEM FOR ANSWERING C++ AND PYTHON-RELATED QUESTIONS

Vuong Le Minh Nguyen[1*], Luong Cong Tam[1], Nguyen Viet Hung[1], Nguyen Do Thai Nguyen[1], Luong Tran Hy Hien[1], Luong Tran Ngoc Khiet[1], Phan Thi Trinh[2]

[1]Ho Chi Minh City University of Education, Vietnam

[2]Thu Duc College of Technology, Vietnam

[*]Corresponding author: Vuong Le Minh Nguyen – Email: nguyenvlm.43@student.hcmue.edu.vn

Received: April 12, 2021; Revised: June 03, 2021; Accepted: September 03, 2021

ABSTRACT

During the fourth industrial revolution, the ability of programming is one of the most essential skills for the youth to earn an edge over the competitors in their specializations. Programming techniques are not only important to software development but also useful for statistical analytics and mathematical modelling in other fields of study. However, fundamental programming materials on the Internet are mostly written in English instead of Vietnamese, which sets a distance between these materials and Vietnamese youth. This led to the idea of having a simple yet effective Vietnamese question answering chatbot to engage and motivate Vietnamese students to climb the steep learning curve of programming. This paper, combining natural language processing with knowledge representation and reasoning, aimed to implement such a question answering chatbot in pure Vietnamese to help students with their programming-related questions. A simple knowledge representation method was introduced to integrate external knowledge into the system. A knowledge reasoning and retrieval-based question answering method was also proposed to effectively yield proper responses from user's queries. The range of topics the chatbot supports is limited to C++ and Python, two of the most taught programming languages in Vietnamese colleges and universities. At the heart of our chatbot, two machine learning models were designed to classify user's intents. They were trained and evaluated on our annotated dataset, which was contributed by students from the Faculty of Information Technology, Ho Chi Minh University of Education. Our proposed models achieved surprisingly high F1-scores of 0.96 and 0.99 on our evaluation dataset.

Keywords: programming teaching; Vietnamese educational chatbot; Vietnamese natural language processing; Vietnamese question answering system

Cite this article as: Vuong Le Minh Nguyen, Luong Cong Tam, Nguyen Viet Hung, Nguyen Do Thai Nguyen, Luong Tran Hy Hien, Luong Tran Ngoc Khiet, & Phan Thi Trinh (2021). CodEbot – A Vietnamese chatbot system for answering C++ and Python-related questions. *Ho Chi Minh City University of Education Journal of Science, 18*(9), 1711-1723.

1. Introduction

1.1. CodEbot

Vietnam is investing a great deal of money and effort into embracing new technologies of the fourth industrial revolution (IR 3.0) across all sectors. Previous industrial revolutions have been featured by the adoption of various techniques in production. The IR 3.0 is now characterized by a combination of technologies that blurs the lines between the physical, digital, and biological spheres. The rapid technological advancement mainly focused on mobile communication, interconnectivity (internet of things), big data, artificial intelligence, robotics, autonomous vehicle, nanotechnology, biotechnology, and quantum computing. Lying at the intersection of these trending fields of study is the skill of programming, which essentially helps humans to automate a wide variety of repetitive processes in production. As an obvious consequence, the teaching of programming in the early stages of education has become crucially important for the youth to prepare well for their specialization in the future.

In the era of information, fundamental programming materials can easily be acquired from the Internet in the form of web pages (Wikipedia, programming blogs.) or ebooks. However, these materials are rarely written in Vietnamese, so not many Vietnamese students without a solid English background can cultivate these precious sources of knowledge. Furthermore, most of the Vietnamese programming materials often include quite a lot of academic terminologies, which makes them extremely difficult for any beginner to get used to. For these reasons, a structural approach to help Vietnamese students with programming is necessarily required. This led to the idea of having an engaging Vietnamese conversational chatbot as well as a programming knowledge base to answer programming-related questions for students. Therefore, this paper, combining natural language processing techniques with knowledge representation and reasoning techniques, aimed to implement such a question answering chatbot in pure Vietnamese to help students with their programming-related questions. The range of topics the chatbot can cover is limited to C++ and Python, two of the most taught programming languages in Vietnamese colleges and universities.

1.2. Literature Review

1.2.1. Educational Software

Educational software refers to any computer software which is developed to make some part of education more effective and efficient to both teachers and learners. It could be in the form of scientific problem-solving software, language learning software, or even classroom management software.

Some particular examples of educational software are Wolfram Alpha, Mathway, and Microsoft Math Solver (used to be Microsoft Mathematics). These software mostly use predefined reasoning strategies as well as knowledge bases to solve predefined mathematical or scientific problems after some fixed number of steps. One of the biggest drawbacks of these software is that input queries or problems must be strictly formulated or they can never be solved accurately.

Another common form of educational software is online E-learning platforms. These platforms often provide learners with a wide variety of courses and award them with certifications on completion of those courses. Coursera, Udemy, and edX are some of the most popular online E-learning platforms among self-taught students. As for Computer Science, there are also some honorable mentions such as FreeCodeCamp, Codecademy, and DataCamp. For every course on these platforms, learners can watch video lectures, read notes, and practice by solving quizzes or coding problems.

1.2.2. Educational Knowledge Ontology

The knowledge embedded in a learning course can often be expressed using some kinds of knowledge bases. Recently, effort was made to describe the knowledge extracted from an Introduction to Programming course using an ontology called "Rela-model." Combining the ontology "Rela-model" with predefined reasoning rules, a search engine was proposed to rank the most relevant answer to a given query (Nguyen, Pham, Nguyen Le, Tran, & Pham, 2020).

1.2.3. English Chatbot Systems

In recent years, natural language processing (NLP), together with the advancement of deep learning, has achieved state-of-the-art benchmark results on a wide variety of NLP tasks. Consequently, Human – Computer Interaction (HCI) is no longer limited to simple interactions such as sliding or pressing buttons.

"There is a growing interest in chatbots, which are machine agents serving as natural language user interfaces for data and service providers. The most frequently reported motivational factor is "productivity"; chatbots help users to obtain timely and efficient assistance or information. Chatbot users also reported motivations pertaining to entertainment, social and relational factors, and curiosity about what they view as a novel phenomenon" (Følstad & Brandtzaeg, 2017).

Automated chatbot has been an emerging trend for customer cares in various domains. Many virtual assistants have been introduced such as Apple's Siri, Microsoft's Cortana, or Amazon's Alexa. These virtual assistants are efficiently replacing human assistants in many repetitive conversational tasks. More and more companies are integrating chatbots into their customer support channels to reduce stress on customer support service.

1.2.4. Vietnamese Chatbot Systems

Despite the activeness of studies on English chatbot systems, to the best of our knowledge, most Vietnamese chatbot systems are closed source projects. The development of these chatbots was not published in any research journal or conference proceeding. There have been only two published papers on this topic so far. The first one was about developing a lead engagement chatbot in the real estate sector (Quan, et al., 2018). Another one was about developing a high school tutor chatbot, which provides students with hints to their problems instead of any direct solution (Nguyen, Tran, Do, & Pham, 2020).

1.2.5. Intent Classification

In the world of natural language understanding, intent classification is the process of classifying an utterance with respect to the function it serves in a dialogue. Without this NLU step, chatbot systems are unable to drive user conversations properly.

Among many classification algorithms that work well on unstructured data, traditional machine learning algorithms like Support Vector Machine (SVM), Conditional Random Field (CRF) along with deep learning architectures such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) are highly appreciated for their great performance. Typically, in a study on intent classification in Vietnamese, the authors built a model to classify user intent in many domains with a Long Short-term Memory (LSTM), a special implementation of RNN, followed by CRF, which overall achieved very impressive result (Luong, Cao, Le, & Phan, 2017). In the same year study, the authors used a Convolutional Neural Network (CNN) with Softmax layer as the output to classify user intent, which also achieved a solid performance (Ngo, Pham, Takeda, Pham, & Phan, 2017). In another study on intent classification tasks in the real estate domain, the authors used a combination of CNN and Bidirectional Gated Recurrent Unit (Bi-GRU), a modified implementation of the original LSTM, followed by CRF also achieved very good result (Quan, et al., 2018). A recent study using a combination of Bi-LSTM and CNN following with CRF to classify user intent in dialogue also achieved a very good result (Tran & Luong, 2020).

In recent years, deep learning has been widely used in previous research for intent classification and context management in Vietnamese. However, in 2017, a study by Vietnamese researchers compared a classical machine learning algorithm (CRF) to a modern deep learning method, Bi-LSTM – CRF on the same data set, which consists of text messages, showed that there was no big difference in the performance between the above two methods. The F1-Score difference between the traditional machine learning method and the modern deep learning method was no more than 5% (Ngo, Pham, Takeda, Pham, & Phan, 2017). As for the current state-of-the-art deep learning methods, the difference of 5% does no longer hold, but we could still not underestimate the effectiveness of traditional methods. These methods can always serve as a good baseline in text classification problems, particularly in case the size of training data is small as in the intent classification task.

2. Methodology

2.1. Question Answering Pipeline

The input of our system is an utterance, and the output of the system is a response to the utterance, both in Vietnamese. The input utterance will be transformed into features through the process of feature extracting. The extracted features will then be used to classify the intents of a user using machine learning models. Finally, both the original sentence (in natural language) and the results of our machine learning models will be fed into a retrieval-based answering component to get the corresponding response.

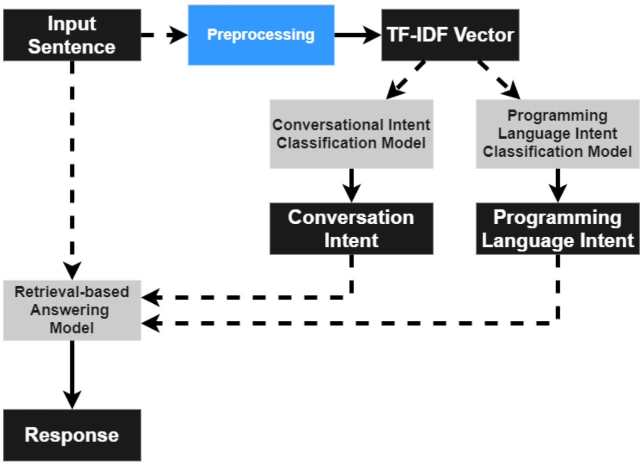

Fig. 1. *Overall pipeline*

2.2. Preprocessing and Feature Extracting

Text data cannot be directly fed into machine learning models. Instead, through the word-embedding process, utterances must first be transformed into vectors of real numbers.

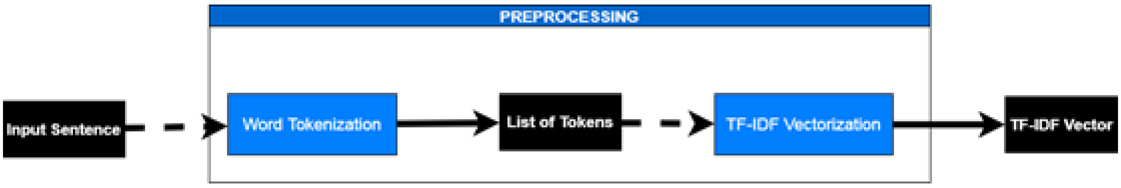

Fig. 2. *Preprocessing and Feature Extracting*

However, before progressing to word-embedding, Vietnamese utterances must be word-segmented properly to improve the overall performance of our language models (Nguyen, Ngo, & Jiamthapthaksin, 2019). English words can be accurately extracted from an utterance just by using whitespaces. The same rule does not hold for the Vietnamese. For instance, the sentence "Mọi học sinh nên được đào tạo lập trình cơ bản từ cấp học phổ thông." should be tokenized as following: "Mọi học_sinh nên được đào_tạo lập_trình cơ_bản từ cấp_học phổ_thông." (English version: "Every high school student should be equipped with fundamental programming skills."). As you may already be aware that, in Vietnamese, a multi-syllable word will contain whitespaces in between their syllables, which can cause misleading if we only treat these syllables independently from each other. For the given reason, word segmentation is an extremely important preprocessing task in Vietnamese NLP. In our experiments, we use the pre-trained word segmentation model from underthesea, the open-source Vietnamese NLP toolkit, to do word segmenting for Vietnamese utterances. The pre-trained segmentation model is a Conditional Random Field model, which was trained in a supervised fashion using annotated data (Underthesea, 2019).

Along with the outstanding growth of both NLP and deep learning, nowadays NLP practitioners have a wide range of word embedding methods such as Bag of Words, TF-IDF, word2vec, doc2vec, or GloVe to choose from. Each method has its advantages and disadvantages. Deep learning embeddings can impressively capture a good generalization of words' meanings. However, the drawback of these methods is that a very large amount of data is required to achieve the expected accuracy. Consequently, in our experiments, to adapt to the limited size of our annotated dataset, TF-IDF, a simple yet effective word-embedding method, will be used to vectorize input utterances for our text classification models.

TF-IDF is the product of two statistics, term frequency (TF) and inverse document frequency (IDF). In our experiments, the TF and IDF are calculated as

$$\mathrm{tf}(\mathrm{t,d}) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{1}$$

$$\mathrm{idf}(t, D) = log\frac{N}{|\{d \in D: t \in d\}|} \tag{2}$$

where $\mathrm{tf}(t, d)$ is the term frequency of term t in document d, $\mathrm{idf}(t, D)$ is the inverse document frequency of term t with respect to the set of documents D, and N is the total amount of documents in the given set D.

TF-IDF of term t in document d with respect to the given set of documents D is calculated as

$$\text{tf-idf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D) \tag{3}$$

2.3. Intent Classification

Since the size of our dataset is limited, the Support Vector Machine is the go-to method for the intent classification task in our experiments.

The Support Vector Machine (SVM) is a binary classification algorithm, in which a hyperplane will be constructed to support the classification of training examples. The distance between the hyperplane and the examples of the two classes will be maximized (maximum-margin hyperplane). The training examples lying on the two margins are called support vectors. During inference, unseen examples will be projected to the same space as the training examples and classified to the proper class based on which subspace of the hyperplane they belong to.

SVM can also be applied to a multiclass classification scheme by effectively applying "one vs. one" or "one vs. the rest" strategy. "One vs. the rest" is often applied in practice because the number of hyperplanes scales linearly to the number of classes in our task. Hence, we use this strategy in our experiments.

When receiving a user utterance, the chatbot will have to recognize the intents lying in the utterance. Thus, we propose two types of intents that need recognizing in user utterances: conversational intents and programming language intents.

There are 11 conversational intents an utterance can belong to: agree, disagree, ask_credit_info, greeting, need_help, ask_references, ask_tip, ask_definition, ask_comparison, ask_application, and out_of_context.

There are three programming language intents an utterance can belong to: general, cpp, and python.

For each of the intent types defined above, we trained a "one vs the rest" SVM to extract the proper intents from user utterances.

2.4. Retrieval-based Answering

From the reasoned intent, we propose an inference engine to decide the next best action based on a set of rules and a knowledge base. A corresponding answer will be randomly drawn from a set of all available responses and return to the user.

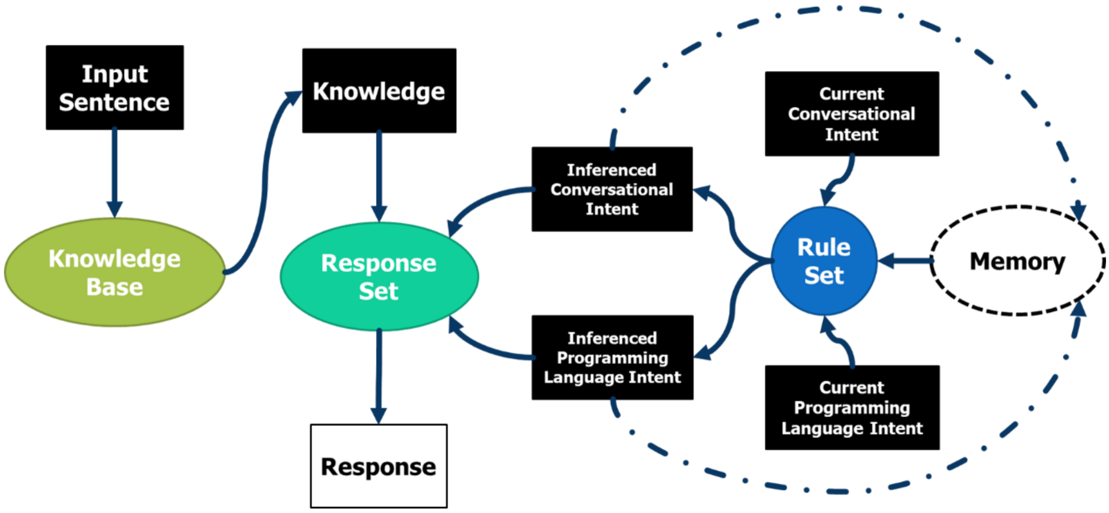

Fig. 3. *Retrieval-based answering*

Ruleset

For each type of extracted intents, we prepared a customizable ruleset consisting of tuples in the form of (I_{t-1}, I'_t, I_t) where $I_{t-1}$is the intent of the previous utterance, I'_t is the intent of the current utterance defined by machine learning models, and I_t is the intent deduced from I_{t-1} and I'_t.

Programming language intent (I_p) rules:

Given $\boldsymbol{P}$ ∈ {**general, cpp, python, low_conf**}

- If $\boldsymbol{I_{p_{t-1}}} \neq \boldsymbol{I'_{p_t}}$ and $\boldsymbol{I_{p_{t-1}}}, \boldsymbol{I'_{p_t}} \in \boldsymbol{P}$, then $\boldsymbol{I_{p_t}} \leftarrow \boldsymbol{I'_{p_t}}$.
- If $\boldsymbol{I_{p_{t-1}}} = \{\boldsymbol{i_p} | \boldsymbol{i_p} \in \boldsymbol{P}\}$ and $\boldsymbol{I'_{p_t}} =$ **low_conf**, then $\boldsymbol{I_{p_t}} \leftarrow \boldsymbol{I_{p_{t-1}}}$.
- If $\boldsymbol{I_{p_{t-1}}} = \boldsymbol{I'_{p_t}} =$ **low_conf**, then $\boldsymbol{I_{p_t}} \leftarrow$ **general**.

Conversational Intent (I_g) rules:

$$\text{Given } \boldsymbol{G} \in \left\{ \begin{array}{c} \textbf{agree, ask\_application, ask\_comparison,} \\ \textbf{ask\_credit\_info, ask\_definition,} \\ \textbf{ask\_references, ask\_tips, disagree,} \\ \textbf{greeting, need\_help, out\_of\_context, low\_conf} \end{array} \right\}$$

- If $\boldsymbol{I_{g_{t-1}}} \neq \boldsymbol{I'_{g_t}}$ and $\boldsymbol{I_{g_{t-1}}}, \boldsymbol{I'_{g_t}} \in \boldsymbol{G}$, then $\boldsymbol{I_{g_t}} \leftarrow \boldsymbol{I'_{g_t}}$.
- If $\boldsymbol{I_{g_{t-1}}} = \{\boldsymbol{i_g} | \boldsymbol{i_g} \in \boldsymbol{G}\}$ and $\boldsymbol{I'_{g_t}} = \{\boldsymbol{i} | \mathbf{i} \in \{$**agree, disagree**$\}\}$, then $\boldsymbol{I_{g_t}} \leftarrow$ **out_of_context**.
- If $\boldsymbol{I_{g_{t-1}}} = \boldsymbol{I'_{g_t}} =$ **low_conf**, then $\boldsymbol{I_{g_t}} \leftarrow$ **out_of_context**.

*** low_conf** *is an additional intent that will be assigned to a statement whenever the intent classifiers are not confident of their predictions (given some fixed threshold).*

Knowledge base

The knowledge base consists of tuples in the form of (E, I_p, I_g, K). E is the programming entity extracted from the original utterance using regular expression. I_p is the programming language tag. I_g is the query type tag, which could be one of the following values: ask_application, ask_comparison, or ask_definition. K is the corresponding knowledge.

Response set

The set of retrievable response patterns consists of tuples in the form of $(\{I_i \in I\}, R)$ where I is a set of available extracted intents, and R is a list of possible response patterns. A random template will be chosen from R to construct the final answer.

Combining the above components, our chatbot will respond to a user's utterance most appropriately and drive the conversation toward the predefined topics of fundamental programming knowledge in C++ and Python.

3. Experimental results

3.1. Dataset

Lying at the heart of any machine learning model is the data it was trained on. In our experiments, to serve our machine learning model, we collected two types of data, and each kind has its own collecting method.

3.1.1. Programming-related Corpus

To build a programming-related corpus, we first collected Vietnamese Wikipedia paragraphs on C++ and Python programming topics. After that, we combined all of them into a large corpus to improve the performance of TF-IDF, a statistical word embedding method to map words, sentences, and documents into vectors of real numbers (term-frequency), which can then be served as inputs for our machine learning models (more details in 0).

The corpus is organized in the form of a plain text so that when building the IDF table in TF-IDF, it will be read sentence by sentence, each sentence corresponds to a "document" in the IDF table.

3.1.2. Intent-annotated Dataset

Besides the corpus for TF-IDF embedding, we prepared a human-labelled dataset to serve the training of our machine learning models on the intent classification task.

The dataset consists of 700 utterances with the respective human-annotated intents. A large portion of the dataset was contributed by students from the Faculty of Information Technology at Ho Chi Minh University of Education. The remaining examples were crawled from many programming-related forums and programming-related groups on Facebook. To fine-tune the machine learning models in our favor, a small portion of the dataset was added manually during experimenting.

3.2. Evaluation Metric

The performance of our chatbot mostly relies on the performance of the intent classification task. Therefore, measuring how accurate our chatbot could define user's intent also indirectly evaluate the overall performance of our chatbot.

In most real-life classification problems, the imbalanced target distribution is often a problem that also needs addressing. However, Accuracy does not consider how the labels are distributed, which might lead to an incorrect conclusion. F1-Score, on the other hand, measures the False Positives and False Negatives in model predictions by taking the harmonic mean of Precision and Recall. This effect is very useful for the case of intent classification where user intents are usually not uniformly distributed.

F1-Score is the harmonic mean between **Precision** and **Recall**:

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4}$$

Precision is the model's accuracy when predicting Positive class, which is determined by the following formula:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{5}$$

Recall is the model's sensitivity to predict the Positive class, which is determined by the following formula:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{6}$$

3.3. Intent Classification

3.3.1. Experimental Setup

We train our SVM models using the implementation of LinearSVC from scikit-learn (Pedregosa, et al., 2011).

3.3.2. Results

Using the aforementioned training setup, we do cross-validation on an 80% – 20% split of our intent annotated dataset to get the experimental results.

Table 1. *The results of the conversational intent classification (average F1-Score: 0.96)*

Intent	Precision	Recall	F1-score
agree	1.00	0.86	0.92
ask_application	1.00	1.00	1.00
ask_comparison	1.00	1.00	1.00
ask_credit_info	1.00	1.00	1.00
ask_definition	0.80	1.00	0.89
ask_references	0.86	1.00	0.92
ask_tips	1.00	1.00	1.00
disagree	1.00	1.00	1.00
greeting	1.00	1.00	1.00
need_help	1.00	0.80	0.89
out_of_context	0.90	1.00	0.95

Table 2. *The results of the languauge intent classification (average F1-Score: 0.99)*

Intent	Precision	Recall	F1-score
cpp	1.00	0.94	0.97
python	1.00	1.00	1.00
general	0.98	1.00	0.99

3.3. CodEbot Demo Application

A web application for the demonstration purpose of our chatbot system was developed on top of Bootstrap Framework, which supports responsiveness across a variety of screen sizes (PC, laptop, smartphone, tablet, etc.). A knowledge management web portal was also implemented to aid system admin in adding, removing, and modifying the chatbot's knowledge.

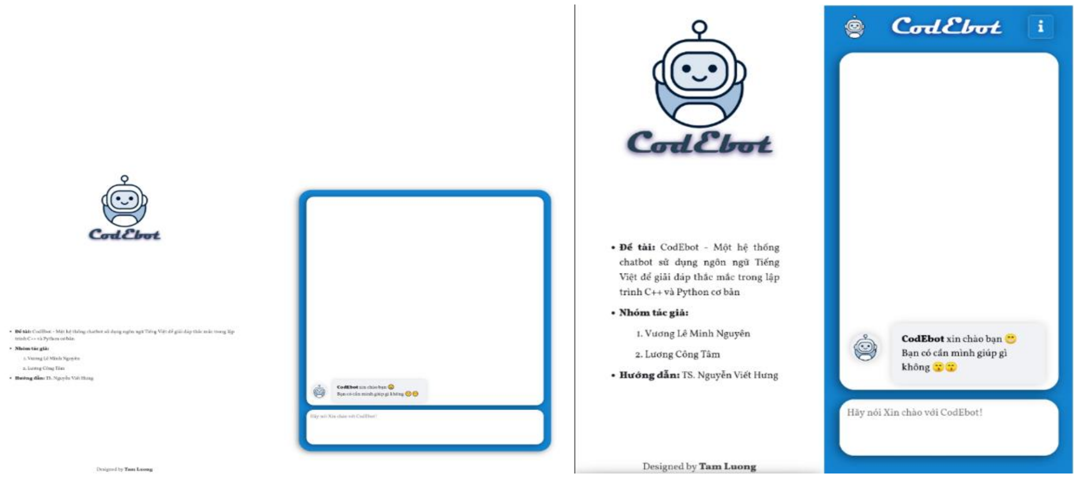

__Fig. 4.__ CodEbot DEMO on computer's screen (left) and smartphone's screen (right)

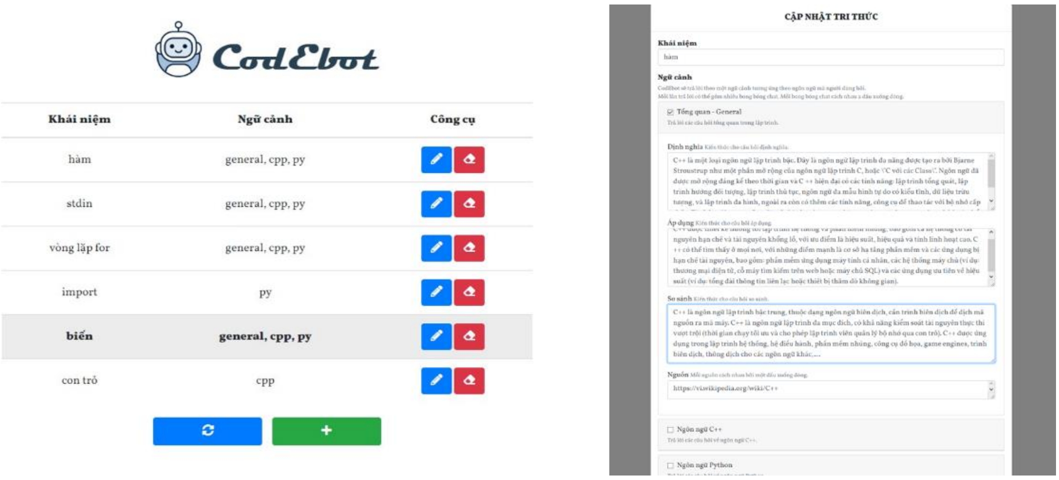

__Fig. 5.__ Knowledge Management – Knowledge List (left) and Knowledge Editor (right)

4. Conclusion

Despite the impressive experimental results, we believe that there might be sampling bias in our annotated dataset. For instance, Internet slangs were not included in the dataset, which in real life could cause a huge decrease in the performance of chatbots. Furthermore, the proposed knowledge base was too simple to represent the true complexity of domain knowledge. These limitations should be addressed in a future study on this matter.

❖ ***Conflict of Interest:*** *Authors have no conflict of interest to declare.*

REFERENCES

Følstad, A., & Brandtzaeg, P. (2017). Chatbots and the new worlds of HCI. *Interactions*, 38-42.

Luong, L. T., Cao, S. M., Le, T. D., & Phan, H. 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) (pp. 215-220). IEEE.

Ngo, L., Pham, L., Takeda, H., Pham, S., & Phan, H. (2017). *On the Identification of Suggestion Intents from Vietnamese Conversational Texts*. ACM International Conference Proceeding Series (pp. 417-424).

Ngo, L., Pham, S., Pham, L., Phan, H., & Son, C. (2018). *Dialogue act segmentation for Vietnamese human-human conversational texts.* 2017 9th International Conference on Knowledge and Systems Engineering (KSE) (pp. 203-208). IEEE.

Nguyen, H. D., Pham, V., Nguyen Le, V., Tran, T., & Pham, X. (2020). *Build a search engine for the knowledge of the course about Introduction to Programming based on ontology Rela-model.* 2020 12th International Conference on Knowledge and Systems Engineering (KSE*)*, 207-212.

Nguyen, H. D., Tran, D., Do, H., & Pham, V. (2020). Design an Intelligent System to automatically Tutor the Method for Solving Problems. *International Journal of Integrated Engineering*, 211-223.

Nguyen, S., Ngo, Q., & Jiamthapthaksin, R. (2019). State-of-the-Art Vietnamese Word Segmentation. *arXiv:1906.07662 [cs.CL]*, 1-6. Retrieved from https://arxiv.org/abs/1906.07662

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830.

Quan, T., Trinh, T., Ngo, D., Phan, 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 2018), (pp. 357-359).

Shah, H., Warwick, K., Vallverdu, J., & Wu, D. (2016). Can Machines Talk? Comparison of Eliza with Modern Dialogue Systems. *Computers in Human Behavior*, 278.

Smutný, P., & Schreiberova, P. (2020). Chatbots for learning: A review of educational chatbots for Facebook Messenger. *Computers and Education*, 103862.

Tran, O., & Luong, T. (2020). Understanding what the users say in chatbots: A case study for the Vietnamese language. *Engineering Application of Artificial Intelligence*, 103322.

Underthesea (2019). *Underthesea*. Retrieved from Underthesea: http://undertheseanlp.com/

Winkler, R., & Söllner, M. (2018). Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis. *Academy of Management Proceedings*, 15903.

CODEBOT – MỘT HỆ THỐNG CHATBOT TIẾNG VIỆT TRẢ LỜI CÁC CÂU HỎI LIÊN QUAN ĐẾN LẬP TRÌNH C++ VÀ PYTHON

Vương Lê Minh Nguyên[1*], Lương Công Tâm[1], Nguyễn Viết Hưng[1], Nguyễn Đỗ Thái Nguyên[1], Lương Trần Hy Hiến[1], Lương Trần Ngọc Khiết[1], Phan Thị Trinh[2]

[1]Trường Đại học Sư phạm Thành phố Hồ Chí Minh, Việt Nam

[2]Trường Cao đẳng Công nghệ Thủ Đức, Việt Nam

[*]Tác giả liên hệ: Vương Lê Minh Nguyên – Email: nguyenvlm.43@student.hcmue.edu.vn

Ngày nhận bài: 12-4-2021; ngày nhận bài sửa: 03-6-2021; ngày duyệt đăng: 03-9-2021

TÓM TẮT

Với cách mạng công nghiệp 4.0, lập trình trở thành một trong những kĩ năng thiết yếu giúp giới trẻ có ưu thế cạnh tranh trong nhiều lĩnh vực. Kĩ năng lập trình không chỉ cần thiết cho phát triển phần mềm mà còn hữu ích cho thống kê, phân tích, xây dựng mô hình toán học trong các lĩnh vực khác. Dù vậy, tài liệu lập trình trên Internet hầu hết sử dụng tiếng Anh khiến cho giới trẻ Việt gặp nhiều khó khăn khi tiếp cận. Nhận thức được điều đó, nghiên cứu này đã kết hợp xử lí ngôn ngữ tự nhiên với biểu diễn và suy luận tri thức nhằm xây dựng một chatbot trả lời các câu hỏi liên quan đến lập trình bằng tiếng Việt. Nghiên cứu đề xuất một phương pháp biểu diễn tri thức đơn giản để tích hợp tri thức bên ngoài cùng với một phương pháp suy diễn tri thức và truy xuất trả lời câu hỏi nhằm đưa ra phản hồi phù hợp từ truy vấn đầu vào. Chatbot được giới hạn tri thức với hai ngôn ngữ lập trình là C++ và Python vốn được giảng dạy phổ biến trong các trường cao đẳng và đại học tại Việt Nam. Hai mô hình máy học dự đoán ý định người dùng được huấn luyện và đánh giá trên tập ngữ liệu có gán nhãn do sinh viên Khoa Công nghệ Thông tin, Trường Đại học Sư phạm Thành phố Hồ Chí Minh đóng góp. Kết quả thử nghiệm của hai mô hình khả quan, đạt F1-score lần lượt 0.96 và 0.99 trên dữ liệu kiểm thử.

Từ khóa: dạy học lập trình; chatbot tiếng Việt trong giáo dục; xử lí ngôn ngữ tự nhiên tiếng Việt; hệ thống trả lời câu hỏi tiếng Việt