Computers and Education: Artificial Intelligence 2 (2021) 100036

Contents lists available at ScienceDirect

Computers and Education: Artificial Intelligence

journal homepage: www.sciencedirect.com/journal/computers-and-education-artificial-intelligence

NEU-chatbot: Chatbot for admission of National Economics University

Trung Thanh Nguyen [a,*], Anh Duc Le [b], Ha Thanh Hoang [c], Tuan Nguyen [d,**]

[a] Faculty of Information Technology, Hanoi University of Science and Technology, Viet Nam
[b] Vice Head of Academic Affairs Office, National Economics University, Viet Nam
[c] Academic Affairs Office, National Economics University, Viet Nam
[d] Faculty of Mathematical Economics, National Economics University, Viet Nam

ARTICLE INFO

Keywords:
Rasa
Admission chatbot
RNN
BERT

ABSTRACT

In the last few years, intelligent chatbot systems have been prevalent in various application fields, especially in education. Therefore, the demand for such online consulting services like chatbots is getting higher respectively. However, most communications between potential students and universities are performed manually, which is very time-consuming procedure, becoming a burden on the head of admissions. In this paper, we introduce an AI-based chatbot where students can instantly get daily updates of curriculum, admission for new students, tuition fees, IELTS writing task II score, etc. Our chatbot was developed by Deep Learning models, which are already integrated into the Rasa framework. We also proposed a rational pipeline for Vietnamese chatbots with our data preprocessing to obtain optimal accuracy and to avoid the overfitting of the model. Our model can detect more than fifty types of questions from users' input with an accuracy of 97.1% on test set. The chatbot was applied for National Economics University's official admission Fanpage on the Facebook platform, which is the most famous social network in Vietnam. This research shows detailed guidelines on how to build an AI chatbot from scratch, and the techniques we used, which can be applied to any language globally.

1. Introduction

Virtual assistants or chatbots are software agents that interact with users through natural language conversation (Følstad & Brandtzæg, 2017). Recently, chatbots have crossed the boundaries of commercial, and are helpful in various fields of education, business and e-commerce, health, and entertainment (Shawar and Atwell, 2007a). Several studies have revealed that chatbots can bring entertainment to users, provide instant feedback, enhance peer communication skills (Hill et al., 2015), and improve students' learning efficiency (Wu et al., 2020). In education, a chatbot is needed to automatically answer all the questions related to a certain university without any human interference. Via this program, high school students and their parents can quickly access the admission information from anywhere and at anytime, which is expected to work for 24 h (Al Fakhri et al., 2019). Hence, they can proactively register and adjust their aspirations to ensure the opportunities involving in the university courses. In addition, this chatbot helps to reduce the overhead cost for the admission department.

To build an effective Chatbot, researchers and developers need to have a solid foundation in deep learning, a subset of machine learning in artificial intelligence. Usually, deep learning is referred to as deep artificial neural networks, a set of high-level abstract modeling algorithms on data using transformation of nonlinear functions arranged by many deep layers (Dadang, 2018). Deep learning has set new records in accuracy for various significant problems, such as natural language processing (NLP). NLP is a branch of artificial intelligence that helps computers understand, interpret, and manipulate human language. NLP researchers aim to gather knowledge on how human beings understand and use language to create appropriate tools and techniques helping computer systems understand and manipulate natural languages and perform desired tasks (Chowdhury, 2003).

An AI chatbot learns everything from data accumulated from human-to-human dialogues. The more data it is fed, the more effective its learning will be. There are two primary types of deep learning chatbots: retrieval-based bots and generative bots. Slightly different from rule-based bots, a retrieval-based bot offers more flexibility. It builds a classification model to extract the intent and all needed information from users' input. Afterwards, are trieval-based bot will select the

* Corresponding author.
** Corresponding author.
E-mail addresses: trung.nt184318@sis.hust.edu.vn, trung.nt184318@sis.hust.edu.vn (T.T. Nguyen), ducla@neu.edu.vn (A.D. Le), htha@neu.edu.vn (H.T. Hoang), nttuan@neu.edu.vn (T. Nguyen).

https://doi.org/10.1016/j.caeai.2021.100036
Received 9 June 2021; Received in revised form 24 September 2021; Accepted 3 October 2021
Available online 14 October 2021

appropriate message from the database with the highest confidence to respond to users. Meanwhile, a generative bot can generate the answer based on current and previous user messages. This type of chatbot is more human-like compared to retrieval-based bots because it has some specific memory savings scheme. However, there are several difficulties in building and training it. Generative bots do not seem to work well in practice; they often make grammatical mistakes, and their replies are inconsistent and nonsensical. In this paper, we used the retrieval-based model to build NEU-chatbot due to the limitation of questions and to avoid the aforementioned mistakes.

Nowadays, there is a wide range of AI chatbot frameworks available such as Rasa, Chatfuel, Mobile Monkey, etc. In this work, we used the Rasa framework to build the chatbot for some reasons. Firstly, Rasa is an open-source chatbot framework that began as a project on GitHub in 2016. Thus, it is easy to integrate and customize Rasa components. Secondly, Rasa supports the connection with other messaging apps and the deployment in multiply environments. Moreover, the implementation of deep learning on chatbots requires the role of the chatbot framework, which is for unifying and pipelining the deep learning models integrated into the Rasa framework. The aim and philosophy of the initiators of the Rasa chatbot framework are to create Machine learning-based dialogue management that provides ease of use in terms of implementation as well as bootstrapping by starting the stages from existing sources to create something more complex in a more efficient way, even with minimal initial training data (Bocklisch et al., 2017). Finally, Rasa is a library of tools in Python programming language, which is an appropriate choice for projects including NLP tasks because of its simple syntax and transparent semantics.

The primary goal of our study is to investigate how a chatbot can reduce the burden of admission counseling by automatically assisting admissions, student support and providing immediate access to information. To achieve this goal, we developed NEU-chatbot, a chatbot functioning to provide admission information at National Economics University (NEU), founded in 1956 with 54 specializations. As one of the leading universities in Economics, Public Management, and Business Administration in Vietnam, approximately 6000 students enroll in NEU courses each year. To meet the high demand for admission information, NEU-chatbot was created with the mission of answering all admission-related questions. We experimented with a dataset that includes over 50 intents with approximately 1500 examples. With rational pipeline and data preprocessing, our bot can detect the users' intents with an accuracy of 97.1% on test set. After three months of adding this chatbot to the NEU official admission Fanpage on Facebook, we have received more than 50,000 questions in total from students and parents about NEU enrollment procedures. In detail, the chatbot appropriately answered 90.29% of the questions because there were many irrelevant questions. However, if only inquiries related to the NEU admission count, the prospect of appropriate replies would have been comparatively 95.79%, The rest of the paper is organized as follows: Section 2 describes the existing works. Section 3 illustrates a method to build a chatbot for admission; and techniques to customize actions in Rasa for Vietnamese language. In section 4, we will show and discuss the result and our own experience. The evaluation of our work is introduced in section 5, followed by Section 6, the conclusion and our future work.

2. Related work

In (Prashant et al., 2017), an online chatting system for college inquiries using a knowledgeable database did pattern matching to perform chatbots' information retrieval. All the detailed working steps in this research are clear with UML and various process diagrams. However, their bot did not use a machine learning approach, so that is still too rigid by rules for students and parents to make inquiries. The patterns were built manually, so it does not scale in the real use case.

In (Thakkar et al., 2018), Erasmus is an AI-based chatbot, which answers questions on university information. The authors designed Erasmus as an end-to-end system using cloud services, starting from api.ai (Dialogflow), Mlab (MongoDB cloud), IBM Bluemix (webhook API). However, their chatbot took quite a long latency in responding to the users because it uses too many cloud services.

Another work to compare with is Eaglebot (Rana, 2019). The authors implemented an approach for semantics search to answer users' inquiries about a university, based on the information on the university's website. One of their approaches involved document retrieval and paragraph selection using BERT. It is a scalable chatbot system utilizing three route selection methods with the main framework using Dialogflow. However, Eaglebot still has some limitations because of the request limits in the Dialogflow chatbot framework, and the bot only achieved 56% accuracy.

The authors (Windiatmoko et al., 2020) created a chatbot integrated with MySQL database and API for University inquiries with explanations step by step. However, this simple chatbot is only capable of answering clients with a small number of intents. In addition, Indonesian is quite different from other languages like English or Vietnamese, and the authors did not mention how they customized their tokenizer and pipeline. The study only used the basic structure given by Rasa when the project was initialized without utilizing the platform's advanced capabilities.

Many studies on generative chatbot models are based on sequence-to-sequence networks called encoder-decoder models (Vinyals & Le, 2015). These models require a vast single-turn dialogue corpus for training. However, it is not easy to gather a high-quality dialogue corpus with millions of entries except for English and Chinese in practice. As a result, to the best of our knowledge, we rarely find any generative chatbots for other languages, especially Vietnamese.

In (Segura et al., 2019), a social chatbot for football was designed to answer questions about the Spanish football league called "La Liga". The chatbot is deployed as a Slack client, with text-based interaction. It uses SPARQL queries to retrieve data from Wikidata to generate responses; and a retrieval-based chat-oriented dialogue to reply to out-of-scope user queries. Nevertheless, one of the most apparent generation-based chatbot's drawbacks is that we cannot control the outputs delivered to clients. These outputs may contain sensitive contents or grammatical errors, which are unsuitable for formal fields such as education.

A rational pipeline is considered as one of the essential parts of each Rasa project to build an efficient AI-based chatbot. In (Jiao, 2020), the author Anran Jiao compared the performance of the Rasa NLU stack with Neural Network Classifier and Entity Extractor. The research stated that the Rasa NLU method outperforms by extracting all the entities and classifying users' intents. As a result, we established the NLU pipeline for Vietnamese chatbots using the Rasa platform with our data preprocessing and the renowned pre-trained model like BERT in NEU-chatbot. Also, if we only connect the Rasa chatbot through the webhook channel, it will be lagged when about 25 users connect simultaneously. To deal with this problem, we have constituted a gateway provided by the proxy server. Thus, the chatbot can respond to up to 50 users at the same time. Then, we deployed the chatbot to Messenger so students and parents could easily access the services. We also replaced the long boring answer text with the image to be visualizable for the user. These techniques can also be applied to other chatbots in any language.

3. Method

In this study, we used Rasa platform to develop NEU-chatbot. Rasa Core and Rasa NLU are the two main components of the Rasa platform. Rasa Core is used to handle the flow of conversation, utterances, and actions, while Rasa NLU is used to understand, classify the intents, and extract entities of the text inputs. In other words, NLU gives the intent; the Rasa core performs the action corresponding to it, and the bot replies with that action. This section shows how we constructed a Rasa chatbot for inquiries of National Economics University.

3.1. Rasa platform

Rasa is an open-source project based on natural language understanding (NLU), dialogue management, and interactions. A tracker object functions to manipulate a dialogue. Each conversation session will have only one tracker object. This tracker will store the slots and a log of all the events that occurred within a conversation.

In Fig. 1, all consecutive steps, except for step 1 performed by Rasa NLU, are handled by Rasa Core. After the input message is received and then forwarded to the interpreter to extract intents, entities, and needed information, Rasa NLU and tracker will track, detect, and maintain the conversation context's status through the received message notifications. Then the Tracker (context status) output passes into the policy manager, and the policy will determine the next action. In step 5, the tracker records the actions before they are executed and sent to the users, using the specific predefined utterance in domain.yml, which defines the universe including intents, entities, slots, and actions. If the users neglect the executed actions, the process returns to the third step.

3.2. Preprocessing data

Data preprocessing can often significantly impact a supervised machine learning algorithm (Kotsiantis et al., 2006). Text preprocessing means cleaning noise, such as stop words, punctuations, and terms that do not carry much weight in context (Kalra & Aggarwal, 2017, October). In this paper, we present some techniques to achieve better results when constructing chatbots in general and chatbots for admission to the university in particular.

In our project, we divide preprocessing part into four main factors:

Add diacritics: This step aims at correcting spelling mistakes in the user's input. Other languages have different processing methods, but Vietnamese uses a Latin-script based Vietnamese alphabet where the lexicon contains native Vietnamese words derived from the Latin script. The official Latin-based Vietnamese alphabet consists of twenty-nine letters: seventeen consonants and twelve vowels except for "f, j, w, z" and twenty-two letters from the Roman alphabet. However, the most challenging issue of handling Vietnamese words is diacritical marks. There are 134 letters, including uppercase and lowercase, with Vietnamese diacritics such as "á, ạ, ê, ó, ô". To type Vietnamese tones, users normally use the Vietnamese keyboard. However, if a user's laptop or mobile is not integrated with the Vietnamese keyboard, the default one does not allow them to type diacritics. To address this problem, we wrote a function, namely TelexConvert, to convert all the sentences having the intent to use Vietnamese tones for sentences with diacritics. For example, the input word of the user is "nhieeu", with TelexConvert function, we obtain "nhiều" for the output. The latter is more understandable for the user and will fit our dataset.

Clean text: the purpose of this part is to remove noise from data, which means removing meaningless symbols like "$%&#". The simplest way to deal with this problem is using a filter based on regular expressions.

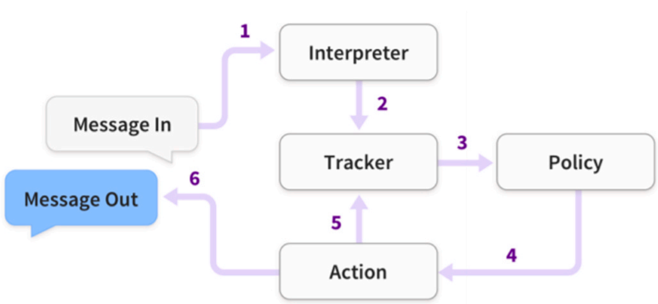

Fig. 1. Rasa stack framework.

Remove StopWords (SWs): StopWords often appear in natural language; however, they have little or no meaning. In Vietnamese, SWs are words like "vâng, này, kia" corresponding to "is, that, this" in English. There are many ways to eliminate SWs, and in this paper, we chose to use a dictionary. We have compiled all the SWs in Vietnamese in a list, and any SWs included in the input will be removed after going through StopWordRemover function that we wrote in preprocesser.py.

Convert number to text: Since we were building a chatbot that handles text inputs, we expected that all information should be in texts. After converting number to text, we obtained better results compared to the original data.

3.3. Choosing rasa NLU pipelines

The primary purpose of Rasa NLU is to analyze the information given by the user to the chatbot. This information contains the intents and entities needed to be extracted. In Rasa Open Source, incoming messages are processed by a sequence of components. These components are executed one after another in a so-called processing pipeline defined in your config.yml (RASA, 2021c). With the supervised embeddings pipeline, you can train with any languages globally because this work can train anything from scratch. This kind of pipeline is something like the one given in Fig. 2. Firstly, WhitespaceTokenizer splits text into tokens with white spaces as a separator. Then the user message will be converted into a vector using regular expression. CRF Entity Extractor will extract entities, such as person names or location, from the user message. If the training data contains defined synonyms, Entity Synonym Mapper will make sure that the detected entity values will be mapped to the same value. Then Count Vectors Featurizer creates bag-of-words representation of user messages, intents, and responses. The first featurizer counts whole words; the second one looks at sub-word sequences of characters. Finally, Embedding Intent Classifier assigns one of the intents defined in the domain file to incoming user messages.

Rasa has provisions for authoring new components and altering existing ones (Rafla & Kennington, 2019). Besides the above pipeline, we customized a rational pipeline for the chatbot using the state-of-the-art language model BERT to use pre-trained word vectors. Although BERT is considered a heavy model and is considered approximately six times slower than the ConveRT model, we do not use the ConveRT pipeline because it is only used for English and does not support Vietnamese. The example of the BERT pipeline that we have used for NEU-chatbot is shown in Fig. 3.

We need to pay attention to two essential components in this pipeline, which are Tokenizer and Featurizer. The choice of Tokenizer might affect the type of Featurizer, and the order of the components of the NLU pipeline needs to be considered. For instance, you cannot define a Featurizer before a Tokenizer because the latter's output acts as the input of the former. In most cases, WhitespaceTokenizer performs well

```
language: "en"

pipeline:
- name: "WhitespaceTokenizer"
- name: "RegexFeaturizer"
- name: "CRFEntityExtractor"
- name: "EntitySynonymMapper"
- name: "CountVectorsFeaturizer"
- name: "CountVectorsFeaturizer"
  analyzer: "char_wb"
  min_ngram: 1
  max_ngram: 4
- name: "EmbeddingIntentClassifier"
```

Fig. 2. Supervised embeddings pipeline.

```yaml
language: vi

pipeline:
  - name: preprocesser.VietnamesePreprocesser
  - name: HFTransformersNLP
    model_name: bert
    model_weights: bert-base-uncased
    cache_dir: null
  - name: WhitespaceTokenizer
  - name: RegexFeaturizer
  - name: CRFEntityExtractor
  - name: EntitySynonymMapper
  - name: CountVectorsFeaturizer
  - name: CountVectorsFeaturizer
    analyzer: char_wb
    min_ngram: 1
    max_ngram: 4
  - name: DIETClassifier
    epochs: 400
  - name: EntitySynonymMapper
  - name: ResponseSelector
    epochs: 100
  - name: FallbackClassifier
    threshold: 0.65
    ambiguity_threshold: 0.1
```

Fig. 3. NEU-Chatbot-NLU pipeline.

for any languages. However, you need to customize your tokenizer if you construct a chatbot with hieroglyphs like Chinese text, which has no whitespace between words, not even between sentences. Moreover, pre-trained models like BERT tend to be resource-intensive and time-consuming. So besides using BERT for embedding, we also use Dual Intent and Entity Transformer (DIET) to handle both intent classification and entity recognition together. In (Blog, 2020), DIET is proven to improve upon the current state of the arts and is six times faster to train. Furthermore, it parallels large-scale pre-trained language models in accuracy and performance.

3.4. Policies

When you implement a dialogue management solution, the main task is to decide what happens next concerning the conversation. The class rasa.core.policies.policy decides what action to be taken during each conversation steps with bot (Sharma & Joshi, 2020). We can redo the policies that our chatbot uses by indicating the policy in config.yml file.

There are two types of policies: machine-learning policies and rule-based policies. These policies will be used in tandem to help the bot decide which actions should be taken at each step in a conversation. Depending on our purposes, we can choose multiple policies in a single configuration. The Table 1 shows the details of each type.

3.5. Custom actions

An action can run any code. Custom actions can turn on the lights, add an event to a calendar, and check a user's bank balance (Legacy--docs RASA).

In our project, if the bot receives some messages with low classification confidence, we will use Fallbacks, which helps to ensure the gracefulness in handling these low confidence messages. However, these circumstances often occur when users ask for things that are out of the scope of your bot's domain. Instead of sending a mere misunderstanding message to users, we created a form through slots (RASA, 2021b), containing information provided by the user, which serves as bot's memory, to fill the questions. If they wish to send it to us, we will receive an email indicating the reason why the bot misunderstood the message; and also the request of the user. Because our bot could answer approximately 96% of the questions related to the NEU admission, we only received about 10–15 emails per day in terms of misunderstanding messages. This action is beneficial when your scales of users are enormous. You need to check the email to figure out the problem. If this problem happens quite often, you should add another intent for the bot to handle these cases.

Another activity that we would like to discuss in the section is checking of IELTS Writing task II band score. It is widely known that more than 10,000 international organizations trust IELTS, so when we take the test, we can be confident that it is recognized by educational institutions, governments, and professional bodies worldwide. Moreover, IELTS is one of the graduation requirements at National Economics University in Vietnam. We created an intent that allows users to input the question and the text of Writing task II. After all the required information is provided, an API will be called from another website through an HTTP request. This website will scan the text given by our bot for all types of mistakes, from typos to sentence structure problems and beyond. After that, hundreds of algorithms will assess the writing

Table 1
List of policies.

Type of Policy	Name of policy	Featurization
Machine learning policy	TED policy – the transformer embedding dialogue	The policy concatenates on user input, system actions, and slots.
	Memorization	The policy remembers stories from the training data. It checks the matching story of the current conversation and predicts the next action from the matching story with the confidence of [0,1]. The number of conversation turns is indicated in max_history.
	Augmented Memoization Policy	The policy remembers examples from matching story for up to max_history turns. It is similar to the Memoization Policy. In addition, this policy has a forgetting mechanism.
Rule-based policies	Rule policy	The policy handles conversation parts that follow a fixed behavior and makes predictions using rules that have been in the training data.
Configuring Policies	Max History	The policy controls the size of the dialogue history that model looks at to predict the next action
	Data Augmentation	The policy determines how many augmented stories are subsampled during training.
	Featurizers	The policy allows the use of machine learning algorithms to build up vector representations of conversational AI.

according to four evaluation criteria. Then the server will send us a link to the result, and the bot will utter a message containing the result link to the user.

3.6. Connector module platform

Rasa open source provides many built-in connectors to connect to common messaging and voice channels. You can also connect to your website or app with pre-configured REST channels or build your own custom connector (RASA, 2021a). There were about 75,180,000 Facebook users in Vietnam up to April 2021, which accounted for 75.5% of its entire population. We decided to use Facebook Messenger instead of other channel connectors like Slack Telegram, Twilio, etc. Moreover, Facebook Messenger also allows us to attach an image with text; this technique is beneficial when we want to transfer a long message without boring text.

As a connector module platform, to connect to Facebook Messenger, we first need to set up a Facebook page and install the app to get credentials from Facebook Developer. Once we have them, we can add these to our credentials.yml. Then insert the call-back URL, which will look like "https://<HOST>/webhooks/facebook/webhook", Verify Token, App Secret, and page access token into credential.yml file. Supposed that the quantity of users accessing the bot at the same time is beyond the limitation, you will need to create a proxy server, which provides a gateway between the user and the internet. Proxy servers are common solutions to relieve organizational networks from heavy traffic by storing the most frequently referenced web objects in their local cache (Tsui et al., 2013). So now, the chatbot will receive real-time HTTP notifications of changes to specific objects and postback messages.

4. Experiment

4.1. Experimental setup

In this work, we experimented with a dataset that includes over 50 intents, inquiring about main issues such as tuition fees, admission forms, scholarships, or deadlines, with approximately 1500 examples. As you can see from Table 2, if you want to ask about a scholarship at National Economics University in Vietnam, you can send the message "Tiêu chí xét duyệt học bổng của trường NEU là gì?" ("What were NEU's scholarship criteria?"). Then, Rasa NLU will extract all the needed information and know that the user's intent is "hoc_bong" (scholarship). In addition, we also created a global 80% train and 20% test split from nlu.yml because our dataset contained considerably high observations. Later, the training data was split into five groups of equal size to apply k-fold cross-validation. Although this cross-validation strategy is computationally expensive, it offers an effective model when our dataset is not too big. After the training data was split into five-folds, in the first iteration, the first fold was used to test the model, and the rest were used to train the model. In the second iteration, 2nd fold was used as the testing set while the rest served as the training set. This process was repeated until each of the five folds was used for testing. Since we do not know which intent the conversation will begin, we have added each story in file stories.yml with a single intent mentioned in domain.yml file. The pipeline and policies are in config.yml.

Our preprocessing data was created and then stored in preprocesser.py. With 19 faculties at NEU, there will be a considerable number of questions if clients are able to ask specific questions about each specialization. To deal with this problem, we built an admission chatbot for National Economics University and are constructing a chatbot system so that each faculty at NEU will have a chatbot. Thus, messages from users will go through a filter that enables classification of the faculty included in a message by mapping the entities extracted by Rasa NLU with a list of faculties' names, and the bot will automatically switch to this faculty's chatbot. Conversations are a lot more than simple text messages when a bot is built on the Messenger Platform. In addition to text, the platform allows you to send rich media, like audio, video, and images (Facebook for Developer, 2020). Therefore, we customized the input of Facebook Messenger to help our bot utter rational content, and this work is located in CustomFBInput.py. For example, when people send our bot an image, it will be hard for our bot to understand the content of this image. As a result, the input of Facebook Messenger is customized to detect the type of message. Then the image URL is converted to the form of message our bot can understand. Thus, our model can define the appropriate actions to respond to users like sending back icons or messages to notify the user about this issue.

Table 2
Intent example in Vietnamese example in English.

Intent	Example in Vietnamese	Example in English
Scholarship	Tiêu chí xét duyệt học bổng của trường NEU là gì?	What were NEU's scholarship criteria?
Dormitory	Em muốn đăng ký ký túc xá thì cần làm thủ tục gì	How do I apply for a dormitory?
Application	Hồ sơ nhập học cần có những tài liệu gì ạ?	What are the documents needed for admission?
Deadline	Hạn cuối nộp hồ sơ là khi nào?	When is the deadline for admission application?
English proficiency	NEU chấp nhận các chứng chỉ tiếng anh nào?	Which English proficiency certificates are accepted by NEU?

4.2. Experiment results

Once we have finished our model, the most important question arises: how good is our model? In (Shawar and Atwell, 2007b), to measure the quality of each response, we need to classify responses according to an independent human evaluation of "reasonableness": reasonable reply, weird but understandable, or nonsensical reply. After three months of applying the NEU chatbot, we conducted a survey on Facebook about the satisfaction level of users, and the result will be discussed in section 5. In this section, we will evaluate our chatbot via precision (1), f1-score (3), and accuracy (4) metrics. A true positive (TP) is an outcome where the model correctly predicts the positive class. Similarly, a true negative (TN) is an outcome where the model correctly predicts the negative class. A false positive (FP) occurs when you predict an observation belonging to a class while it is not in reality. And a false negative (FN) is an outcome where the model incorrectly predicts the negative class. Then,

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

The intent classification is evaluated by F1-score, accuracy, and precision using cross-validation. Table 3 shows the micro-average score on the chosen model.

Table 3
Intent classification.

Metric	Score
F1-score	0.976
Accuracy	0.971
Precision	**0.979**

5. Evaluation

After three months of adding this chatbot to the NEU official admission Fanpage on Facebook, we have received more than 50,000 questions in total from students and their parents about the procedures of NEU enrollment. In detail, the chatbot appropriately answered 90.29% of the questions because many irrelevant questions were asked. However, if only inquiries related to the NEU admission counted, the prospect of appropriate replies would be comparatively 95.79%. We also carried out a survey on Facebook about the satisfaction level of users. The result revealed that almost 98.61% of 1000 clients giving inquiries were happy with the answers of this chatbot. In the other aspect, the chatbot has brought about several benefits, for instance, the number of admission consultants can be cut down by 80%, but outstanding service quality is still guaranteed. Our bot can answer all the questions automatically at any time without any human interference. Furthermore, all the chatbot's utterances must conform to a pre-defined format and content, so the information is consistent and there is no grammatical error or amiss information. For all the above-mentioned factors, it is rational to apply this chatbot to the admission to any universities, especially the National Economics University, reducing the burden of admission counseling.

6. Conclusion and future work

NEU-chatbot was developed with the aim of assisting prospective students and parents with admission inquiries related to National Economics University. This approach introduces users to new and emerging technological solution for optimal and real-time feedback in the educational sector. With this solution, the admission officer's workload will be reduced and the rate of inconsistencies and false admission information will decline considerably. By adopting our techniques, people can easily build their own chatbot with Rasa platform and customize the chatbot's actions to accommodate their purposes. Based on the promising result in this research, we can leverage the technology to achieve new levels of productivity, implement useful digital tools to expand learning opportunities for students, and increase student support and engagement. It can provide students with easy-to-access information, accelerated learning, and opportunities to practice what they learn.

NEU-chatbot has achieved 97.1% accuracy on the test set, and this chatbot is practically applied for admission to National Economics University in Vietnam. However, the content of nlu.yml file needs to be updated manually each year by staff with new intents and records to adapt to new academic year information and address the misunderstanding of intents, which occurs during the consulting process. Training the bot as thoroughly as possible will improve its accuracy. And with time, the performance of the chatbot will be upgraded. In the future scope of this study, we will build another model like FastText, an open-source, free, lightweight library that allows users to learn text representations and text classifiers. In (Joulin et al., 2016), FastText has been trained on more than one billion words in less than 10 min using a standard multicore CPU and classified half a million sentences among 312K classes in less than a minute. Besides using WhitespaceTokenizer, we will customize a tokenizer for only Vietnamese words to get the best result. Moreover, we are developing NEU-chatbot version two with voice assistant using Google Assistant platform, saving time for typing. Furthermore, we are also striving to develop the MBTI-test feature of NEU-chatbot, which will help students understand their personality type and suggest the appropriate field of studies.

Declaration of competing interestDoCI

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors of this paper sincerely thank all the members of the Data Science Lab at National Economics University for their support.

References

Al Fakhri, S., Lutfi, H. U., Wardana, W. K., Munawar, G., & Wisnuadhi, B. (2019, August). Aplikasi chatbot informasi Kampus Polban menggunakan aplikasi LINE Messenger. In *, 10. Prosiding industrial research workshop and National Seminar* (pp. 302–313). No. 1.
Blog, R. A. S. A. (2020). Introducing DIET: State-of-the-artarchitecture that outperform fine-tuning BERT and is 6X faster to train. Retrieved from https://blog.rasa.com/introducing-dual-intent-and-entity-transformer-diet-state-of theart-performance-on-a-lightweight-architecture/. (Accessed 5 June 2021).
Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017). *Rasa: Open source language understanding and dialogue management*. arXiv preprint arXiv:1712.05181.
Chowdhury, G. G. (2003). Natural language processing. *Annual Review of Information Science and Technology, 37*(1), 51–89.
Dadang, W. (2018). Memahami kecerdasan Buatan berupa deep learning dan machine learning. In *, 10. Prosiding industrial research workshop and National Seminar*. No. 1.
Facebook for Developer. (2020). Conversation components. Retrieved from https://developers.facebook.com/docs/messenger-platform/introduction/conversation components/. (Accessed 5 June 2021).
Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *Interactions, 24*(4), 38–42.
Hill, J., Ford, W. R., & Farreras, I. G. (2015). Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. *Computers in Human Behavior, 49*, 245–250.
Jiao, A. (2020). March). An intelligent chatbot system based on entity extraction using RASA NLU and neural network. *Journal of Physics: Conference Series, 1487*(1), 12014. IOP Publishing.
Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). *Bag of tricks for efficient text classification*. arXiv preprint arXiv:1607.01759.
Kalra, V., & Aggarwal, R. (2017, October). Importance of text data preprocessing & implementation in RapidMiner. In *ICITKM* (pp. 71–75).
Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised leaning. *International journal of computer science, 1*(2), 111–117.
Legacy-docs, R. A. S. A.. Actions. Retrieved from https://legacydocs.rasa.com/docs/core/0.14.5/run-code-in-customactions/#. (Accessed 5 June 2021).
Prashant, B. P., Anil, M. S., & Dilip, K. M. (2017). *Online chatting system for college Enquiry using knowledgeable database*.
Rafla, A., & Kennington, C. (2019). *Incrementalizing RASA's open-source natural language understanding pipeline*. arXiv preprint arXiv:1907.05403.
Rana, M. (2019). *Eaglebot: A Chatbot Based Multi-Tier Question Answering System For Retrieving answers From Heterogeneous sources using BERT*.
RASA. (2021a). Connecting to messaging and voice channels. Retrieved from https://rasa.com/docs/rasa/messagingand-voice-channels. (Accessed 5 June 2021).
RASA. (2021b). Domain. Retrieved from https://rasa.com/docs/rasa/domain. (Accessed 5 June 2021).
RASA. (2021c). Tuning your NLU model. Retrieved from https://rasa.com/docs/rasa/tuning-your-model. (Accessed 5 June 2021).
Segura, C., Palau, A., Luque, J., Costa-Jussà, M. R., & Banchs, R. E. (2019). Chatbol, a chatbot for the Spanish "La Liga". In *9th international workshop on Spoken dialogue system technology* (pp. 319–330). Singapore: Springer.
Sharma, R. K., & Joshi, M. (2020). An analytical study and review of open source chatbot framework, rasa. *International Journal of Engineering Research, 9*(6).
Shawar, B. A., & Atwell, E. (2007 a, April). Different measurement metrics to evaluate a chatbot system. In *Proceedings of the workshop on bridging the gap: Academic and industrial research in dialog technologies* (pp. 89–96).
Shawar, B. A., & Atwell, E. (2007 b, January). Chatbots: Are they really useful? *Ldv forum, 22*(1), 29–49.
Thakkar, J., Raut, P., Doshi, Y., & Parekh, K. (2018). Erasmus–AI chatbot. *International Journal of Computational Science and Engineering, 6*(10), 498–502.
Tsui, K. C., Kaiser, M. J., & Liu, J. (2013). *Distributed proxy server management: A self-organized approach*.
Vinyals, O., & Le, Q. (2015). *A neural conversational model*. arXiv preprint arXiv:1506.05869.
Windiatmoko, Y., Hidayatullah, A. F., & Rahmadi, R. (2020). *Developing FB chatbot based on deep learning using RASA framework for university enquiries*. arXiv preprint arXiv:2009.12341.
Wu, E. H. K., Lin, C. H., Ou, Y. Y., Liu, C. Z., Wang, W. K., & Chao, C. Y. (2020). Advantages and constraints of a hybrid model K-12 e-learning assistant chatbot. *IEEE Access, 8*, 77788–77801.