VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY
INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DEVELOPING A CHATBOT USING NLP - NEURAL NETWORK TECHNIQUES AND INTEGRATING SPEECH RECOGNITION FOR SUPPORTING ADVISING EDUCATION ENROLMENT

By

LY MINH TRUNG

Supervisor: Dr. Nguyen Thi Thanh Sang

A thesis submitted to the School of Computer Science and Engineering in partial fulfillment of the requirements for the degree of Bachelor of Data Science

Ho Chi Minh City, Vietnam
2023

DEVELOPING A CHATBOT USING NLP - NEURAL NETWORK TECHNIQUES AND INTEGRATING SPEECH RECOGNITION FOR SUPPORTING ADVISING EDUCATION ENROLMENT

APPROVED BY:

____

____

____

____

____

____

THESIS COMMITTEE
(Whichever applies)

Acknowledgement

The School of Computer Science and Engineering lecturer Dr. Nguyen Thi Thanh Sang, who served as my mentor during my thesis, has my sincere gratitude for her helpful advice, support, and encouragement. Her extensive knowledge and skill in data science have greatly influenced the course of my research and improved the quality of my thesis.

I would also like to thank Thesis grading committee and the School of Computer Science and Engineering for their insightful feedback and constructive criticism, which have been immensely helpful in refining my ideas and improving the overall quality of my thesis.

I am also grateful to my colleagues and friends for their unwavering support and encouragement, especially during the most challenging times of my research. Their enthusiasm and encouragement have been a constant source of motivation for me.

Finally, I would like to acknowledge my family for their love, support, and encouragement throughout my academic journey. Their unwavering support and encouragement have been the driving force behind my success.

Table of Contents

List of Figures

List of Tables

Abbreviations

1. **DNN**: Deep Neural Network
2. **ANN**: Artificial neural networks
3. **QA**: Question Answering
4. **QAS**: Question Answering System
5. **AI**: Artificial Intelligence
6. **NLP**: Natural Language Processing
7. **URL**: Uniform Resource Locators
8. **GPU**: Graphics processing unit
9. **CPU**: Central Processing Unit
10. **CDQA**: Closed Domain Question Answering System

Abstract

The question-answer system is now widely used and helps individuals by automatically responding to frequently asked questions in a variety of industries. These systems, however, are dependent on the user's context, training data, and the learning strategies used. As a result, creating such a solid dataset and comprehensive contexts is required, however it is a difficult task. Through numerous layers of inquiries, Deep Neural Network may assist in inferring semantic information and offer useful responses to user enquiries. This thesis suggests a novel Deep Neural Network-based chatbot model to automatically produce appropriate contextual answers. The suggested chatbot includes a case study on admissions counseling at International University-Vietnam National University in Ho Chi Minh City. To see how the system responds to various inquiries, three experiments were run. The first test looks at the loss experienced when the DNN model is being trained, the second test is based on a survey and evaluation of nearby users, and the third test is based on a practical test consisting of 40 questions covering a variety of topics. varying degrees of difficulty and pass judgment. The main function of the chatbot is to answer questions related to international university admissions accurately and clearly according to the context, moreover with the integration of voice asking will make it easier for users. According to test results, a chatbot powered by Deep Neural Network can provide in-depth responses. The outcomes are examined to demonstrate the viability and potential of the suggested chatbot.

Chapter 1 Introduction

1.1 Background

Automatic supports, such as communicating with consumers or online users, are essential in social networking sites and help draw in additional people. In social networks nowadays, a chatbot is not odd; it might be a buddy, a consultant, or an assistant who solves difficulties in a certain sector. In other words, a chatbot has the capacity to comprehend human speech, engage in conversation, and carry out duties. It is utilized in apps that allow automated verbal interactions in natural language processing. For instance, a chatbot [1] was created using the social networking platform of Meta AI to respond to inquiries about the details of a certain sales page on Facebook and even engage in conversation. Customers now find it simple to obtain the things mentioned on Facebook, especially given how common internet transactions are these days. These sentient beings may interact with people in a variety of ways depending on the architecture they are using, whether it be to give instructions, respond to inquiries, or amuse them. QA, or Mining Question Answering, is an essential building component for assessing linguistic competence in artificial intelligence algorithms. In recent years, automated models for managing QA duties have made enormous strides. On several common QA datasets, automated intelligence models can currently comprehend language and information better than humans. This is partly because deep neural networks are becoming more complicated, and because pre-trained language models are transferring information unsupervised from enormous corpora of data to construct meaningful word representations [2].

1.2 Problem statement

The model trained on a meticulously compiled and processed database performs better than the open source chatbot models, but it needs a lot of information to provide a satisfactory response. designed with great care for model training. Since this type of data is frequently unavailable to begin with, domain data must be gathered to create a knowledge base for a chatbot model, such as a Deep Neural Network-based chatbot. This thesis therefore focuses on developing a domain knowledge base for a closed domain chatbot using a case study of university admissions counseling offering a chatbot approach for admissions guidance at the International University (IU) - Vietnam National University, Ho Chi Minh City (VNU-HCM). Based on data from the IU college admission website, an experimental chatbot model was created. The model can provide accurate responses in accordance with the information the user needs to ask based on the questions and answers that have been retrieved. Along with that, the chatbot will be integrated with a voice input feature through Google API Speech Recognition with relatively high accuracy in Vietnamese.

1.3 Scope and Objectives

The thesis focuses on developing a Deep Neural Network-based chatbot that can assist in automatically responding to frequently requested inquiries about International University – Vietnam National University, Ho Chi Minh City admissions. The International University's (VNU HCMC) admission statistics is the source of the input data. The suggested chatbot model is furthermore dynamically created, allowing for future expansion by the addition of new data to the input data table to expand the knowledge base. In addition, Chatbot can recognize voices and answer questions from it by applying Google API Speech Recognition. The chatbot's functionality will be substantially enhanced in this way. The datasets are uploaded and shared on Drive. It can be reached with the link in the footer of this page.

1.4 Thesis structure

Chapter 1 – Introduction: states the background and the focus of this work.

Chapter 2 – Literature review: The techniques used are listed briefly. An overview about DNN pre-train models.

Chapter 3 – Methodology: The technical details of the DNN model and the improvement made.

Chapter 4 – Experiments and Discussion: The experimental settings, the evaluation results, and discussion on the results is described in this chapter.

Chapter 5 – Conclusion and Future work: Conclude the main points of this thesis.

Dataset:
https://drive.google.com/file/d/1wKzc1vXI92AEVYSOvWFTocWafLRXFdpj/view?usp=share_link
https://drive.google.com/file/d/1tRewJS_GEHsFuxbb2u7kxN8PJ-VgngXT/view?usp=share_link
https://drive.google.com/file/d/1tRewJS_GEHsFuxbb2u7kxN8PJ-VgngXT/view?usp=share_link

Chapter 2 Literature Review

2.1 Theoretical Background

2.1.1 Deep Neural Network

[3] A deep neural network is a neural network with more than two layers and a specified amount of complexity. Deep neural networks handle data in intricate ways by using advanced mathematical models.

According to [3], Millions of artificial neurons are connected in numerous buried layers of deep neural networks, also known as deep learning networks. A weight, which is a numerical value, is used to express the connections between each node. The weight is positive if one node stimulates the other; the weight is negative if the other is inhibited. As the weight values of a node increase, so does the impact that node has on other nodes.

Deep neural networks may theoretically translate any input type to any output type. In contrast to other machine learning techniques, they also require significantly more training. Instead of the hundreds or thousands of instances that a simpler network could require, they need millions of samples of training data.

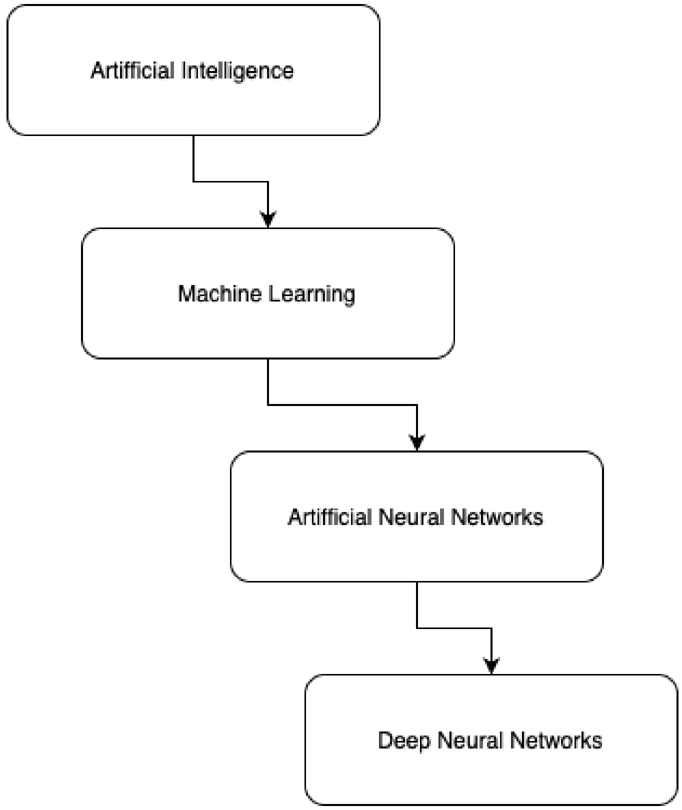

Figure 2. 1 The evolution to Deep Neural Networks (DNN)

First, machine learning (ML) needs to be developed. To increase prediction accuracy, machine learning (ML) provides a framework for automating statistical models like a linear regression model (through algorithms). A model is a single hypothesis that makes predictions about anything. These predictions are somewhat accurate. A machine learning model that makes mistakes adjusts the weights of the model to develop a model with fewer errors.

Due to the learning process required in creating models, artificial neural networks (ANNs) were developed. ANNs employ the hidden layer to store and evaluate the significance of each input to the output. The hidden layer maintains note of each input's importance and connects the importance of input combinations.

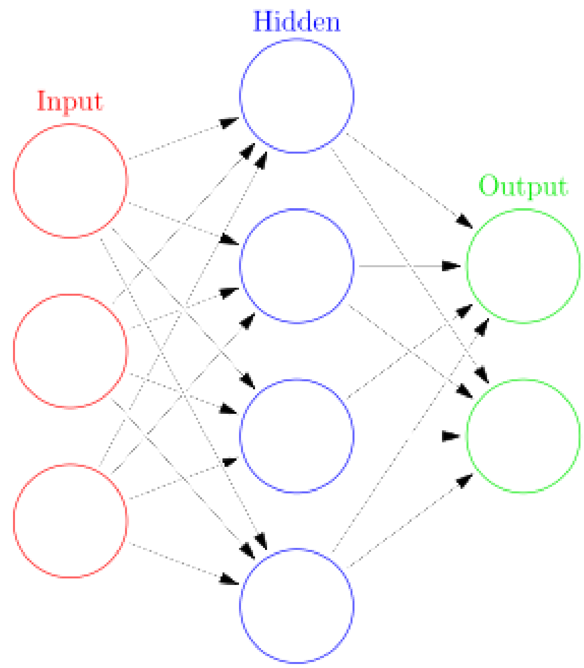

Figure 2. 2 One hidden layer is considered an ANN

A basic neural network has three layers of artificial neurons connected:

- Input Layer: An artificial neural network's input layer is where outside data enters the system. Before being transferred to the layer below, data is processed, examined, or organized by input nodes.
- Hidden Layer: Hidden layers get input from the input layer or from other hidden layers. Artificial neural networks may contain numerous hidden layers. Each hidden layer assesses and improves the output from the layer before sending it to the one below.
- Output layer: The output layer displays the results of the data processing performed by the artificial neural network. There could be one or many nodes in it. One output node in the output layer will show whether the result is 1 or 0, for example, if the classification task is binary (yes/no). However, if our classification problem includes multiple classes, the output layer can contain many output nodes.

Therefore, the ANN component is used in deep neural networks. They contend that since the hidden layer is so effective at improving a model (since each node in the hidden layer makes associations and rates the relevance of the input to determining the output), why not stack more of these on top of each other and benefit even more from it?

Consequently, there are numerous covert levels in the deep net. When there are numerous layers in a model, it is considered to have "deep" layers [4].

Neural networks training: Neural network training is the process of teaching a neural network to do a task. Neural networks initially process several large collections of labeled or unlabeled data. They can handle undefined inputs more precisely by utilizing these examples. In supervised learning, labeled datasets that already know the right answer are fed to artificial neural networks. For instance, training a deep learning network on a simple chatbot model that initially analyzes hundreds of thousands of letters, each of which is described by a different keyword linked to the word's ethnic origin, location, or emotion. These data sets are gradually used to train the neural network, which predicts the right response. Following training, the network starts to infer answers or contexts that it has never encountered before.

2.1.2 TensorFlow for creating Deep Neural Network model

- Arguments:

  + Network: Neural Network to be used.

  + Tensorboard_verbose: Summary verbose level; accepts tensorboard logs of various verbosity levels.

  + Tensorboard_dir: Tensorboard logs directory

  + Path to model checkpoint storage, checkpoint_path. If None, there will be no model checkpoint stored.

+ max_checkpoints: Maximum number of checkpoints. best_checkpoint_path: Path to store the model when the validation rate achieves its peak for the current training session and is higher than best_val_accuracy. No limit if None

+ session: a running operations session. If there isn't one, one will be made. Variables must be initialized before supplying a session; otherwise, an error will be raised.

- Attributes: trainer, predictor, and session

2.1.3 Google Speech Recognition [6]

Google Speech Recognition's speech-to-text technology is based on a combination of several fields of study and techniques, including:

- Signal processing: The first step in speech recognition is to capture and process the audio signal. Signal processing techniques are used to filter out noise and enhance the speech signal.
- Acoustic modeling: Acoustic modeling is used to create statistical models of speech sounds and their variations in different contexts. This involves analyzing the spectral and temporal characteristics of speech signals and creating models that can recognize different phonemes, or units of sound.
- Language modeling: Language modeling is used to predict the probability of words or phrases based on their context. This involves analyzing the frequency and sequence of words in a language and creating models that can predict the most likely words or phrases given a particular context.
- Hidden Markov Models (HMMs): HMMs are statistical models that are widely used in speech recognition. They model the probability distribution of speech sounds and their transitions from one sound to another.

- Deep learning: Deep learning is a powerful machine learning technique that is used in many modern speech recognition systems, including Google's. It involves training artificial neural networks to recognize patterns in data, allowing the system to learn from large datasets of speech and text.

In Google Speech Recognition, all these techniques are combined to create a highly accurate speech-to-text system. The system is trained on large datasets of speech and text to improve its accuracy and can adapt to different speakers and accents. It has become a widely used technology for applications such as voice assistants, speech-to-text transcription, and voice search.

2.1.4 Structure Language Model

Structured Language Models attempt to establish a hierarchy for the words in a sentence. This combined with n-grams will give results that cover the context of the whole sentence, improving the accuracy of the model.

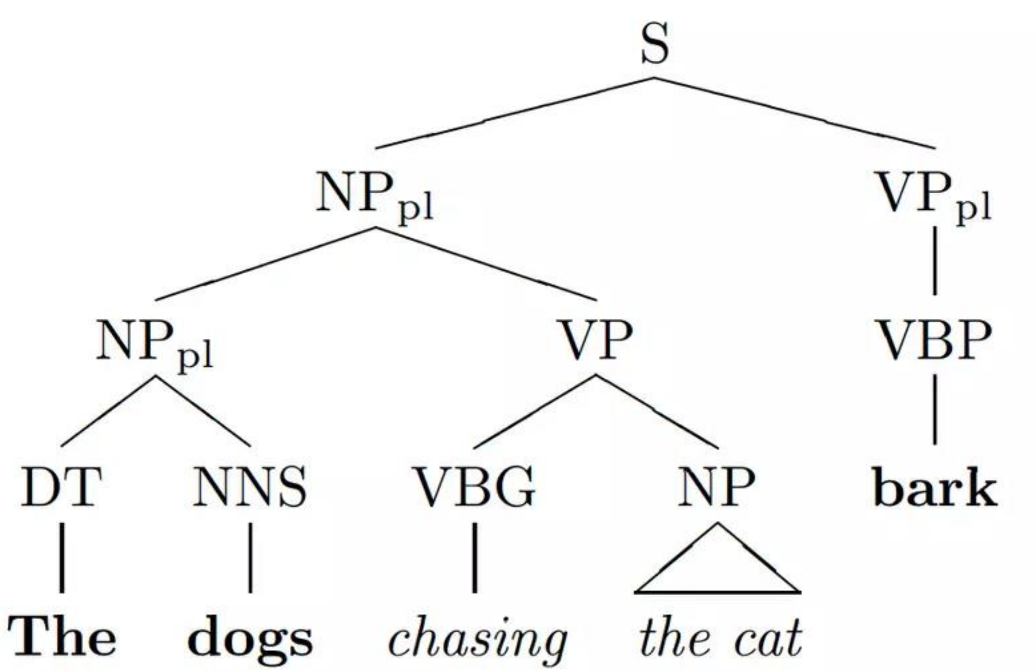

Figure 2. 3 Structure Language Model

2.1.5 Neural Network Language Model

Neural Network Language Models are the latest methods based on Neural networks to build language models; these models are also known as Continuous-space language models. Neural Network Language Model is also divided into 2 main approaches.

- Feed-forward neural network-based LM: Feed-forward neural network is proposed to solve another problem of N-grams related to data sparsity (when there are many sentences and phrases, but they do not exist). collected in train data). The neural network used here is a 3-layer neural network, aiming to learn the necessary parameters to calculate the probability.
- Recurrent Neural Network Based LM: Recurrent Neural Network focuses on solving another aspect of the context of the entire sentence, helping to overcome the limitations of context. Currently, language models based on RNN or developments from it such as LSTM, ... are achieving State of the art results.

According to [7], If with the Feed-forward neural network model, the input data needs to require a fixed number of words, however, this is not possible in practice, due to the different short sentences, the Recurrent Neural Network also overcome this drawback by accepting input of any length.

2.1.6 Deep Neural Network in case of Speech Recognition [8]

In the case of Google Speech Recognition, the deep neural network is trained on large datasets of spoken language. During the training process, the network is presented with many examples of audio input and their corresponding transcriptions. The network learns to recognize patterns in the audio input that are associated with specific phonemes, words, and sentences.

When a user speaks into a microphone and the audio input is sent to the Google Speech Recognition service, the deep neural network analyzes the input in real-time. The network breaks the audio input down into small segments and determines the most likely phonemes that make up the spoken word. These phonemes are then combined to form words, and the words are combined to form sentences.

The deep neural network used by Google Speech Recognition is designed to be highly accurate and robust, even when dealing with noisy or distorted audio input. It is also designed to be able to recognize a wide variety of accents and dialects, making it useful for users all over the world.

2.2 Existing Deep Neural Network-based Chatbot models

DNN is a way to determine the meaning of a statement. DNN-based chatbot models were created as a result. Additionally, it is focused on domain expertise to enable domain-driven interactions. DNN is used to store and search via domain knowledge. As a result, this DNN-based knowledge base may offer data to produce dialog replies. "Randomly extracting responses from labeled groups" is the advantage of the DNN-based method. DNN has been included into closed domain chatbots because of its advantages, and it offers incredibly detailed responses to inquiries.

There are several developing Chatbot varieties. The majority of chatbots do three basic tasks: analyze the inquiry, look for the answer in a document or database, and extract the result. Their objective is to provide consumers with natural language responses using their own terminology. A semantic knowledge base, an unstructured free text database, or a structured database can all be utilized as the Chatbot's database [9]. In which a DNN-based Chatbot builds a semantic knowledge base using a neural network that can semantically infer data from a DNN-based knowledge domain to respond to user inquiries. These systems have the advantage of not requiring the data to be divided into training, testing, and assessment phases. Users can naturally ask queries in a certain domain consequently. Currently, there are not many applications of this DNN model for chatbots but having some closed domain question answering systems like CDQA for legal [10] or for banking [11], on search portals there are only studies on its effectiveness for a small dataset of less than 50 samples. In terms of chatbots for admissions consultants, there are only a few other models, such as Ontology.

Ontology has recently been suggested as the basis for a counseling system for educational programs. Using a keyword-based text search engine, it outperformed the Apache Lucene system in terms of performance [9]. Nowadays, university admissions counseling or advising is extremely important since many prospective students want to know how to select an appropriate major from among the various majors offered by institutions. Particularly in Vietnamese colleges, admissions counseling takes a lot of time and effort. Advisors must continuously update admissions data to assist future students in selecting the best academic program. In Vietnam's universities, there hasn't yet been a chatbot to support these positions. Therefore, it is crucial for Vietnamese institutions to create a QA system for advising university admissions. So, in this graduation thesis, a chatbot model based on DNN will be published to compare the results with the Ontology-based model in the field of admission counseling of International Universities (IU-VNU).

2.3 Closed Domain Question Answering System

A closed domain question answering system is a type of artificial intelligence (AI) system that is designed to answer questions within a specific domain or topic area. These systems are trained to understand and extract information from a particular set of documents or knowledge sources related to a specific subject, such as medicine, law, or finance.

Unlike open domain question answering systems, which can answer questions on any topic, closed domain question answering systems are designed to provide more accurate and precise answers within their specific area of expertise. They typically rely on natural language processing (NLP) techniques, such as named entity recognition and semantic analysis, to understand the meaning of the questions and the knowledge contained in the documents.

Closed domain question answering systems can be used in a variety of applications, such as customer support chatbots, educational tools, and expert systems for specific industries. They can help users quickly find the information they need, without having to manually search through a large amount of data.

A closed domain QA system typically consists of several components that work together in a pipeline to process questions and generate answers. The pipeline typically includes the following stages:

- Question parsing: The system parses the user's question to understand its structure, identify keywords and entities, and determine the type of question being asked.
- Information retrieval: The system searches a database or other sources of information to find relevant data that can be used to answer the question.
- Answer generation: The system uses the retrieved information to generate a candidate answer to the question.
- Answer selection and ranking: The system selects the best answer from the candidate answers generated in the previous stage and ranks them based on their relevance and confidence level.
- Answer presentation: The system presents the selected answer to the user in a suitable format, such as text, speech, or a visual display.

Overall, closed domain QA systems and pipelines are powerful tools for automating the process of answering questions within a specific domain, providing fast and accurate responses to users.

Chapter 3 Method and Implementation

3.1 Chatbot Framework

The proposed chatbot framework, namely Chatbot IU_DNN, includes two functions, including one main function of text input and functions of voice input (Figure 3.1). DNN construction is the main process step, DNN for QA of university admissions information is built to understand input queries in natural language and infer relevant things to give appropriate answers based on most of the trained responses. The following subsections provide more details on the proposed framework.

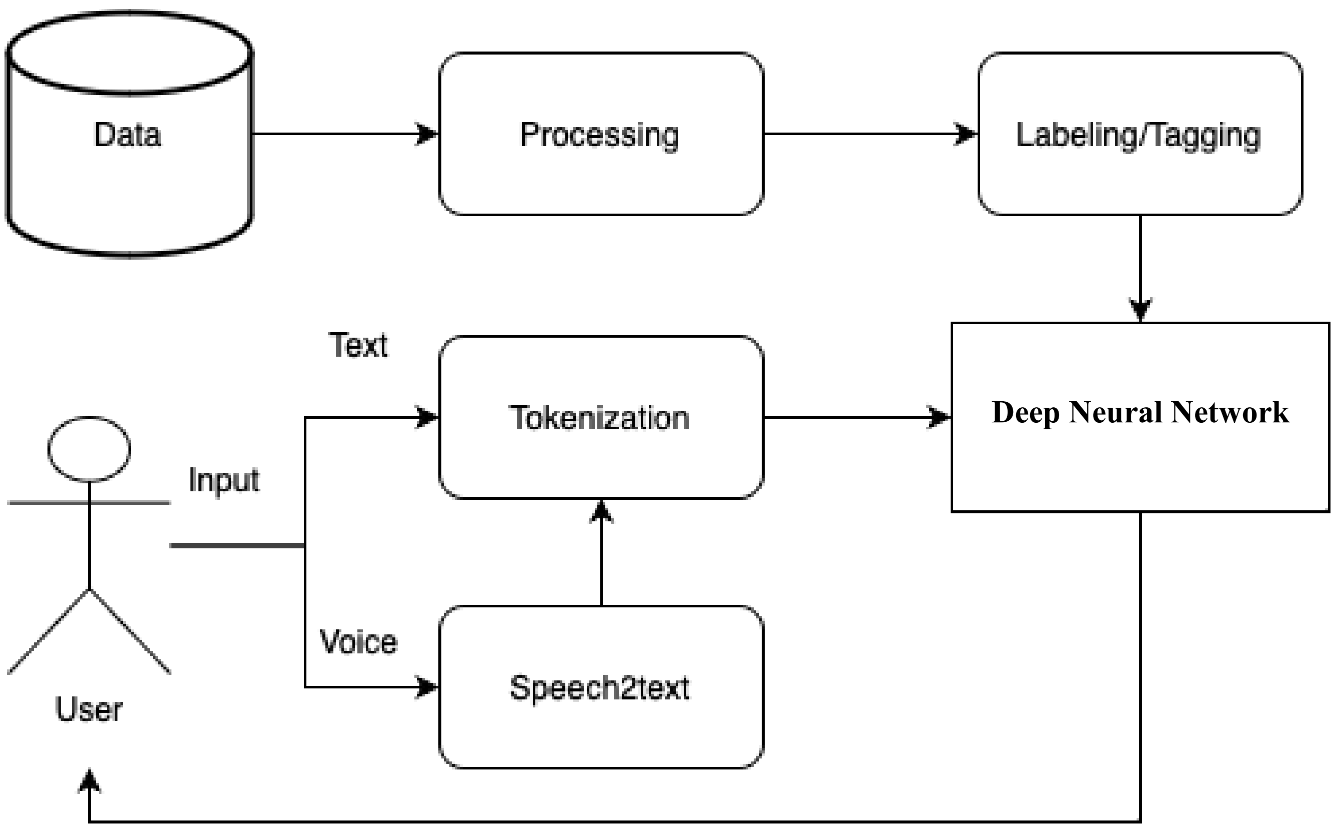

Figure 3. 1 General framework

3.2 Deep Neural Network Framework

Referring the model using deep learning in [14], this study proposes a framework for the chatbot model (question answering system - QAS) of university admission advising as Figure 3.2.

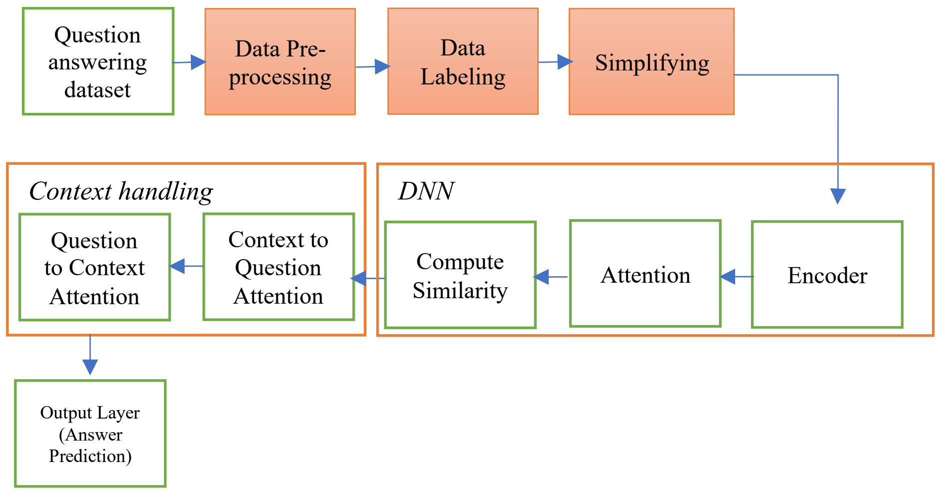

Figure 3. 2 The proposed framework of the DNN

Different from [14] using Embedding Layer, we propose the processes of Data Pre-processing, Data Labeling, Simplifying steps instead. The input data is a question answering dataset which will be processed through these steps before, encoding in order to be learned by DNN. To predict answers effectively fitting the conversation context, the context handling process is proposed and performed. Moreover, an algorithm of answer generation is also built to rank and find top-n suitable answers. The following presents the details of this proposal.

3.3 Data preprocessing

3.3.1 Original data

The Original data includes two columns “Question” and “Answer” that collected on “IU Tư vấn tuyển sinh” website. The dataset has not been processed or labeled, it only consists of raw questions and answers. This dataset is partly provided by the instructor, and my contribution is to collect more diverse questions and answers.

Table 1: Question Types

Question types	Example	Samples
Introductory question about a certain field trained at the university.	Cho tôi hỏi về ngành Khoa học dữ liệu? *(May I ask about Data Science?)* Cho tôi hỏi về ngành Khoa học Máy tính? *(May I ask about Computer Science?)*	120
Questions about the main function of a certain department or department at the university.	Chức năng của khoa Quản trị Kinh doanh là gì? *(What is the function of the School of Business Administration?)*	50
Questions about the standards and admission criteria of each major trained at university.	Điểm chuẩn ngành Kỹ thuật Y sinh là bao nhiêu? *(What is the admissions score of the Biomedical Engineering major?)* Chỉ tiêu ngành Ngôn ngữ Anh là bao nhiêu? *(What is the admissions quota of the English Linguistics and Literature major?)*	100
Questions about general information such as tuition fees, training majors,…	Trường đào tạo những chuyên ngành nào? *(What majors does the school offer?)* Học phí trung bình của trường là bao nhiêu? *(What is the average school fee?)* Khối A00 có ngành gì? *(What major does A00 have?)*	110
The question about what subjects you like to study or what activities you like to do, which major should you choose?	Tôi giỏi sinh học thì nên chọn ngành gì? *(I am good at biology, what major should I choose?)* Tôi thích lập trình thì chọn ngành gì? *(I like programming, what major should I choose?)* Tôi muốn học môn sức bền vật liệu thì chọn ngành gì? *(I want to study the strength of materials, which major should I choose?)*	250
Questions about the entrance exam and some extra questions	Trường có những phương thức tuyển sinh nào? *(What admission schemes do the university have?)*	15

3.3.2 Data after preprocessing (Regular Expression)

The data, after being preprocessed, will remove the characters that are cloned and misspelled in Vietnamese. Some digits are misplaced and there are lots of spaces between words.

The main method that makes data preprocessing easier and more precise is to use Regular Expression:

- Regular expressions (regex) in Python are a powerful tool for matching and manipulating text data. They provide a way to search for patterns in strings, extract specific information from text, and replace or modify parts of a string.
- In Python, the re module provides support for regular expressions. To use regular expressions, you first need to create a regular expression pattern using special syntax that defines the pattern you want to match. Then, you can use functions like re.search(), re.findall(), and re.sub() to search for and manipulate text based on the pattern.
- Regular expressions can be combined and nested to create complex patterns for matching and manipulating text. They are widely used in text processing and data analysis tasks, such as data cleaning, web scraping, and natural language processing.
- Some characters need to be omitted such as: -:´`‘’̣ ̉,;–… \n

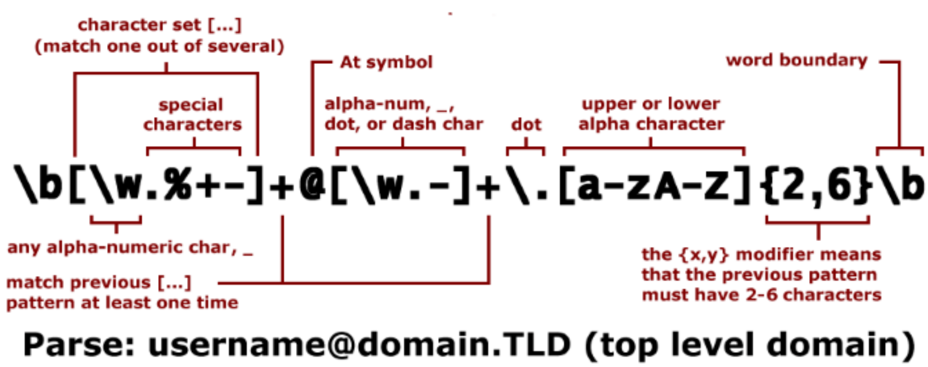

Figure 3. 3 Power of Python Regular Expressions

In addition, the data preprocessing also helps with the data labeling process in the next step. Because when you have removed unnecessary numbers, as well as paragraphs that are not grammatically correct, the word segmentation will become more accurate. Data preprocessing is an essential step in preparing data for machine learning models, including labeling data, regardless of the language. It involves transforming raw data into a format that can be easily understood and utilized by the model in the case of Vietnamese language data [12].

- Cleaning: Preprocessing can remove irrelevant or redundant information, such as stop words, punctuation, or special characters, which can reduce the amount of data to be labeled and improve the accuracy of the labeling process.
- Tokenization: Vietnamese language relies heavily on word segmentation, which means dividing the text into meaningful units. Data preprocessing can tokenize the text by identifying and separating words, phrases, or sentences, which can provide the labeled data in a more structured and readable format for the model.

3.4 Data Labeling

Before creating a label column for the dataset, we will pass the question column through the stop work removal process. Those stop-words include Vietnamese stop-words and a few words specific to the field of higher education. we will label each question with the content of the question itself but remove unnecessary words and may affect the process of identifying question types while modeling. For example, the process of labeling a question is depicted as shown in Figure 3.4.

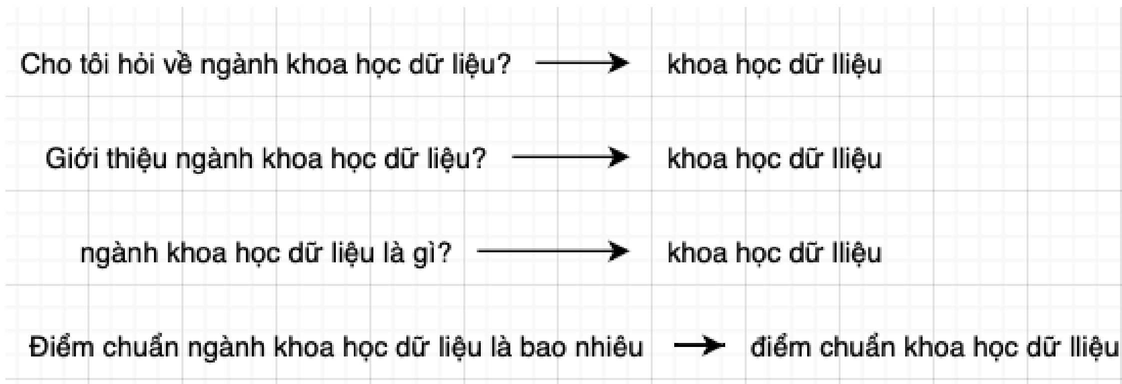

Figure 3. 4 Examples of removing stop-words

Just removing stop-words is not enough, in Vietnamese, word segmentation is an essential step for many NLP tasks, as Vietnamese is a tonal language with no explicit word delimiters. There are several approaches to segmenting Vietnamese text, including rule-based methods, statistical methods, and machine learning-based methods. For example, in English, only a word can describe an object, but Vietnamese needs 2 or 3 words to describe that object. More specifically, the English word "fan" has only one word, but the Vietnamese word "máy quạt" has two words. If we do not connect the two words "máy" and "quạt" together, there will be two words with two different meanings. To do this, we need the Underthesea - Word Segment process [13].

Why we use Underthesea for Word Segment in Vietnamese? Underthesea is a Vietnamese natural language processing (NLP) library that has several advantages compared to other NLP libraries for Vietnamese. Some of these advantages include:

- Accuracy: Underthesea is known for its high accuracy in analyzing Vietnamese text. It has been trained on a large corpus of Vietnamese text and uses state-of-the-art machine learning algorithms to improve its accuracy.
- Speed: Underthesea is designed to be fast and efficient, making it ideal for real-time applications. It can analyze large volumes of text quickly, which makes it a popular choice for businesses and organizations that need to process large amounts of data.

- Extensibility: Underthesea is an open-source library, which means that developers can add their own modules and features to it. This makes it highly extensible and customizable to suit specific needs.
- Easy to use: Underthesea has a simple and intuitive API that is easy to use, even for developers who are new to NLP. It also comes with pre-trained models that can be easily integrated into applications.
- Vietnamese-specific features: Underthesea has been specifically designed for the Vietnamese language, which means that it has features that are tailored to the unique characteristics of Vietnamese text. This includes support for Vietnamese word segmentation, part-of-speech tagging, named entity recognition, and more.

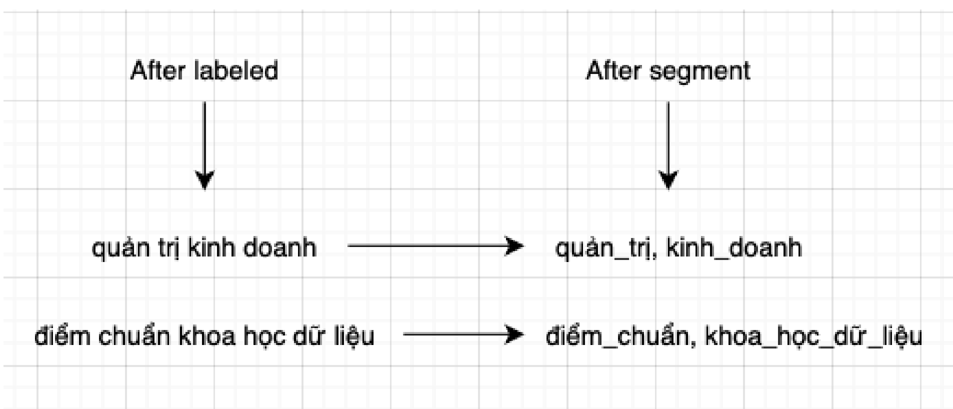

Figure 3. 5 Examples of Word Segment

After labeling, the dataset will be change to dictionary type (json file). That mean the “label” columns will become the “tag”, the “question” columns become the “patterns”, and the “answer” columns become the “responses”.

```
  tag: "['thực phẩm']"
▾ patterns: [] 1 item
    0: "Cho tôi hỏi về Công nghệ Thực phẩm"
▾ responses: [] 1 item
    0: "Ngành Công nghệ Thực phẩm (Mã ngành: 7540101) Chương trình: Trong nước Chương trình đào tạo ngành CNTP tại Trường
      Đai học Quốc tế được thiết kế nhằm đào tạo các kỹ sư CNTP, có những nét chính như sau:, Cung cấp kiến thức rộng, cân
      bằng về các khoa học thực phẩm và kỹ thuật thực phẩm., Cung cấp kiến thức cơ bản về quản lý thực phẩm (bao gồm cả quả
      n lý sản xuất, chất lượng, vệ sinh an toàn thực phẩm và vệ sinh môi trường)., Chú trọng đến kỹ năng thực hành và khả
      năng tiếp cận với thiết bị công nghệ cao., Giảng dạy hoàn toàn bằng tiếng Anh (kỹ sư sau khi tốt nghiệp sẽ đạt trình
      độ tiếng Anh TOEF pBT 550 hoặc tương đương (IELTS 6.0, TOEFL iBT 60))."
```


Figure 3. 6: Dataset as dictionary (json file)

3.5 User Input Tokenization

The process of normalizing user input is the process of dropping stopwords and word segment.

```
function tokenize_input(user_input):
   // Convert input to lowercase
   set text to the lowercase version of user_input

   // Replace input stop words with their corresponding values
   for each key j in the list of input_stopword keys:
      set text to the result of replacing all instances of j in text with the corresponding value in input_stopword[j]

   // Tokenize the text and return the result
   set tokens to the result of calling the word_tokenize function on text
   return tokens
```

Figure 3. 7 Tokenization function

Tokenization is a process of breaking down a text into smaller units called tokens, which are usually words or sub-words. Tokenization is an essential step in Natural Language Processing (NLP) because it enables the computer to understand and analyze text data.

Here are some benefits of tokenization of user input in NLP:

- Simplifies text processing: Tokenization simplifies text processing for the computer by breaking down the text into individual tokens. This makes it easier for the computer to analyze and understand the text.
- Enables text analysis: Tokenization enables text analysis by allowing the computer to count the frequency of individual words in a text. This is useful for tasks such as sentiment analysis, topic modeling, and text classification.

- Reduces data complexity: Tokenization reduces data complexity by breaking down the text into smaller units. This makes it easier to process large amounts of text data.
- Improves accuracy: Tokenization improves the accuracy of NLP models by reducing the noise in the text data. By breaking down the text into smaller units, tokenization removes unnecessary punctuation, capitalization, and other noise that can interfere with the analysis of the text.

3.6 Deep Neural Network-based

Deep neural networks can be used for question answering tasks through a process called "reading comprehension". These models are typically designed as a combination of two neural networks: a "contextual encoder" and an "answer pointer" [14].

The contextual encoder is responsible for reading and understanding the text that contains the answer to the question. It can be a pre-trained language model like BERT or GPT, which is trained on a large corpus of text to predict the next word in a sentence given its previous context. The answer pointer, on the other hand, is responsible for selecting the location of the answer in the text that was read by the contextual encoder. It is typically a recurrent neural network (RNN) or a transformer-based network that learns to predict the start and end positions of the answer in the text.

During inference, the question and the context are fed into the contextual encoder, which produces a sequence of contextualized embeddings for each token in the text. These embeddings are then passed to the answer pointer, which predicts the start and end positions of the answer in the text.

Finally, the answer is extracted by selecting the span of text between the predicted start and end positions. The selected text is then returned as the answer to the original question. The following will be the detailed explanation for the Bidirectional.

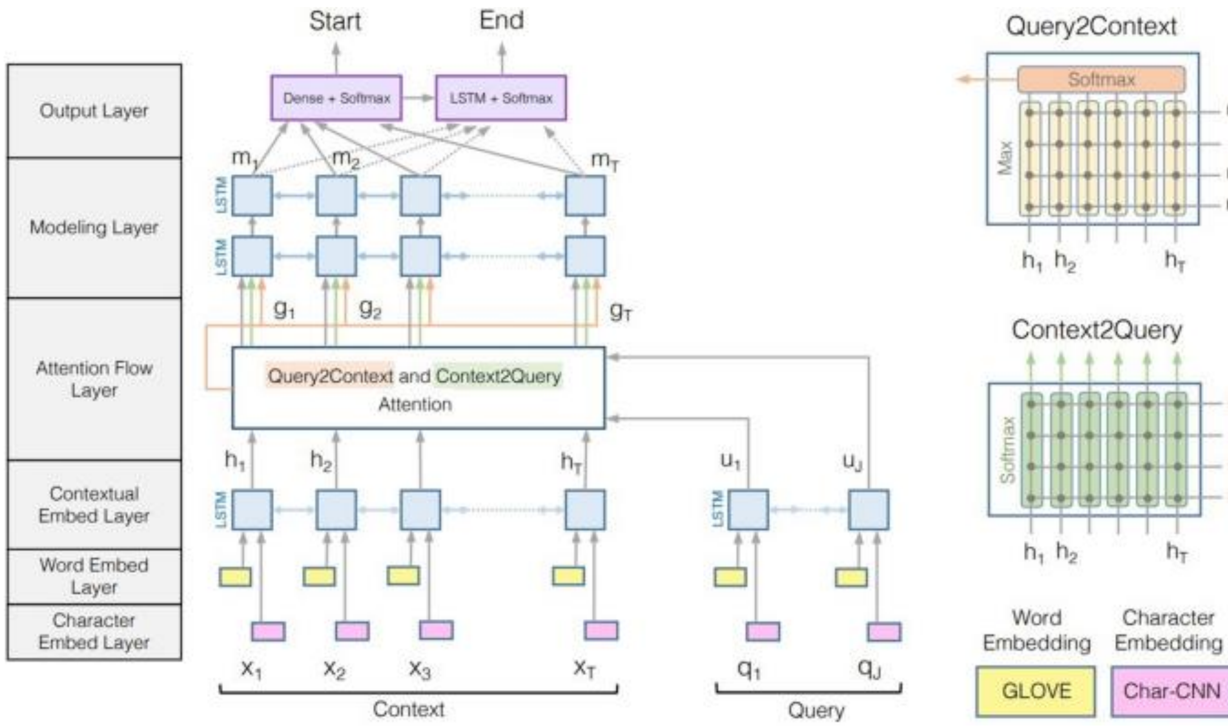

Figure 3. 8 Bidirectional Attention Flow [15]

The Question Answering system model for admissions advising is similar to the question answering system for an intelligent humanoid robot [14] but Embedding Layer is replaced by the processes of data pre-processing, labeling, and simplifying. The following are the attention weights for each word in the context:

Compute the similarity matrix S between the context and question embeddings:

$$S = C * Q^T$$

Where *C* is the contextual embedding matrix and *Q* is the question embedding matrix.

Compute the context-to-question (C2Q) attention weights a:

$$a = softmax(S)$$

Where ‘softmax’ is the softmax function, which normalizes the similarity scores to ensure that they sum up to 1, and S is the similarity matrix.

Compute the question-to-context (Q2C) attention weights b:

$$b = softmax(max(S, axis = 1))$$

Where max is the maximum function, which takes the maximum similarity score over each row of the similarity matrix S.

Compute the attended context representation c:

$$c = a * Q$$

Where * represents element-wise multiplication, a is context-to-question (C2Q) attention, and Q is the question embedding matrix.

Compute the contextual embedding with attention M:

$$M = [C, c, C * c, C * b]$$

Where [*C*, *c*, *C* * *c*, *C*] represents concatenation along the last axis, *C* is the contextual embedding matrix, and *c* is the attended context.

Finally, the output layer uses a fully connected neural network to generate the final answer:

$$y = W * M + b$$

Where *W* and *b* are the weight and bias parameters of the output layer, respectively.

Simplifying is a process of reducing the dimensionality of input data and improves model efficiency by grouping similar words together. The Simplifying process includes the following steps: Converting words to lowercases, removing any leading or trailing whitespace.

One hot encoding can be a useful preprocessing step when dealing with categorical variables as inputs. Many machine learning algorithms, including deep neural networks, require inputs to be in a numeric format, and one hot encoding is an effective way to convert categorical data into a format that can be easily understood by the model. For example, if we have a dataset with a categorical variable "major" that can take on three values: "hóa sinh", "ngôn ngữ anh", and "công nghệ thông tin", we can represent each value as a binary vector as follows:

- " hóa sinh": [1, 0, 0]
- " ngôn ngữ anh": [0, 1, 0]
- " công nghệ thông tin": [0, 0, 1]

This way, the categorical variable "major" is transformed into three numerical features that can be fed into the deep neural network.

However, one hot encoding can lead to high dimensionality when dealing with many categories, which may increase the computational complexity of the model. Additionally, one hot encoding may not be the best approach for ordinal data, where the order of the categories matters. In such cases, alternative encoding methods like label encoding or ordinal encoding may be more appropriate.

Figures. 3.8 and 3.9 show a few of the chatbot's web interface pages. The DNN-based chatbot responds to user queries that are entered into the chat box. Two consecutive questions, for instance, are answered correctly in Figures 3.8 and 3.9.

Figure 3. 9 Web interface of the chatbot

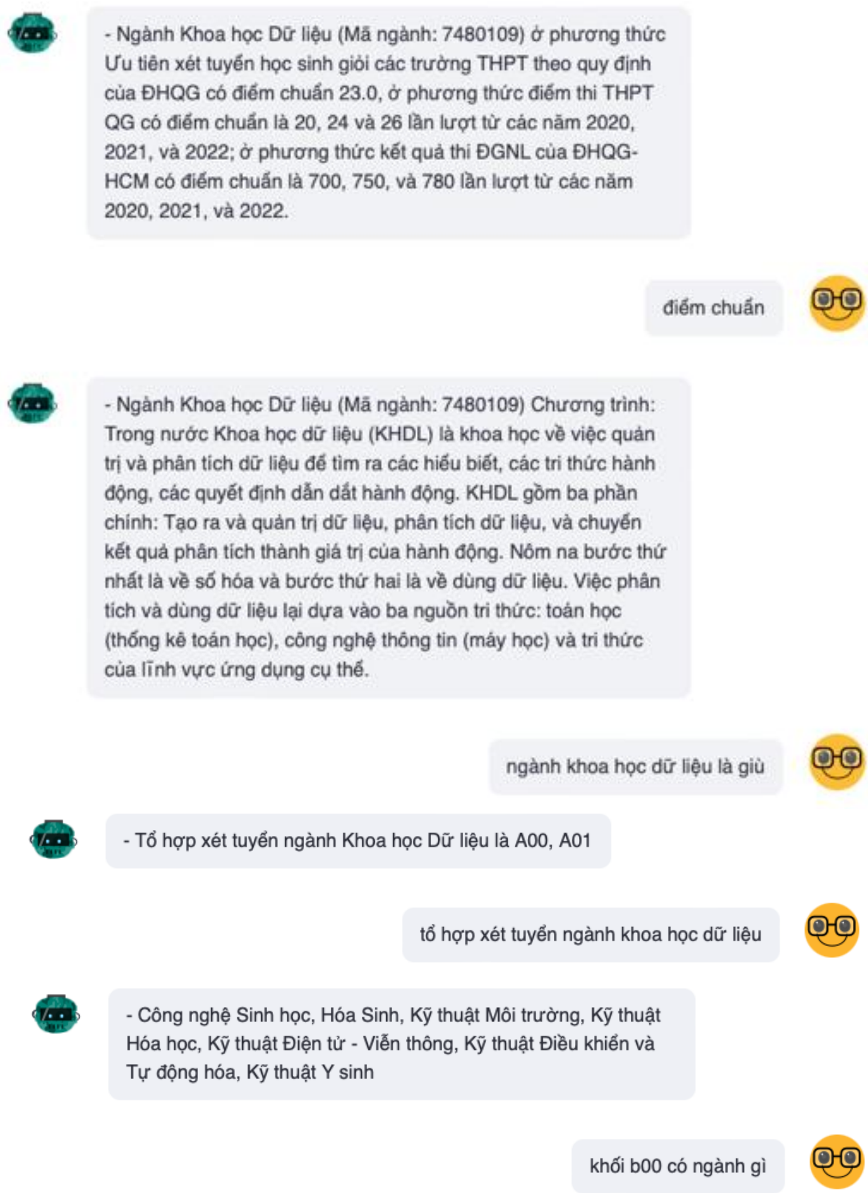

Figure 3. 10 Web interface of chatbot (cont.)

The following presents experiments in chapter 4 run on a MacBook with Apple M2 CPU, 8-Core GPU, 16-Core Neural Engine and 16GB RAM with hyperparameters below.

Here are the hyperparameters:

- n_epoch: The number of epochs (passes through the entire training set) to train the neural network. In the above code, n_epoch is set to 400.

- batch_size: The batch size used for training the neural network. In the above code, batch_size is set to 8.
- show_metric: Whether or not to display training metrics such as accuracy during training. In the above code, show_metric is set to True.
- activation: The softmax activation function is used in the last fully connected layer to output a probability distribution over the classes, where each class represents a different category that the neural network is trained to classify. The softmax activation function is a popular choice for the output layer in classification problems, as it ensures that the predicted probabilities are normalized and sum up to one, making them easy to interpret as class probabilities. In the above code, softmax activation function is used.

In addition to these hyperparameters, there are several other hyperparameters that can be set for training deep neural networks in general, such as:

- Learning rate: The learning rate determines how much the weights are updated during backpropagation. This can be set by passing a learning_rate argument to the tflearn.regression() function, it will be equal to 0.001 by default.
- Number of hidden layers: The number of hidden layers in the neural network can be adjusted by adding or removing tflearn.fully_connected() layers.
- Number of nodes per hidden layer: The number of nodes (neurons) in each hidden layer can be set by passing a n_units argument to the tflearn.fully_connected() function. In the above code, each hidden layer has 10 nodes.
- Dropout rate: Dropout is a regularization technique used to prevent overfitting. The dropout rate can be set by passing a dropout argument to the tflearn.fully_connected() function. Dropout is a regularization technique that can be used to prevent overfitting in neural networks. It works by randomly setting some of the neuron outputs to zero during

training, which can help to prevent the network from relying too heavily on any one neuron.

- Regularization strength: Regularization is used to prevent overfitting by adding a penalty term to the loss function. The strength of the regularization can be set by passing a regularizer argument to the tflearn.fully_connected() function.

Table 2: Hyper-Parameters

N_epoch	Batch_size	Show_metric	Activation	Learning Rate	Number of hidden layers	Number of nodes per hidden layer
400	8	True	softmax	0.001	3	Layer 1^{st} :512
						Layer 2^{nd} : 256
						Layer 3^{rd} : 128

3.7 Voice Input

3.7.1 Getting microphones index

It first creates a PyAudio object and then uses the get_host_api_info_by_index method to retrieve information about the default host API (MacOS install PortAudio instead of PyAudio).

The numdevices variable is then set to the number of available input devices on the default host API. The code then iterates through each device and checks if it has any input channels.

If a device has input channels, it prints the device ID and name to the console. The device ID can be used as the microphone index when recording audio using PyAudio.

```
// Create a new PyAudio object
set p to a new instance of the PyAudio class

// Get information about the host API
set info to the result of calling p's get_host_api_info_by_index method with parameter 0

// Get the number of audio devices
set numdevices to the value of the 'deviceCount' key in info

// Iterate over all audio devices
for each i in the range from 0 to numdevices:
  // Check if the device has input channels
  if the value of 'maxInputChannels' key in the device info for device i is greater than 0:
    // Print information about the device
    print "Input Device id ", i, " - ", the value of the 'name' key in the device info for device i
```

Figure 3. 11 Get device index

3.7.2 Implementing Google Speech Recognition

The Google Speech Recognition service also uses **a deep neural network** to analyze the audio input and convert it into text. The neural network is trained on large datasets of spoken language, allowing it to recognize patterns in speech and accurately transcribe what is being said.

The process of speech recognition involves several steps. First, the audio input is broken down into small segments, usually lasting a fraction of a second each. Each segment is then analyzed to determine the most likely phonemes (individual speech sounds) that make up the spoken word. These phonemes are then combined to form words, and the words are combined to form sentences.

Google Speech Recognition also uses a language model to help interpret the transcribed text. The language model considers the context of the spoken words, as well as the grammar and syntax of the language being used. This helps to improve the accuracy of the transcription by reducing the number of possible interpretations of the spoken words.

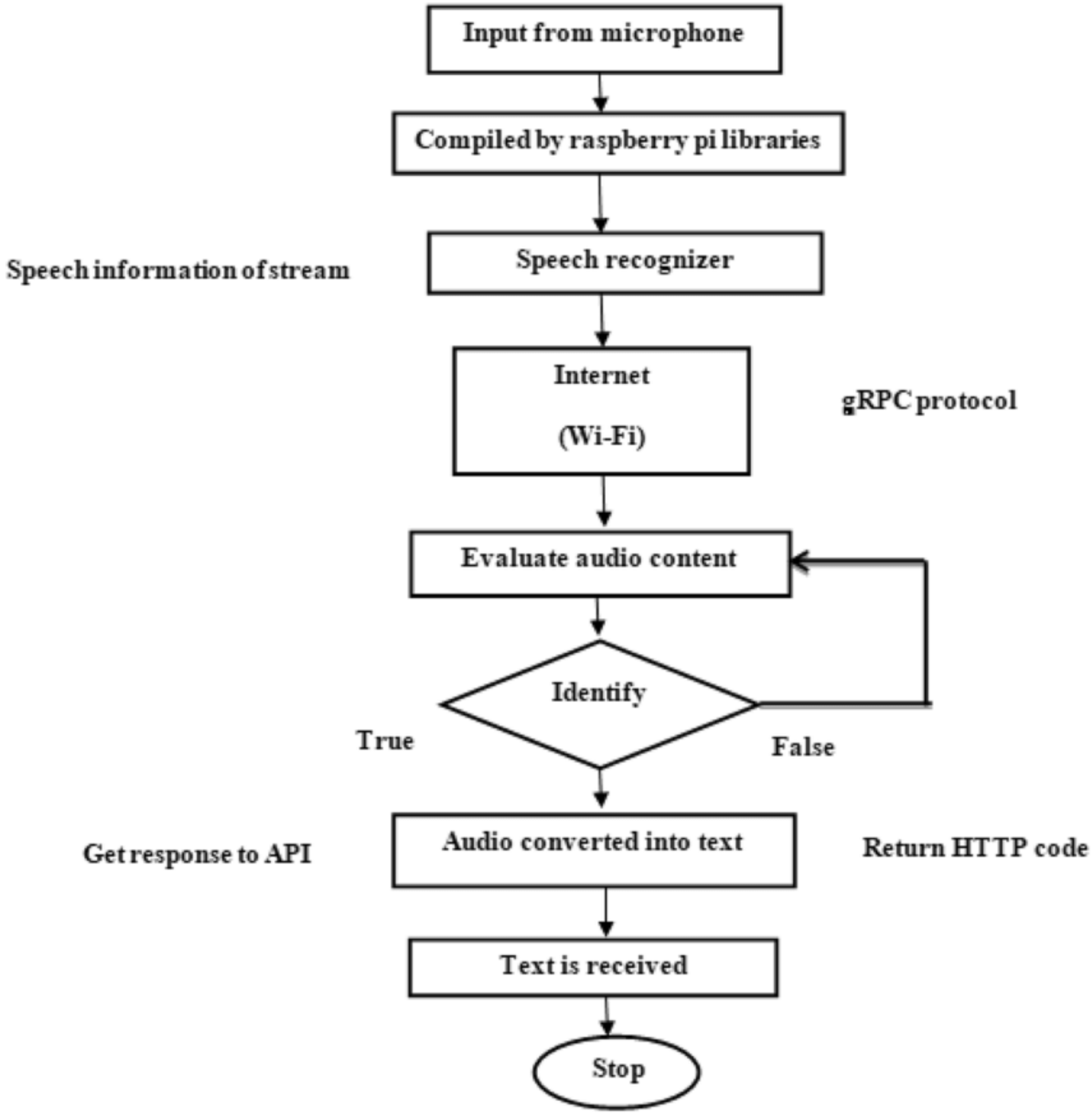

Figure 3. 12 Flow chart of Google Speech Recognition converting speech to text

```
function transcribe_audio(audio_data, format):
   // Convert audio data to WAV format
   set format to 'audio/wav'
   audio_data = st.audio(audio_data, format)

   // Save audio data to file
   create a new BytesIO object called audio_file
   write audio_data to audio_file
   create a new file called "recorded_audio.wav"
   write the contents of audio_file to "recorded_audio.wav"

   // Transcribe audio to text
   create a new Recognizer object called r
   open the "recorded_audio.wav" file with an AudioFile object called source
   use r's record method to read audio from source
   set language to 'vi-VI' and show_all to True
   use r's recognize_google method to convert audio to text
   extract transcript from the first alternative returned
   convert transcript to lowercase

   return the lowercase transcript
```

Figure 3. 13 The pseudo code about implement speech recognition

This code first initializes the Recognizer object from the SpeechRecognition library, which is used to recognize speech. It then sets up the microphone as the audio source using the Microphone class.

The listen() method is used to record the audio input from the microphone and store it in the audio variable. The recognize_google() method is then used to transcribe the audio input into text using the Google Speech Recognition API.

Finally, the transcribed text is displayed to the user. If the API is unable to transcribe the audio input, the code will catch the UnknownValueError or RequestError exceptions and display an error message (Figure 3.13).

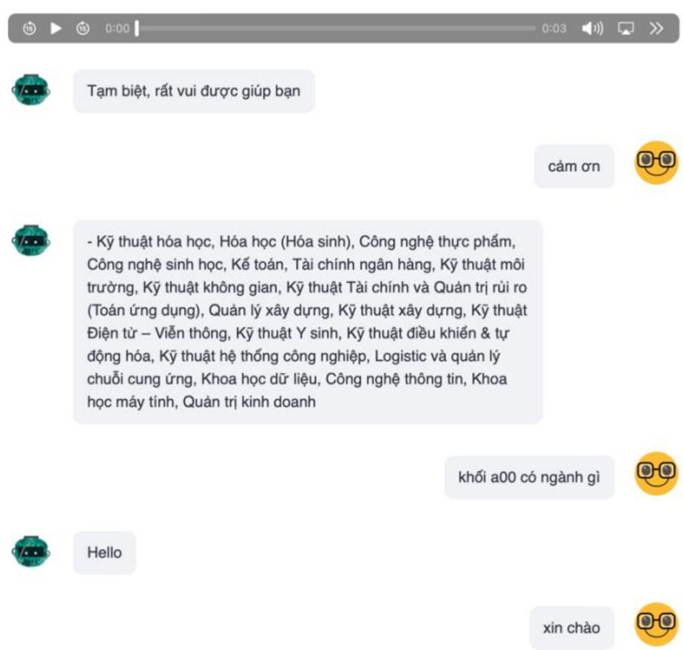

Figure 3. 14 Example of how speech recognition work in chatbot

3.8 Context handling

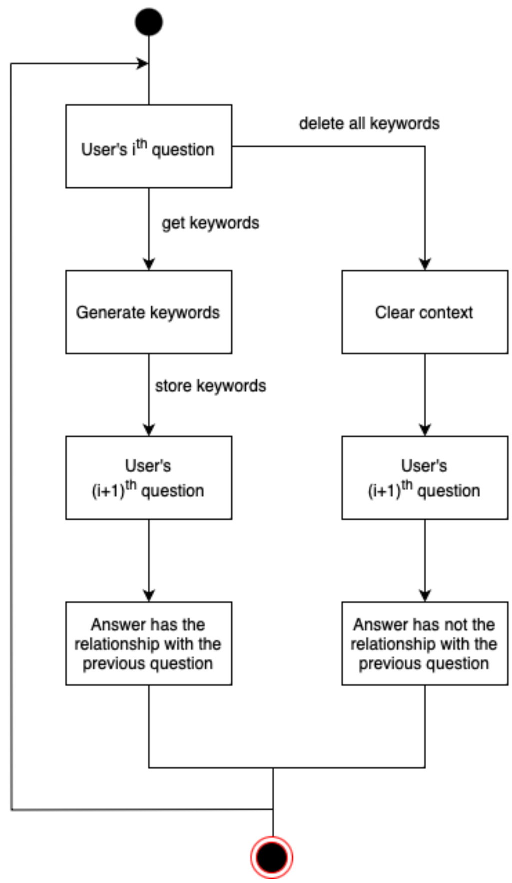

Figure 3. 15: Context handling workflow

The context handling workflow starts from finding keywords (disciplines, subjects, ...) based on the user's previous questions and matching it with the next question. For example, a user's first question is "Giới thiệu cho tôi ngành khoa học dữ liệu?" then the keyword would be "khoa học dữ liệu". It will then be paired with the 2nd question "Điểm chuẩn bao nhiêu?", the 2nd question will become "Điểm chuẩn khoa học dữ liệu bao nhiêu?". This saves users from having to enter too many redundant characters. By using keywords from previous questions, the chatbot can provide a more personalized and relevant conversation experience for the user. It also helps the chatbot to understand the user's intent and provide more accurate responses.

If a user clicks the "Clear Context" button, it means that they want to start a new conversation without any previous context. In this case, all the keywords and information that the chatbot has gathered from the previous conversation will be deleted.

This can be useful in situations where the user wants to change the topic of the conversation or start a new task. By clearing the context, the chatbot can start fresh and avoid any confusion or misunderstandings based on the previous context.

For example; in Figure 3.16, in questions 2 and 3 instead of having to ask "điểm chuẩn khoa học dữ liệu" and “tổ hợp xét tuyển khoa học dữ liệu”, the user can simply ask "điểm chuẩn" and “tổ hợp xét tuyển”, because we are in the context of industry "khoa học dữ liệu"

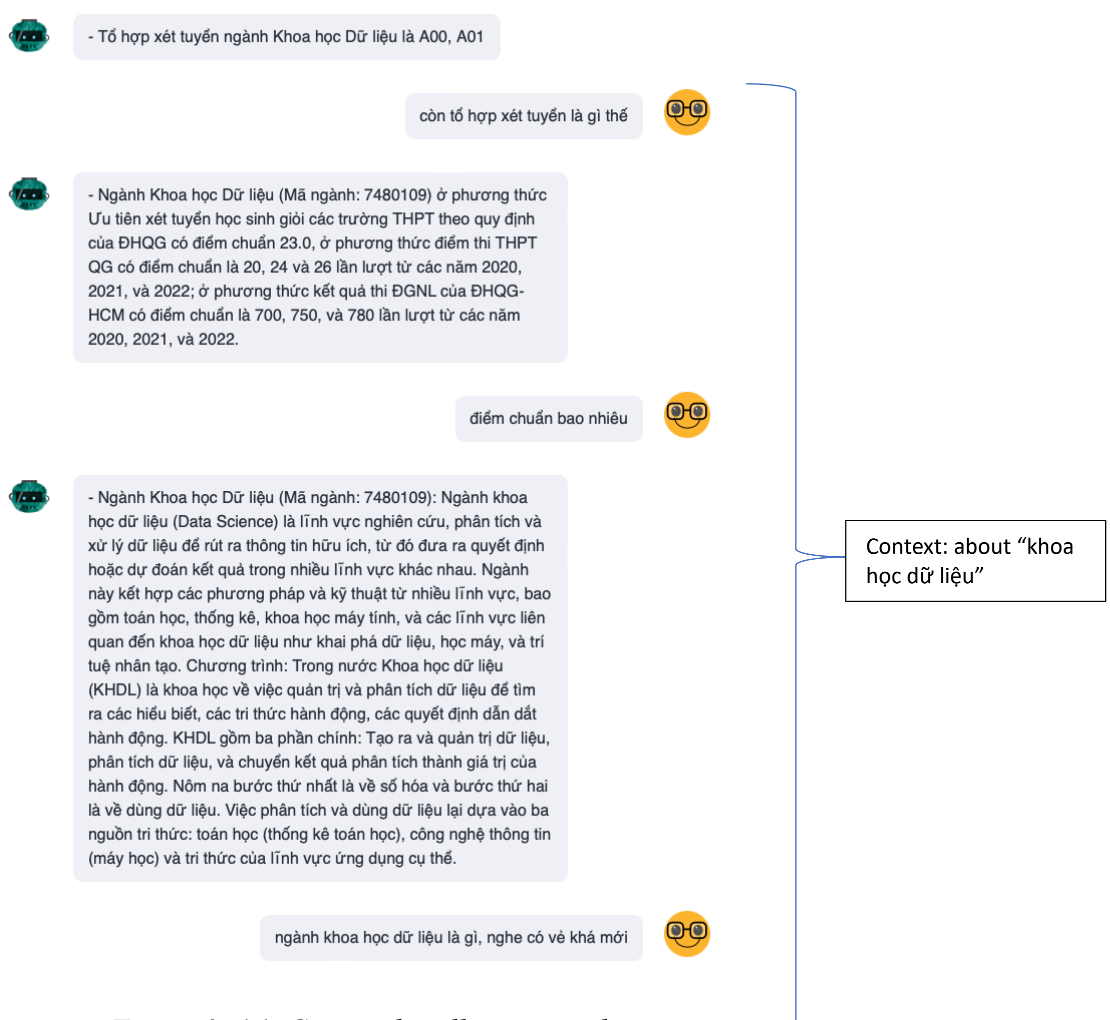

Figure 3. 16: Context handling example

Chapter 4 Experimental Results and Evaluation

Evaluation methods for DNN-based Question Answering System: (1) loss assessment, (2) user-based assessment (following the instructions in this paper [9]) We do a survey for friends to evaluate, (3) evaluate using case studies. In addition, DNN-based QAS is compared with the ChatGPT-3.5 model based on several case studies.

4.1 Experiment 1

In this experiment, in addition to evaluating the accuracy of the model, training DNN-based QAS without and using language processing techniques in Vietnamese will be compared. This is to assess the importance of Vietnamese data processing in natural language processing mentioned in Section 3.3.2.

4.1.1 Evaluation methods

Loss: The loss is a gauge of how effectively the model is accomplishing its goal. It is described as a mathematical function that calculates the discrepancy between the output that the model predicts and the output that actually occurs. Depending on the kind of issue being solved, several loss functions might be employed. For instance, binary cross-entropy and categorical cross-entropy are often used loss functions in classification issues. The difference between the model's anticipated output and the actual output, averaged across all samples in the dataset, is measured by the loss, which is a scalar quantity. The mean squared error (MSE) loss, which is defined as the most used loss function in TensorFlow is:

$$\boldsymbol{loss = \frac{1}{N} * sum_i = 1^N \left(y_i - y_{pred_i}\right)^2}$$

Where N is the number of samples in the dataset, y_i is the actual output for the i-th sample, and y_{predi} is the predicted output for the i-th sample.

In binary classification problems, the binary cross-entropy loss is commonly used instead of the MSE loss. In multi-class classification problems, the categorical cross-entropy loss is commonly used.

Accuracy: The accuracy is a metric that measures how well the model's predictions match the ground truth labels. It is defined as the percentage of correct predictions made by the model on a given dataset. For example, an accuracy of 0.8 means that the model correctly predicted 80% of the labels in the dataset. The accuracy is a scalar value that measures the percentage of correctly classified samples in the dataset. It is defined as:

$$\boldsymbol{accuracy} = \frac{\boldsymbol{X}}{\boldsymbol{N}}$$

Where:

N: total number of samples

X: number of correctly classified samples

The formula of the `evaluate()` method in TensorFlow for Figure 4.1 and Table 2 can be expressed as follows:

$$metrics = model.evaluate(x = test\_data, y = test\_labels, batch\_size = batch\_size, verbose = 1, sample\_weight = None, steps = None)$$

Where:

- `model` is the trained deep learning model
- `test_data` is the input test data (a NumPy array or a TensorFlow `Dataset`)
- `test_labels` is the corresponding test labels (a NumPy array or a TensorFlow `Dataset`)
- `batch_size` is the number of samples per batch (an integer)
- `verbose` is the verbosity mode (0 = silent, 1 = progress bar)

- \`sample_weight\` is an optional weight array to weight the contribution of different samples to the loss and metrics

- \`steps\` is the number of steps to run the evaluation for (an integer or \`None\`)

The \`evaluate()\` method returns a list of metric values, where the order of the metrics corresponds to the order of the metrics specified during training. For example, if during training you specified \`metrics=['accuracy', 'loss']\`, then the \`evaluate()\` method will return a list of two values, where the first value is the accuracy and the second value is the loss.

4.1.2 Original Deep Neural Network (Without processing data)

In this experiment, the DNN-based QAS will be trained with 645 samples of unsegmented and unlabeled questions and answers (samples without steps in sections 3.3.2 and 3.4). It is trained with 400 epochs, 32400 training steps and batch size of 8 (Hyper-Parameter in Table 2). The result is that during training, the total loss is 0.79246 and the accuracy is 0.7568 (Figure 4.1).

```
Training Step: 32399  | total loss: 0.81732 | time: 0.084s
| Adam | epoch: 400 | loss: 0.81732 - acc: 0.7575 -- iter: 640/645
Training Step: 32400  | total loss: 0.79246 | time: 0.085s
| Adam | epoch: 400 | loss: 0.79246 - acc: 0.7568 -- iter: 645/645
--
```

Figure 4. 1: Loss assessment metric without processing data

4.1.3 Deep Neural Network after processing data

In this test, the model is trained with 400 epochs, with a training step of 32400, same as section 4.1.2 (Hyper-Parameter in Table 2). The total loss during training is 0.66553, the time is about 0.095s for one epoch. The accuracy we get is about 83% more with 645 samples (Figure 4.2). In the other hand, when the dataset separated as in Table 1, the result seems better without division.

As Table 3 shows, the accuracy of each model is also higher. Putting data split models into practice is also easy, just put them in the IF ELSE.

```
Training Step: 32399  | total loss: 0.72656 | time: 0.094s
| Adam | epoch: 400 | loss: 0.72656 - acc: 0.8155 -- iter: 640/645
Training Step: 32400  | total loss: 0.66553 | time: 0.095s
| Adam | epoch: 400 | loss: 0.66553 - acc: 0.8340 -- iter: 645/645
--
```

Figure 4. 2 Loss assessment metric without split data

Table 3: Loss assessment with original data

No.	Question types	Samples	Epoch	Loss	Accuracy
1	All questions	645	400	0.66553	0.8345

Table 4: Loss assessment when split data

No.	Question types	Samples	Epoch	Loss	Accuracy
1	Introductory question about a certain field trained at the university.	120	400	0.127345	0.8523
2	Questions about the main function of a certain department or department at the university.	50	400	0.278681	0.9124
3	Questions about the standards and admission criteria of each major trained at university.	100	400	0.091213	0.8812
4	Questions about general information such as tuition fees, training majors,…	110	400	0.123445	0.8731
5	The question about what subjects you like to study or what activities you like to do, which major should you choose?	250	400	0.388901	0.8481
6	Questions about the entrance exam and some extra questions	15	400	0.002345	0.9921

4.1.4 Remark

From the above assessment, it is easy to see that the accuracy of the model without data processing (75.65%) is lower than that of the model with data processing but not dividing the data by question type (83.40%), and significantly lower than the model with clear data processing and question classification (Table 3). If there are more hardware resources, if you increase the number of nodes of each layer, it can be improved even more.

Thus, we can see the input data processing (Vietnamese word segmentation, question classification, answer labeling) especially for Vietnamese in the process of training chatbot model based on Deep Neural Network is extremely important. Accuracy can be improved up to more than 7%. Therefore, the process of processing input data in Vietnamese is extremely important and indispensable if you want to have a system that automatically answers questions based on multilayer processing technique (Table 4).

Table 5: Comparison of 3 DNN-based QAS in experiment 1

	DNN-based QAS without processing data (Original DNN)	DNN-based QAS with processing data but without splitting data depend on question types	DNN-based QAS with processing data and splitting data depend on question types (Table 3)
Loss	0.79246	0.66553	Lowest - 0.002345 Highest - 0.388901
Accuracy	0.7568	0.8340	Lowest - 0.8481 Highest - 0.9921

4.2 Experiment 2

According to [9] and [16] states that the rates of satisfaction, correctness, and usefulness, which are represented by scores ranging from 0 to 1, may be used as the evaluation criteria for responses. The Cambridge Dictionary provides the following definitions for these metrics:

- Satisfactory: good or good enough for a particular need or purpose.
- Correctness: the quality of agreeing with the true facts or with what is generally accepted.
- Usefulness: effective; helping you to do or achieve something.

Domain specialists in charge of creating the International University's website for admissions counseling comment on the chatbot's responses. The highest score (1) indicates that the answer is entirely satisfactory, accurate, or beneficial. The medium score (0.5) indicates that the response might provide some reasonable, significant, or helpful information. The response is utterly irrelevant or useless if it receives a score of zero.

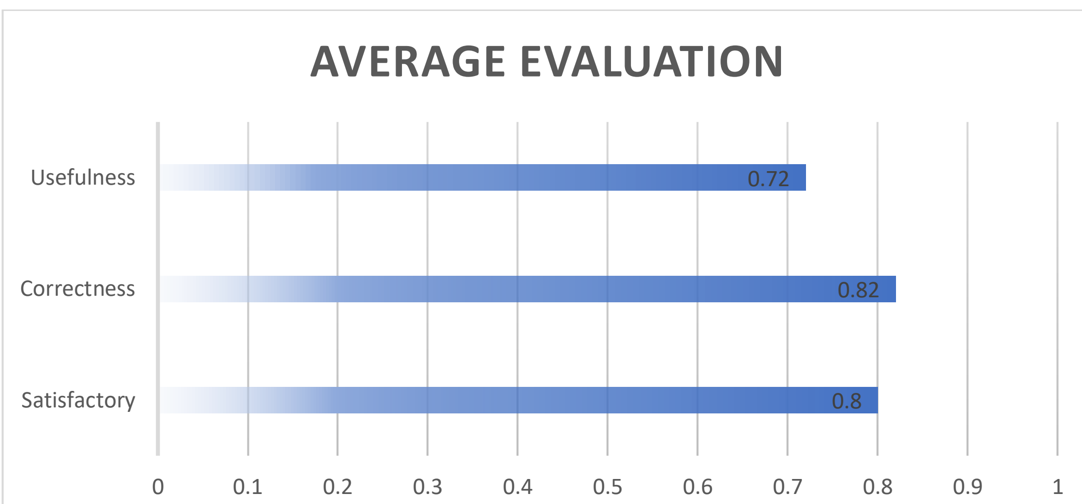

Figure 4. 3: Success rates evaluation metrics of the proposed model in experiment 2

This score is based on the assessment of peers through 40 questions along with the Bot's answers in Table 1. Of those 40 questions (Table 6), about 72% of colleagues think it's Usefulness (meaning 30 questions out of 40 useful questions), 82% think it's Correctness (that is, in the Bot's answers there are still incorrect answers, this coincides with the data in Experiment 1), and 80% said that Satisfactory.

In [9], The satisfaction, accuracy, and usefulness rates are all over 74%, which is considered good. The success rates for usefulness and satisfaction are greater than 70%. If you look at the results in experiment 2 and results in [9], DNN-based QAS has progressed (usefulness 72%, correctness 82%, and satisfactory 80%).

Table 6: Success rates compare to Ontology-based QAS in Education

	Ontology-based QAS in [9]	DNN-based QAS
Satisfactory	77%	80%
Correctness	74%	82%
Usefulness	77%	72%

Table 7: Evaluation survey questions

No.	Questions
1	Phương thức xét tuyển của trường?
2	Giới thiệu trung tâm dịch vụ công nghệ thông tin?
3	Cho tôi hỏi học phí là bao nhiêu?
4	Ngành khoa học dữ liệu là gì
5	Điểm chuẩn ngành logistics
6	Cho tôi hỏi về phòng cơ sở vật chất
7	Chỉ tiêu ngành công nghệ thông tin
8	Điểm chuẩn Ngôn ngữ Anh
9	Khối A00 có môn gì?
10	Khối A01 có môn gì?
11	Khối D07 có môn gì?
12	Khối D07 có ngành gì?

13	Giới thiệu ngành khoa học máy tính
14	Tổ hợp xét tuyển ngành Khoa học dữ liệu?
15	Tổ hợp xét tuyển ngành quản trị kinh doanh?
16-40	[Refer to https://docs.google.com/forms/d/e/1FAIpQLSfxGual5A24ZrsuABjgC-kGZGI8g90xS2udb_kf_G8JdH-tDQ/viewform?usp=share_link

4.3 Experiment 3

In the third experimental part, evaluating the model's answers with practical questions, more specifically, there will be two cases where this model can support:

- We will play the role of a student who has just graduated from high school and wants to enter IU.
- We will act as a student studying at IU and want to find the necessary information.

Here are some questions (Table 7) from IU's admissions consultants, which we consider challenging for the model.

Table 8: Experimental questions at different difficulty levels

No.	Questions	Level
1	Tôi giỏi [một môn học ở cấp 3] thì nên chọn ngành gì?	2
2	Ngành [một ngành được đào tạo tại IU] là gì?	1
3	Điểm chuẩn và chỉ tiêu của ngành đó?	1
4	Khối [A00, A01, D07,…] có môn gì?	1
5	Khối [A00, A01, D07,…] có ngành gì?	1
6	Tổ hợp xét tuyển [của một ngành được đào tạo tại IU] là gì?	1
7	Tôi thích [ví dụ: lập trình, quảng cáo,…] thì chọn ngành gì?	3
8	Tôi thích/giỏi [một môn học hay một thứ gì đó] thì nên chọn ngành gì?	3
9	Chức năng của [một khoa hay phòng ban tại IU]?	1
10	Cho tôi hỏi về [một phòng ban hay trung tâm tại IU]?	1

11	Khoa [tại IU] đào tạo những ngành nào?	2
12	Cho tôi hỏi học phí trung bình bao nhiêu?	3

4.3.1 Experiment in the case of a student who wishes to enter IU

Often a student who has just graduated from high school will not know much about the majors in universities in general and IU in particular. They only know what they like or are good at in high school, but do not know what job they will do in the future. So, we will experiment in this case. Moreover, a new student will often be interested in tuition fees, quotas, or benchmarks for the major they want to study.

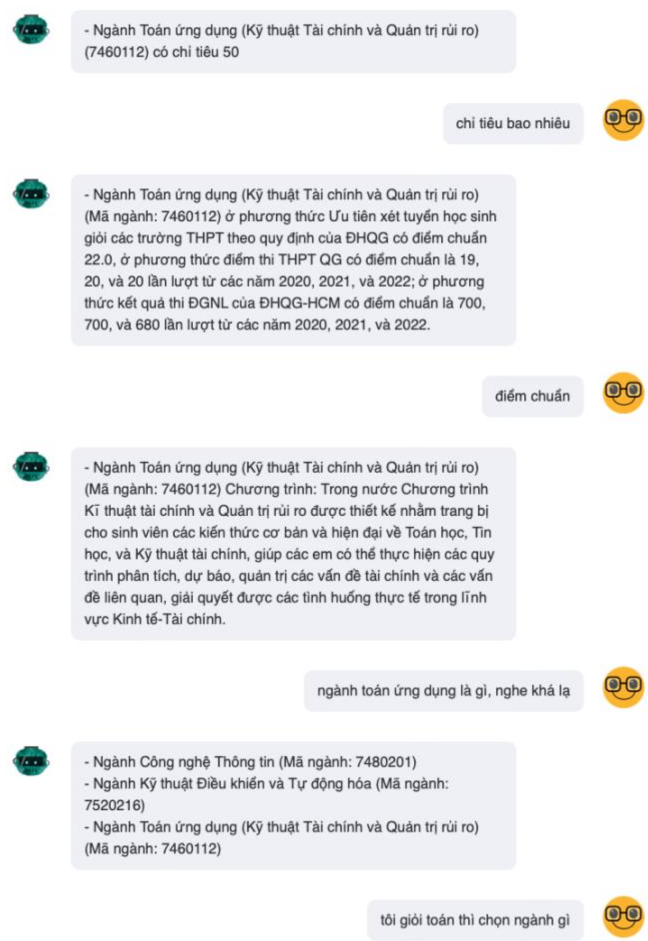

Figure 4. 4: 1st case that a student who wishes to enter IU

For example, in Figure 4.4 in this case, a student asked him what major he should study in math, and the model answered Information Technology, Automation, and Applied Mathematics. According to my assessment, and those of the admissions consultants at IU, this answer is completely reasonable.

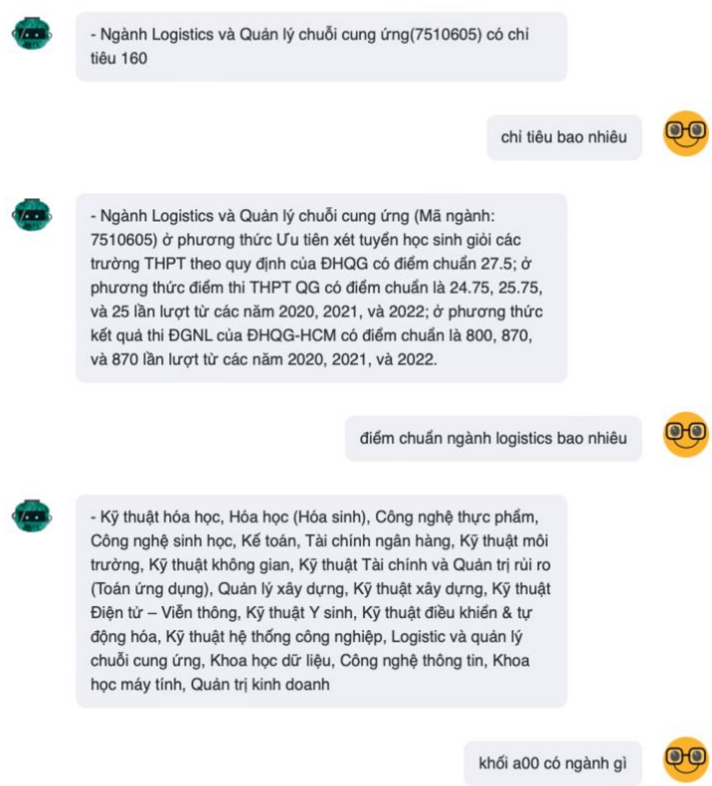

Figure 4. 5: 2nd case that a student who wishes to enter IU

For the next example in Figure 4.5, a student registers for the A01 exam, he wants to know what majors IU enrolls in the A01 class. Based on the block-based admissions table on IU's admissions website, the model's answer is completely correct.

- Ngành Công nghệ Thực phẩm có những môn 'hóa hữu cơ', 'phát triển và tiếp thị sản phẩm thực phẩm', 'khoa học môi trường', 'hóa vô cơ', 'thống kê trong sinh học', 'công nghệ chế biến các sản phẩm thịt', 'bao bì và phụ gia thực phẩm', 'phân tích cảm quan thực phẩm', 'các luật và tiêu chuẩn thực phẩm', 'công nghệ trà, cà phê, ca cao', 'hóa sinh', 'hóa hữu cơ ', 'dinh dưỡng và thực phẩm chức năng', 'công nghệ chế biến sữa và các sản phẩm từ sữa', 'độc tố và an toàn thực phẩm', 'phân tích thực phẩm', 'vi sinh thực phẩm', 'tư duy phân tích', 'công nghệ chế biến lương thực', 'các hệ thống đảm bảo chất lượng thực phẩm', 'quản lý dự án', 'công nghệ sản xuất đồ uống', 'cơ học ứng dụng', 'enzym và lên men thực phẩm', 'quá trình và thiết bị thực phẩm ', 'hóa học công nghiệp', 'nhập môn khoa học và công nghệ thực phẩm', 'phân tích vi sinh thực phẩm', 'giải tích ', 'các quá trình thiết bị công nghệ thực phẩm ', 'di truyền phân tử', 'các nguyên lý kỹ thuật thực phẩm', 'vật lý ', 'hóa lý ', 'thiết kế thí nghiệm', 'hóa học thực phẩm'.
- Ngành Kỹ thuật Hệ thống Công nghiệp có những môn 'quản lý chất lượng', 'kỹ thuật hệ thống', 'kỹ thuật dự báo', 'vận trù học - các mô hình ngẫu nhiên', 'thiết kế và phát triển sản phẩm', 'quản lý vật tư tồn kho', 'kỹ thuật ra quyết định đa mục tiêu', 'kỹ thuật thiết kế chuỗi cung ứng và logistics', 'kỹ năng lãnh đạo', 'thiết kế mặt bằng hệ thống công nghiệp', 'vận chuyển quốc tế', 'thiết kế thực nghiệm', 'vẽ kỹ thuật', 'hệ thống sản xuất tinh gọn', 'hệ thống thông tin quản lý', ': cadcamcnc', 'thương mại điện tử trong logistics và chuỗi cung ứng', 'tin học cho kỹ sư', 'quản lý sản xuất', 'cad cam', 'vận trù học - các mô hình tất định', 'đo lường lao động và thiết kế công việc', 'quản lý bán lẻ', 'cơ kỹ thuật – động học', 'mô hình hóa và mô phỏng', 'xác suất thống kê cho kỹ thuật', 'kỹ thuật điều độ trong sản xuất và dịch vụ'

ngành công nghiệp thực phẩm học môn gì

- Tổ hợp xét tuyển ngành Công nghệ Thực phẩm là A00, A01, B00, D07

tổ hợp xét tuyển là gì

- Ngành Công nghệ Thực phẩm (Mã ngành: 7540101) ở phương thức Ưu tiên xét tuyển học sinh giỏi các trường THPT theo quy định của ĐHQG có điểm chuẩn 22.0; ở phương thức điểm thi THPT QG có điểm chuẩn là 18.5, 20, và 20 lần lượt từ các năm 2020, 2021, và 2022; ở phương thức kết quả thi ĐGNL của ĐHQG-HCM có điểm chuẩn là 620, 680, và 650 lần lượt từ các năm 2020, 2021, và 2022.
- Ngành Công nghệ Thực phẩm chương trình liên kết ĐH Nottingham (Mã ngành: 7540101_UN) ở phương thức Ưu tiên xét tuyển học sinh giỏi các trường THPT theo quy định của ĐHQG có điểm chuẩn 18.0; ở phương thức dựa trên kết quả kỳ thi ĐGNL ĐHQG có điểm chuẩn là 600' ở phương thức kỳ thi THPT QG có điểm chuẩn là 18.0

điểm chuẩn bao nhiêu

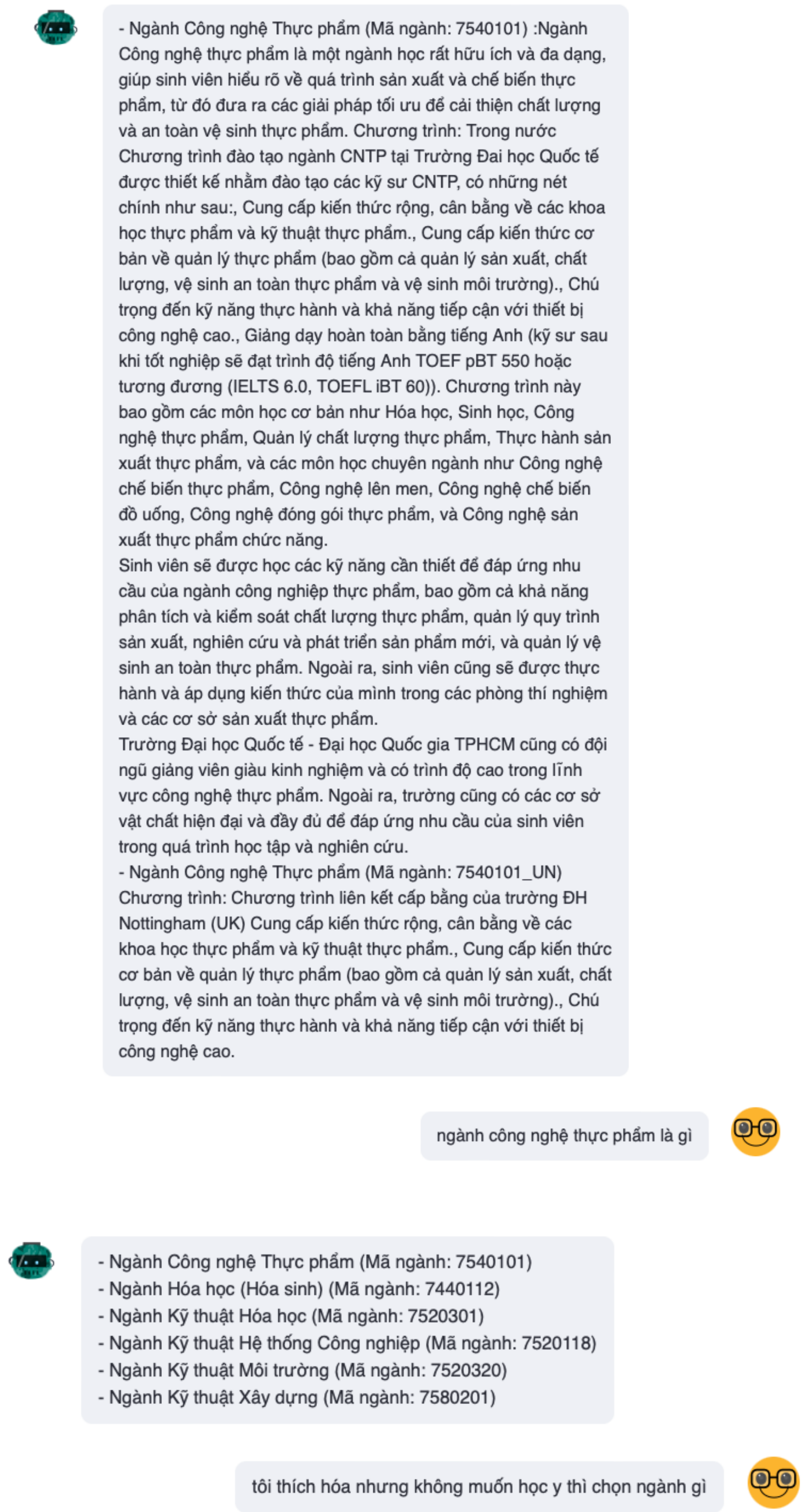

Figure 4. 6: 3rd case that a student who wishes to enter IU

In the example Figure 4.6, a student likes chemistry, but he does not want to study a major related to Medicine. So, the model's answer is perfectly reasonable because those majors need to be good at chemistry, but they are not related to Medicine. If the model's answer includes Biomedical Engineering (Kỹ thuật Y Sinh), it will not be correct.

From the above specific cases, we can judge that this model is completely capable of answering the questions of a new student who wishes to enter IU correctly.

4.3.2 Experiment in the case of a student studying at IU

According to staffs who are working in the training department and administration of the admissions consulting website, often students studying at IU will tend to ask for information about departments or faculties.

Figure 4. 7: Case of a student studying at IU

Based on the information on IU's information pages and the staff of the departments, the model's answer is completely correct. If linking the correlation between Experiment 1 and 3, the loss assessment in Experiment 1 has an accuracy of more than 83%, which is completely valid.

4.4 Compare with ChatGPT-3.5 by case studies

In this experimental part, we will compare with ChatGPT version 3.5 at 3 different question levels as shown in Table 8. To see the strengths and weaknesses of these two chatbot models in the field of consulting support enrollment inquiries in International University - VNU.

4.4.1 Level 1 questions

Level 1 will be general introduction questions about an industry, department, and benchmarks, criteria of a certain industry at IU.

In the first question, two models will be asked about the question of introducing basic information about data science at IU (Figure 4.8). From the answer in GPT-3.5, we can easily see the serious misinformation, which is understandable because it uses the word concatenation method. As for the DNN part, because it has been trained, this question is not too difficult for it. The ease with which this group of questions can be answered is predictable. However, GPT-3.5 is expected that at least be able to answer the definition of Data Science, however it does not answer this question either.

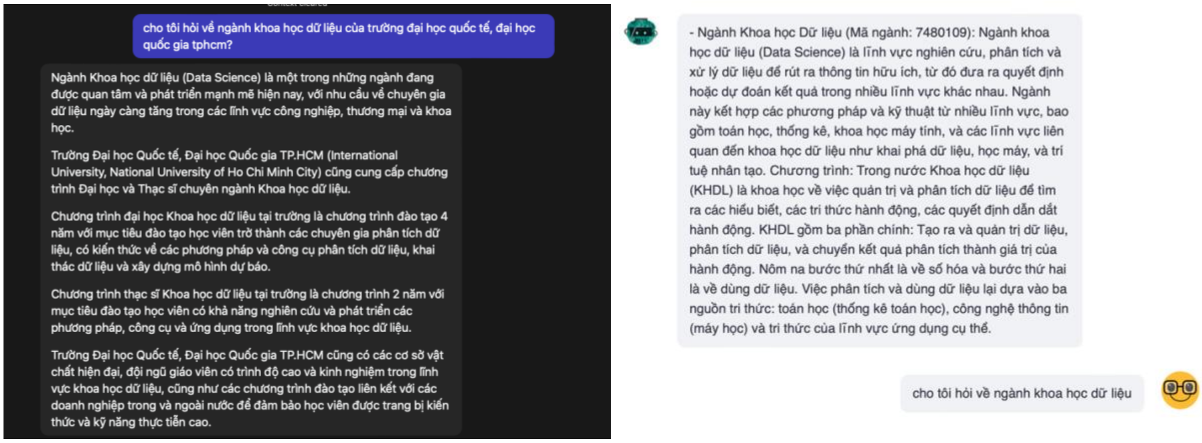

Figure 4. 8: GPT-3.5's answer (left) and DNN-based's answer (right) in general information about data science major

In this question (Figure 4.9), GPT-3.5 has not been studied so it cannot answer correctly, and its answer is also acceptable and very natural. As for DNN, just like the above question, answering level 1 questions is too simple.

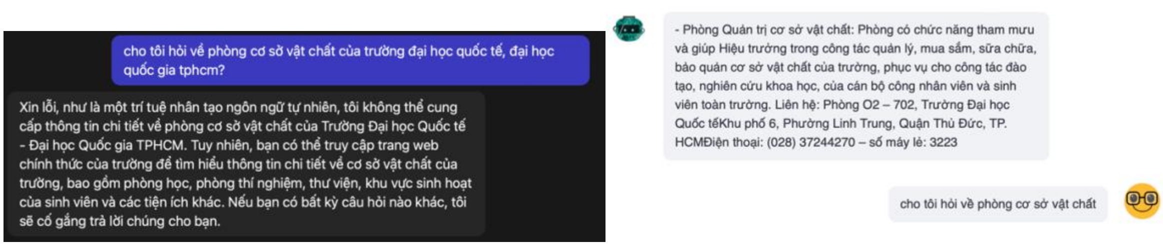

Figure 4. 9: GPT-3.5's answer (left) and DNN-based's answer (right) in information about facilities room

In the question in Figure 4.10 and 4.11, GPT-3.5 answered completely wrong about IU's training information.

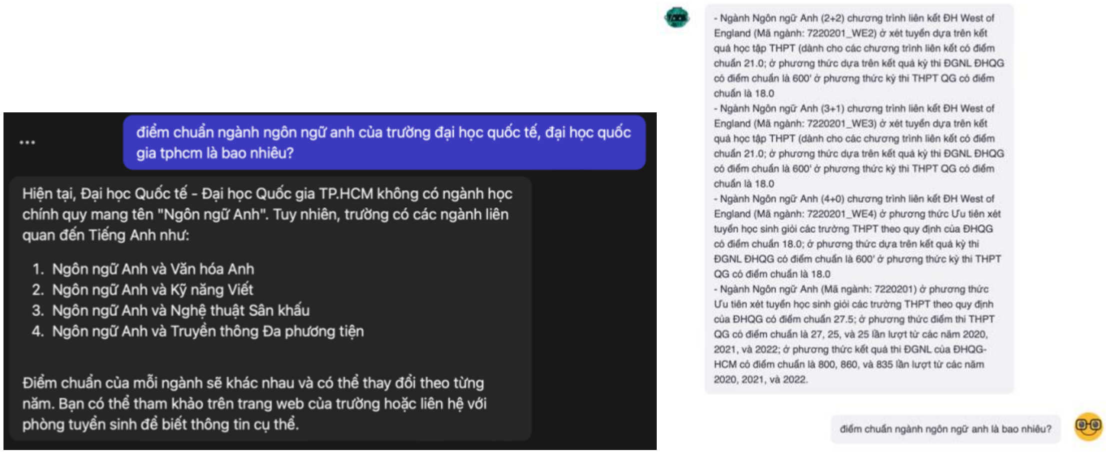

Figure 4. 10: GPT-3.5's answer (left) and DNN-based's answer (right) in Benchmark of English Language major

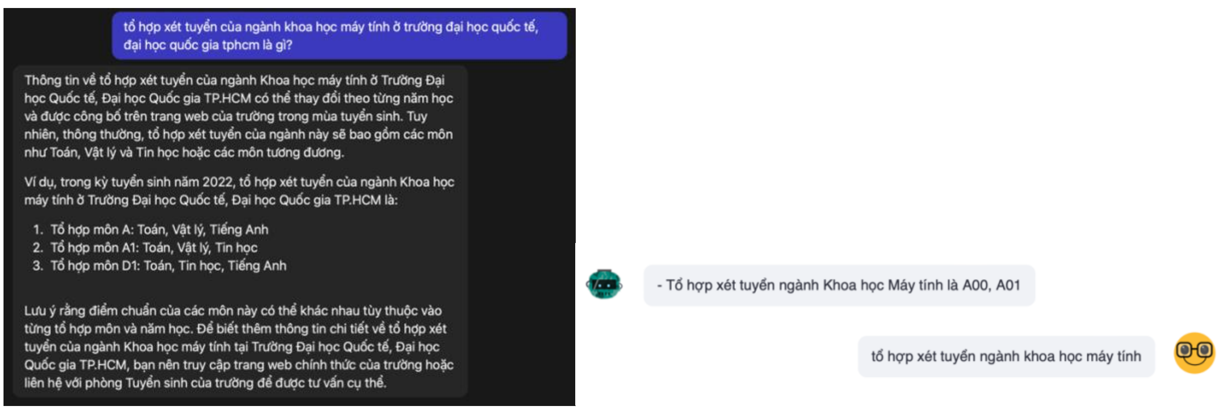

Figure 4. 11: GPT-3.5's answer (left) and DNN-based's answer (right) in Computer science admissions complex

4.4.2 Level 2 questions

In the group of secondary school questions, there will be questions about the strengths of high school students, what majors should they choose for the future (Table 8). The answer of GPT-3.5 is completely grounded and very natural, with more detailed explanation, but there are industries that do not have IU training. While DNN has a shorter answer, it doesn't have the naturalness like GPT-3.5.

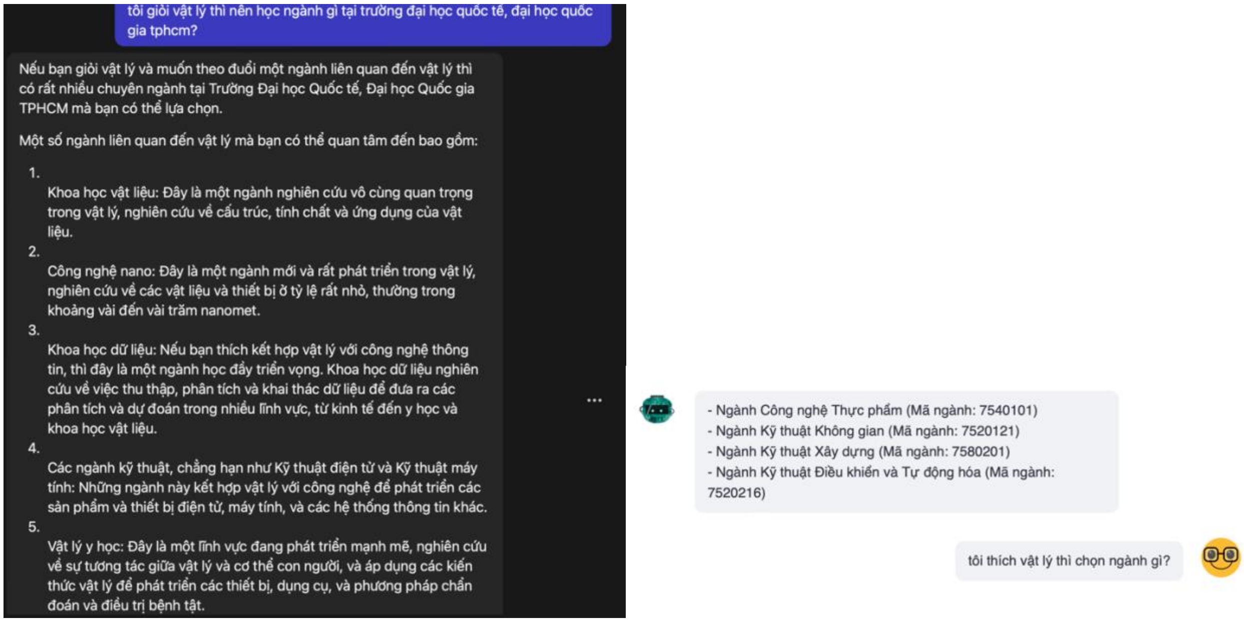

Figure 4. 12: GPT-3.5's answer (left) and DNN-based's answer (right) in strong subjects in high school

4.4.3 Level 3 question

In the group of level 3 questions, GPT-3.5 and DNN will base on the user's preferences to give them suggestions on the field of study they can choose. GPT-3.5's answer is very accurate and natural, but it does not seem to understand the question very well, its answer seems to be a general answer, not IU.

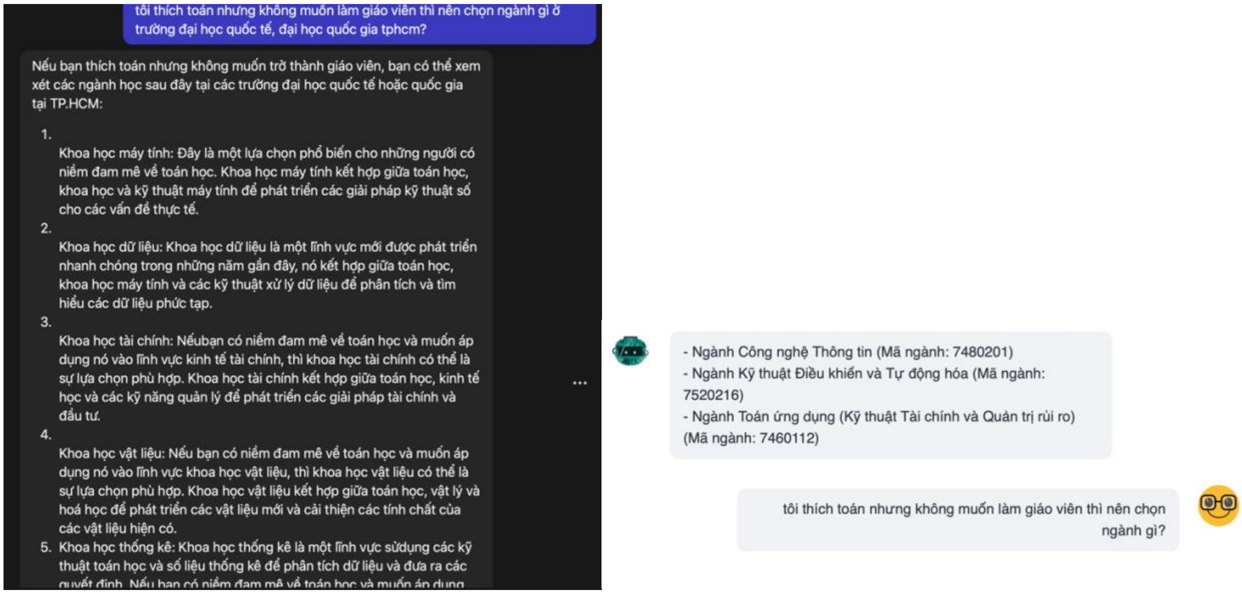

Figure 4. 13: GPT-3.5's answer (left) and DNN-based's answer (right) in answering based on user's interest

4.4.4 Remark

We may draw conclusions about the two models based on the case studies presented above. Regarding GPT-3.5, its response is very natural, very similar to a human response, but it appears that there are some questions that it does not understand very well, possibly because it has not been studied (for example, if the question is about training at IU, GPT answers those who are not trained here). In the case of unlearned queries, GPT does not respond that it does not know, but instead mixes words and provides inaccurate information that might harm users. Regarding DNN-based training, it is based on a dataset of admissions consultants, so the replies are perfectly good, but they are not natural and still appear to be bot responses. Therefore, IU's enrollment counseling support system based on DNN can replace GPT-3.5. However, it needs to be further improved in terms of data and communication.

Table 9: Comparison table between GPT-3.5 and DNN-based

	GPT-3.5	DNN-based
Strength	Naturally, every question has information to answer	Answer correctly in the context of the question
Weakness	Answer in general, without focusing on the context in the question; There are still incorrect and incorrect answers to the question	Unnatural, little data, so the information is limited

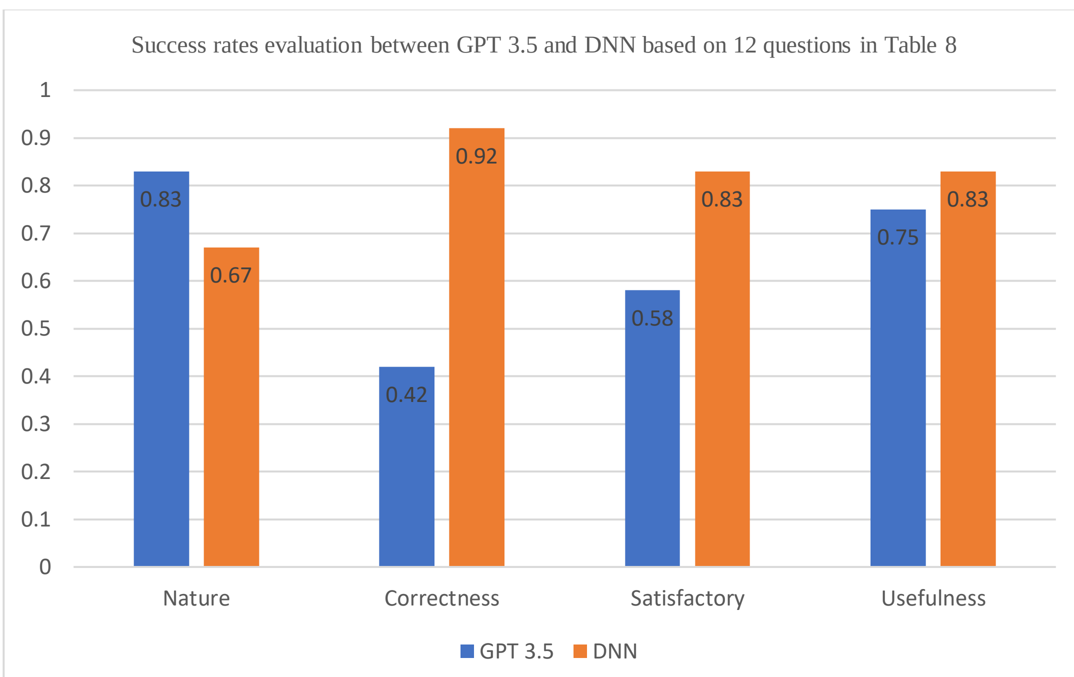

Figure 4. 14: Success rates evaluation between GPT 3.5 and DNN based on 12 questions in Table 8

4.5 Demo Web Application

A finished model will not see its usefulness if it is not designed in an easy and scientific way. The user cannot open the code and run it. Therefore, building a web application for users to interact is extremely necessary. Streamlit will be used during the demo because of the features that are easy to use and take less time to get used to by the developer. However, this is just a demo, to get it perfect, it is necessary to coordinate with partners with Frontend expertise.

The application is designed to help you communicate with our chatbot using text and voice commands. You can start by typing your message in the input text box and clicking the send button to send it to the chatbot.

If you want to clear your message history, simply click the clear text button. If you want to start a new conversation with the chatbot, click the clear context button to clear the chatbot's memory of previous conversations.

For an even more seamless experience, you can use the voice button to speak to the chatbot directly. Simply click the voice button and start speaking your message. The chatbot will automatically transcribe your voice message and respond accordingly. Figure 4.15 is about the text input, and Figure 4.16 is voice input.

Figure 4. 15: Demo Web Interface

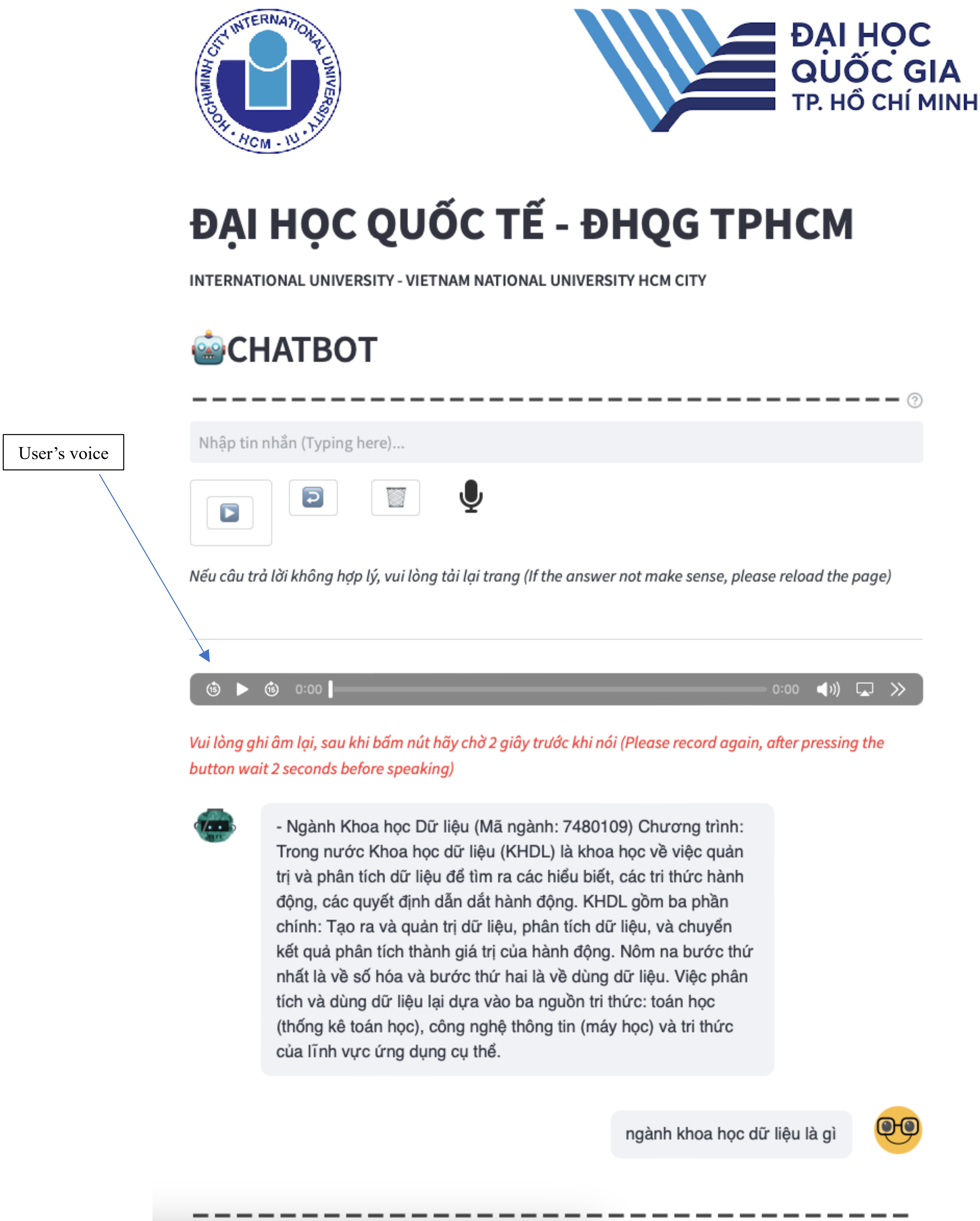

Figure 4. 16: Demo Web Interface (cont...)

Chapter 5 Conclusion and Future work

5.1 Conclusion

The DNN-based chatbot model with voice input can perform well when it comes to responding to queries within the knowledge-based application area. The results of the experiments have demonstrated that the suggested chatbot structure is viable and agreeable. While other chatbots could not provide lengthy responses, it may offer significant responses. This chatbot is particularly helpful for advising future students at Vietnamese universities on educational options. Vietnamese is an extremely difficult language to handle in NLP, to get a standard dataset to train, we need to process quite a few steps. In which, word segmentation, labeling, and content classification are mandatory. Additionally, processing input data specifically Vietnamese word segmentation, question classification, and answer labeling is crucial for training a chatbot model based on a Deep Neural Network. Up to 7% accuracy improvement is possible, if there are more hardware resources along with increasing the number of layers and the number of nodes for each layer, the accuracy can increase even more. Therefore, if you want a system that uses the multilayer processing technique to automatically answer questions, the process of processing incoming data in Vietnamese is crucial and essential.

5.2 Future work

The chatbot's reasoning process will be enhanced in the future to produce more accurate responses. To increase the capacity to comprehend various words in queries, a synonym database might be employed. The hardware resource upgrade will be done, and the results will be improved significantly because then we can increase the number of nodes of each layer, the accuracy will increase by about 5%. Soon, the English version of chatbot will be build.

References

[1] Jason Weston & Kurt Shuster, "Blender Bot 2.0: An open source chatbot that builds long-term memory and searches the internet," 16 July 2021. [Online]. Available: https://ai.facebook.com/blog/blender-bot-2-an-open-source-chatbot-that-builds-long-term-memory-and-searches-the-internet/.

[2] Jeffrey Matayoshi, Eric Cosyn, HasanUzun, "Are We There Yet? Evaluating the Effectivenessof a Recurrent Neural Network-Based StoppingAlgorithm for an Adaptive Assessment," *International Journal of Artificial Intelligence in Education manuscript No. ,* 2019.

[3] Amazon, "AWS," [Online]. Available: https://aws.amazon.com/what-is/neural-network/#:~:text=Deep%20neural%20networks%2C%20or%20deep,one%20node%20suppresses%20the%20other..

[4] J. Johnson, "BMC," 27 July 2020. [Online]. Available: https://www.bmc.com/blogs/deep-neural-network/.

[5] Baeldung, "Baeldung," [Online]. Available: https://www.baeldung.com/cs/porter-vs-lancaster-stemming-algorithms#:~:text=Like%20the%20Porter%20stemmer%2C%20the,the%20word%20goes%20through%20them..

[6] B. Iancu, "Evaluating Google Speech-to-Text API's Performance for Romanian e-Learning Resources," *Informatica Economica ,* pp. 17-25, 2019.

[7] B. T. Tung, "VIBLO," 19 8 2020. [Online]. Available: https://viblo.asia/p/language-modeling-mo-hinh-ngon-ngu-va-bai-toan-them-dau-cau-trong-tieng-viet-1VgZveV2KAw.

[8] I. Papastratis, "AI Summer," 14 07 2021. [Online]. Available: https://theaisummer.com/speech-recognition/.

[9] Thi Thanh Sang Nguyen*, Dang Huu Trong Ho and Ngoc Tram Anh Nguyen, "An Ontology-Based Question Answering System for University AdmissionsAdvising," *Tech Science Press Intelligent Automation & Soft Computing,* 2022.

[10] S. Tirpude, Dr. A. S. Alvi, "Closed Domain Keyword based Question Answering System for Legal Documents of IPC Sections & Indian Laws," Computer Science, 2015.

[11] E. Dündar, Tolga Çekiç, Onur Deniz, S. Arslan, "A Hybrid Approach to Question-answering for a Banking Chatbot on Turkish: Extending Keywords with Embedding Vectors," in *International Conference on Knowledge Discovery and Information Retrieval*, 2018.

[12] Thanh Vu, Dat Quoc Nguyen, "VnCoreNLP: A Vietnamese Natural Language Processing Toolkit," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 2018.

[13] Vu Anh, Nguyen Dang Duc Tai, Bui Nhat Anh, Vuong Quoc Binh, Doan Viet Dung, "Under The Sea," 2018. [Online]. Available: https://underthesea.readthedocs.io/en/latest/api.html#word-tokenize.

[14] Widodo Budiharto, Vincent Andreas & Alexander Agung Santoso Gunawan, "Deep learning-based question answering system for intelligent humanoid robot," *Journal of BigData,* 2020.

[15] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi, "Bidirectional Attention Flow for Machine Comprehension," in *Computation and Language*, 2018.

[16] A. Bouziane, D. Bouchiha, N. Doumi and M. Malki,, "Question answering systems: Survey and trends," *ProcediaComputer Science,* vol. 73, p. 366–375, 2015.

[17] JunnanLi, DongxuLi, CaimingXiong, StevenHoi, "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation," Cornell University, 2022.