Building a Chatbot for Supporting the Admission of Universities

Conference Paper · November 2021

DOI: 10.1109/KSE53942.2021.9648677

Building a Chatbot for Supporting the Admission of Universities

Minh-Tien Nguyen*, Manh Tran-Tien*, Anh Phan Viet†, Huy-The Vu*, and Van-Hau Nguyen*
*Hung Yen University of Technology and Education, Hung Yen, Vietnam.
Email: {tiennm, manhtt, thehv, haunv}@utehy.edu.vn
†Le Quy Don Technical University, Hanoi, Vietnam.
Email: anhpv@lqdtu.edu.vn

***Abstract*—The admission process of universities in Vietnam is a labor-expensive task due to the involvement of humans. This paper introduces an intelligent system (a chatbot) that can support the admission process by automatically answering questions. Different from prior work that usually builds the bot from scratch, we develop the bot by using the Rasa platform. To do that, we investigate different combinations of components of natural language understanding to find the best pipeline. We also create and release a dataset in the admission domain to train the bot. Experimental results show that the pipeline using DIET with features from pre-trained language models is competitive. The introduction video of the system is also available.[1]**

***Index Terms*—Conversational AI, chatbot, Rasa.**

I. Introduction

Conversational AI (also known as virtual assistant or chatbot) constitutes an integral part of natural user interfaces [1] and has received a lot of attention from the research community [2]–[6]. The task of conversational AI is to build an AI "bot" that can naturally make interactive conversations between human an machines [1], [6], [7]. The field of conversational AI can be separated into three groups: task-oriented dialog systems, chat-oriented dialog systems, and question answering (QA) dialog systems [4], [6]. The first group performs tasks on users' perspectives such as making a reservation in a restaurant or booking a flight. The second group needs to carry out natural and interactive conversations with users based on their questions. The final group provides clear and concise answers to users' questions based on natural language processing techniques. Conversational AI bots make a big impact in e-commerce. To businesses, the bots provide an active method of connecting sellers and customers. To customers, the bots satisfy users' experience by providing online services that can work 24/7. Due to its huge benefit, the conversational AI bots have been applied to various industries such as insurance [4], education [3], entertainment [8], health care [9], e-commerce [10], COVID-19 [11], or business intelligence [12]. In practice, there are several well-known conversational AI agents such as Amazon Alexa,[2] Apple Siri,[3] Microsoft Cortana,[4] IBM Watson bot,[5] and Google assistant.

[1]https://youtu.be/gw7wvlADxnE
[2]https://www.amazon.com.au/b?node=5425666051
[3]https://www.apple.com/au/siri/
[4]https://www.microsoft.com/en-us/cortana
[5]https://www.ibm.com/watson/how-to-build-a-chatbot

In the context of admission, universities in Vietnam still use traditional methods for spreading information and calling students. To spread the admission information, one of common methods is that staff have to go to each high school to introduce the information. This task is costly, labor-expensive, and can not work in the outbreak of COVID-19 pandemic. The universities also provide another information channel on their websites; however, information on their websites is usually incomplete. A more practical way of answering admission questions is to use social platforms, e.g. Facebook, in which universities create their fan-pages and directly answer the questions by using the chat function provided by the platforms. However, we argue that this way is not so efficient due to the limited human resources, i.e. can not quickly provide correct answers 24/7. As the result, the low quality of supporting services leads to worse satisfaction of pupils and their parents.

We address the mentioned problem by developing a conversational AI chatbot to automatically answer questions of the admission process of universities in Vietnam. To do that, we first collect conversational data of admission from pupils, their parents, and staff to create a dataset. The dataset is annotated by humans to train an intelligent chatbot by using the Rasa platform. We test several pipelines to find out the best combination. After training, the bot is deployed to support the admission process. This paper makes three main contributions:

- It collects a Vietnamese dataset, including questions and answers in the domain of admission. The dataset is manually annotated by humans for training an intelligent chatbot. The dataset is also publicly accessible.[6]
- It investigates the Rasa platform for building the chatbot, which includes three main functions: intent detection, named entity recognition, and answer retrieval. The investigation allows us to test different settings of Rasa to find out the best combination of the chatbot system.
- It introduces a question answering system built on the Rasa platform. The system is deployed on Facebook to support answering the questions of the admission of Hung Yen University of Technology and Education as a case study.[7] Our system can be also adapted as a part of the admission process of any university in Vietnam.

[6]https://github.com/manhtrantienhn/UTEHY-bot-data
[7]https://www.facebook.com/chatbot.utehy.edu.vn

II. RELATED WORK

Due to the increasing impact on real life, building question answering (QA) dialog systems in conversational AI has been interested by both academia and industry [4], [6]. Although QA systems started in 1960s [13], it has become efficiently practical applications since 2015 due to the use of deep learning approaches and the availability of large-scale datasets.

One of the most practical and powerful framework for developing a chatbots is Rasa.[8] Rasa is an open-source framework which consists of two primary components, Natural Language Understanding (NLU) and dialogue management. The former manages intent classification, entity extraction, and response retrieval, while the later controls the next action in a conversation based on the context. We extend the idea of using Rasa for building chatbots [5] because Rasa provides flexible environment for creating customized pipelines.

Although conversational AI bots have been extensively studied in English [4], [8]–[12], it has few studies in Vietnamese. For example, Nguyen and Shcherbakov [14] implemented a conversational AI bot for Vietnamese based on the seq2seq network with the attention mechanism. In contrast to [14], by using knowledge base, Nguyen et al. [3] proposed a method to build an intelligent AI bot for solving mathematical problems in high school. This system is able to automatically provide tips and step-by-step instructions to high school students.

The system of Nguyen and Shcherbakov [5] is likely the most relevant work to ours. The authors designed a Vietnamese chatbot which uses two pre-trained models (e.g., FastText and BERT) in the Rasa NLU pipeline. The system obtains considerable results. While sharing the idea of building a chatbot by using Rasa, our system distinguishes [5] in two main points. First, we adapt Rasa for building an intelligent bot for the domain of admission while [5] does not mention clearly the domain of the bot. Second, we deeply investigate the ability of Rasa by defining six customized pipelines while [5] only shows three simple pipelines.

III. THE CHATBOT SYSTEM

A. Data Collection and Annotation

This section presents how data is collected and preprocessed for building our chatbot system.

1) Data collection: To build the chatbot system, we collected data from two main sources including the university's website[9] and its fan page,[10] in the last two years. We do this manually since there are not any available tools. Collected data contains documents and conversations relating to admission. After removing questions that are not answered yet, we acquired 400 conversations including 8,500 questions and 6,500 answers. To refine the data, unclear and ambiguous questions were first removed, and then regular expressions were applied to cleaning the data (e.g. removing teen codes or icons).

[8]https://rasa.com/

[9]http://www.utehy.edu.vn

[10]https://www.facebook.com/utehy.org

2) Data annotation: We annotated intents and entities for the Natural Language Understanding (NLU) component of Rasa. As recommended by Rasa, it is necessary to provide at least 10 examples for each intent to train NLU efficiently. This would be time-consuming when doing manually.

In our case, we adopt k-means to build intent examples. From the collected questions, we cluster them into a smaller number of intents. To do this, we first used TF-IDF to represent questions. We then employed the Elbow method to find out the optimized intent number (i.e. k value). Specifically, we evaluated k-means in a range of 100 to 1000 with a step of 20. As suggested in Figure 1, we divided input questions

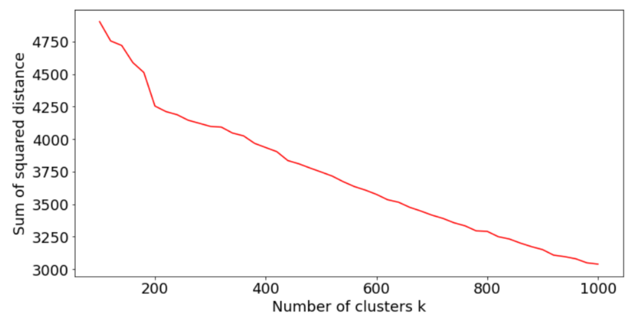

Figure 1. Preliminary experimental results with various k values.

into 200 intents. We manually revised the clustered intents by removing replicated intent examples. Furthermore, we added more examples for intents that have few instances to avoid data imbalance. It satisfies the requirement of intent examples for training NLU. Finally, we obtained a final dataset with 160 intents, in which each of them has over 10 examples. To evaluate the quality of intent annotation, we asked four Vietnamese annotators who have experience in data annotation and NLP background. Each annotator read all 4127 samples of 160 intents. If the annotator agrees with the label of each sample, then the agreement was marked as 1; otherwise is 0. The Fleiss' Kappa among four annotators is 0.67.

Apart from intent identification, Rasa NLU also requires named entity recognition (NER). This is because the policies learn to predict the next action based on a combination of both intent and entities. We, therefore, annotated entities for input data. In this task, we consider five entity types including location, major, organization, person, and time which were frequently asked by pupils and their parents. The agreement of entity annotation is 0.80 computed by using the same procedure of intent annotation on each entity type. The statistics of entity annotation is presented in Table I.

Table I
STATISTICS OF ENTITY ANNOTATION

NER	train	test	total
Location	193	41	234
Major	510	139	649
Organization	903	226	1129
Person	9	2	11
Time	102	39	141

We also annotated stories that are used for training the dialogue management model of the chatbot. This enables the bot system to generalize unseen conversation paths. To annotate the stories, we first combined one intent and one action to make 160 single-level stories. We then added more multiple-level stories with possible scenarios from real collected conversations to improve the generalization of conversation paths. The total number of annotated stories is 211.

B. The Chatbot

This section will introduce Rasa[11] to create our chatbot. We select Rasa because: (i) it is a free platform for building chatbots and (ii) it provides diverse and high-quality components for training NLU and dialog management. We first describe the overall architecture of Rasa and then introduce our investigation of Rasa components to build the chatbot.

1) The Rasa platform: The interaction process between a Rasa assistant agent and human includes steps as follows:

- Receiving the message and passing to an interpreter. The interpreter analyzes the message to several data forms including the original text, intent, and entities.
- The message is passed to the tracker, which keeps tracking conversation states, to obtain the current state.
- The current state is sent to the policy. Wherein, the machine learning and rule-based policies are used to decide the action at each step in a conversation.
- The policy with the highest confidence score is selected to execute the next action.
- The chosen action is logged by the tracker.
- A response is sent to the user.

The above process is implemented by two Rasa modules including Rasa NLU (Natural Language Understanding) and Rasa Core. Rasa NLU interprets the query to classify into pre-defined intents and extract entities. Rasa Core uses a probabilistic model to select the chatbot action based on the data analyzed by Rasa NLU.

2) Natural language understanding: The important task of the NLU module is to understand the user intention and convert unstructured data to structured data. For example, the question "`I live in Tan Dan, how can I obtain the university offer letter`" has the intent of `ask_offer_letter`, entities of `location` and `document` with values of `Tan Dan` and `university offer letter`. Thus, the chatbot must recommend the nearest office to receive the letter or inform the letter status if it is not available. In this work, we investigate the DIET model [15] to implement NLU module. Figure 2 shows the pipeline of the NLU including stages of tokenization, featurization, named entity recognition, and intent classification.

3) Dialog management: The dialog management uses three main policies to decide the next action. The three policies are:

- RulePolicy makes predictions based on rules defined in the "rules.yml" file if the conversation is matched with predefined rules.

[11]https://rasa.com

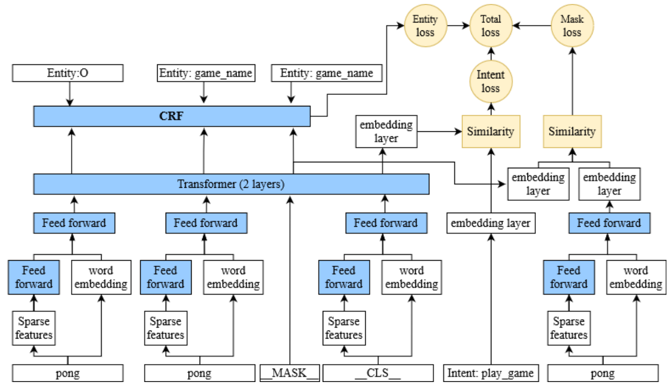

Figure 2. The DIET architecture for joint intent classification and NER [15].

- MemoizationPolicy remembers the stories from the training data and predict the next action from the matching stories in the "stories.yml" file.
- TEDPolicy uses machine learning to predict the next best action based on its probability.

Each policy will make the prediction with a confidence score, and the action with the highest priority is chosen to prepare the response to the user.

C. The Customized Pipelines of the System

We customized the pipelines based on the components of NLU. The purpose is to find out an appropriate combination to build the bot. We defined six pipelines as in Table II. Since these pipelines share the classifier and entity detection by using DIET, and dialog management, so Table II only shows the main difference in terms of tokenizer and featurizer.

The first pipeline uses VNCoreNLP [16] for tokenization and uses regular expression and counting vectors as features. The second pipeline replaces the tokenization of the first pipeline by using white space tokenization. The third pipeline uses white space for tokenization and BERT for embedding (note that this is the BERT multilingual adapted for Rasa).[12] The fourth pipeline uses VNCoreNLP for tokenization and the original BERT multilingual for embedding. The fifth pipeline enriches the hidden representation by combining features of PhoBERT, regular expression, and counting vectors. The final pipeline only changes PhoBERT by BERT (adapted for Rasa).

IV. Settings and Evaluation Metrics

A. Settings

The dataset was divided into training and testing sets with a ratio of 80-20. The counting feature (term frequency) was used on the character level with 4-grams. The DIET classifier uses two transformer layers with relative position attention. The vector size is 768 and it was trained in 50 epochs. The respond selector was trained in 100 epochs. The pipelines use the fallback method with a threshold of 0.5. All pipelines were trained by using a single GeForce RTX 2080 Ti GPU.

[12]Note that the weight of BERT was adapted for Rasa. The detailed guideline can be found at https://huggingface.co/rasa/LaBSE.

Table II
THE SIX CUSTOMIZED PIPELINES OF THE SYSTEM.

Pipline	Component	Setting
1	Tokenizer	VNCoreNLP
	Featurizer	Regex; CountVectors
2	Tokenizer	Whitespace
	Featurizer	Regex; CountVectors
3	Tokenizer	Whitespace
	Featurizer	BERT (weight from Rasa)
4	Tokenizer	VNCoreNLP
	Featurizer	BERT (original multi-lingual)
5	Tokenizer	VNCoreNLP
	Featurizer	PhoBERT; Regex; CountVectors
6	Tokenizer	VNCoreNLP
	Featurizer	BERT (weights from Rasa); Regex; CountVectors

B. Evaluation Metrics

We used precision, recall, and F-score as the main metric to report our experiments of intent detection and NER.

$$Precision = \frac{TP}{TP + FP}; \quad Recall = \frac{TP}{TP + FN} \tag{1}$$

$$F - score = \frac{2 * P * R}{P + R} \tag{2}$$

where:

- TP is the number of correct predicted samples.
- FP is the number of samples that the model predicts correctly, but their labels are incorrect.
- FN is the number of samples that the model predict incorrectly, but their labels are correct.

V. RESULTS AND DISCUSSION

This section first shows the comparison of six pipelines and the accuracy of dialog management. It next reports the ablation study and finally shows the system.

A. Preliminary Results

1) NLU results:

a) Results of intent classification: We report the results of intent detection by using six pipelines. To ensure a fair comparison, the pipelines were trained by using 5-fold cross-validation on the training set. After that, they were re-trained again on the whole training set to predict the test set.

Table III
THE PERFORMANCE OF PIPELINES FOR INTENT DETECTION.

Pipeline	Train set			Test set		
	Precision	Recall	F-score	Precision	Recall	F-score
1	96.5	96.3	96.0	85.9	85.4	84.5
2	96.6	96.4	96.2	86.1	85.6	84.7
3	92.3	92.1	91.1	81.1	81.5	79.6
4	92.6	92.7	91.7	81.4	81.1	79.2
5	97.5	97.4	97.2	86.2	85.9	85.0
6	**98.6**	**98.4**	**98.5**	**89.0**	**86.8**	**87.9**

As observed from Table III, the sixth pipeline consistently outputs better results than others. There are two possible reasons. First, it combines multiple feature types (BERT, regex features, count vectorized features) to enrich the representation of intents. Second, it uses a Vietnamese tokenizer. As the result, this pipeline obtains promising results. There are tiny margins between the best (sixth) and the second-best pipeline (fifth). This is because they share the same setting, except for the language model. It also indicates that PhoBERT can be applicable for intent detection. Interestingly, pipelines using quite simple features (first and second) also achieve promising results. Results of third and fourth pipelines using features from BERT (the original BERT multilingual and BERT with the weights adapted for Rasa) do not show better results than the first and second pipelines. It confirms the contribution of n-grams features for the task of intent classification.

b) Entity extraction: We investigated the efficiency of six pipelines for the task of NER.[13] The setting is similar to the intent detection, in which these pipelines were trained by using 5-fold cross-validation of the training set and then re-trained to predict the test set.

Table IV
THE PERFORMANCE OF PIPELINES FOR NER.

Pipeline	Train set			Test set		
	Precision	Recall	F-score	Precision	Recall	F-score
1	96.7	96.0	96.4	**93.7**	**93.0**	**93.3**
2	96.6	95.1	95.5	93.1	91.6	92.2
3	96.7	95.4	95.5	90.9	90.7	90.7
4	96.8	96.1	96.4	92.2	92.2	92.2
5	96.8	**96.2**	**96.5**	93.3	92.8	93.0
6	**97.1**	96.1	96.4	92.6	92.4	92.5

Table IV shows that the sixth pipeline is not the best. On the training set, the pipeline using PhoBERT outputs promising results. On the testing set, the first pipeline using traditional features shows the efficiency. The possible reason is that the number of training samples of NER is small (Table I), but DIET (a deep learning method) requires more training samples. However, the margins among pipelines are small.

c) The contribution of pre-trained LMs: We observed the contribution of pre-trained language models (preLMs). We changed the language model of the featurizer of the best pipeline (the sixth pipeline). The pipeline was re-trained to compute the F-scores on training and testing sets. Due to space limitation, we only report the comparison of intent detection.

Table V
THE CONTRIBUTION OF PRE-TRAINED LMS.

Pre-LM	Train set			Test set		
	Precision	Recall	F-score	Precision	Recall	F-score
XLNet [17]	97.3	96.3	96.8	84.6	80.8	82.7
GPT-2 [18]	93.4	91.6	92.5	82.8	79.6	81.2
PhoBERT [19]	97.5	97.4	97.2	86.2	85.9	85.0
RoBERTa [20]	97.0	96.4	96.7	86.0	85.2	85.6
DistilBERT [21]	96.8	96.0	96.4	86.7	83.9	85.3
BERT [22]	**98.6**	**98.4**	**98.5**	**89.0**	**86.8**	**87.9**

Results in Table V indicate that the pipeline using BERT[14] acquires competitive results, followed by the pipeline using

[13]Due to the imbalance of entities, we used the weighted average F-scores.
[14]The weight of BERT was adapted for Rasa, not the common BERT multilingual, https://huggingface.co/rasa/LaBSE.

PhoBERT. It is because the weight of BERT adapted for Rasa is more appropriate than other pre-trained LMs. PhoBERT also shows its efficiency because it was adapted for Vietnamese. Other pre-trained LMs do not show the best performance due to their limitation on a low-resource language.

d) Training data and accuracy: We studied the relationship between the number of training samples and the performance of the model for intent classification. To do that, training data were randomly split into different percentages, in the range of [10, 25, 50, 75, 100]. The model was re-trained on such data to make the prediction on the test set.

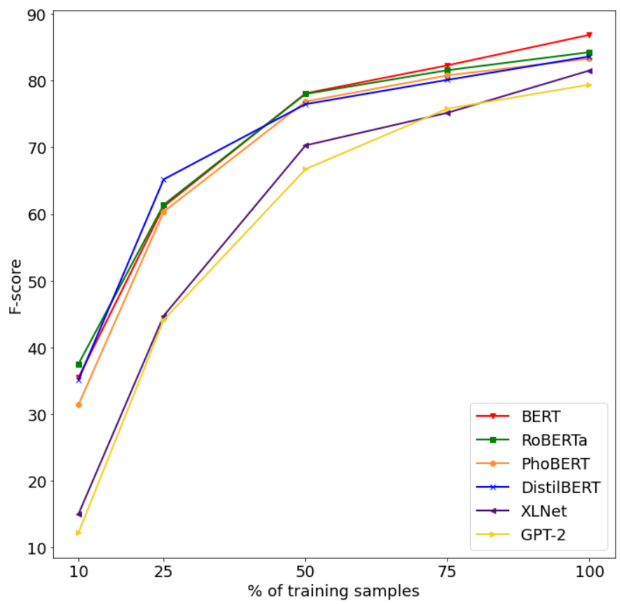

Figure 3. The performance of NLU with different training data sizes.

Figure 3 shows that the number of training samples affects the performance of NLU. The F-scores are low at 10% of training data and significantly increase from 10% to 50%. After that, the increased margins are small. We also observe that the trend of BERT, PhoBERT, RoBERTa, and DistilBERT is close. The performance of the pipeline using these pre-trained LMs is also better than pipelines using GPT-2 and XLNet. This trend supports the results in Tables V and IV.

2) Dialog management: We show the performance of conversation and action prediction. Due to the guideline of Rasa, the conversation performance is reported by accuracy.

Table VI
THE PERFORMANCE OF CONVERSATION AND ACTION PREDICTION.

Conversation	Action		
Accuracy	Precision	Recall	F-score
51.2	81.6	83.1	82.1

Table VI shows that generating conversation is challenging for the bot. This is because the bot was only trained with 211 stories. Also, generating an appropriate dialog is one of the most challenging tasks of building a chatbot. We argue that the quality of conversation generation will be improved by adding more story samples. In contrast, the performance of action prediction is acceptable because this prediction relates to intent detection, which achieves high F-scores in Table III.

B. Ablation Study

We investigated the contribution of each component in the sixth pipeline. To do that, we removed each component and re-trained the pipeline to predict the test set. Because DIET jointly predicts intents and entities, we replaced DIET by using SVM (linear kernel, $C = 1.0$, $gamma = 0.1$) and CRF ($L_1 = 0.1$ and $L_2 = 0.1$ with default features) for intent and entity detection. The F-scores were computed on the test set.

Table VII
THE ABLATION STUDY FOR INTENT DETECTION.

Tokenizer	✓	✓	✓	✓
Featurizer (LM)		✓	✓	✓
Classifier (SVM+CRF)			✓	✓
Entity mapper				✓
F-score	84.1	88.8	87.7	87.9

Table VII shows that each component contributes to the performance of the pipeline. By removing LM features and entity mapper, the F-scores reduce. For example, when removing the LM features, the F-score reduces around 3%, from 87.7 to 84.1. It shows that the LM features play an important role to train the NLU model. Interestingly, changing DIET by SVM and CRF slightly increases the F-scores. As mentioned, the training data is quite limited while DIET requires a large number of training samples. However, the gap is small, showing that we can use DIET or SVM and CRF.

C. The System

The system was deployed on Facebook to support the admission of Hung Yen University of Technology and Education.[15] Figure 4 shows an output of a dialog between a pupil and the bot. As we can observe that the bot can understand questions from the user and correctly answer the questions. For example, the bot can understand the second question of asking the number of admitted students for 2022. By analyzing the question with the NLU, the bot can retrieve a correct answer: the number of admitted students is 2,860 in three admission methods. The bot can capture the sequence of the dialog by understanding the third question and giving the correct answer.

However, in some cases, input questions make challenges for the bot. For example, in the dialog of asking the time of admission, one question is "`Em tới muộn có được không/ (Can I come lately after 8:00 AM?)`". The bot answers the question by giving the information of the 2+2 program collaborated with the SonMoon University. In this case, the bot did not understand the intent of the question, which might not in the training data of intent detection and NER. Therefore, it can not predict correctly the next action and the answer. This supports the result of conversation generation

[15]https://www.facebook.com/chatbot.utehy.edu.vn. This link needs the authority from Facebook to obtain the key for connection. Please contact the authors for this information.

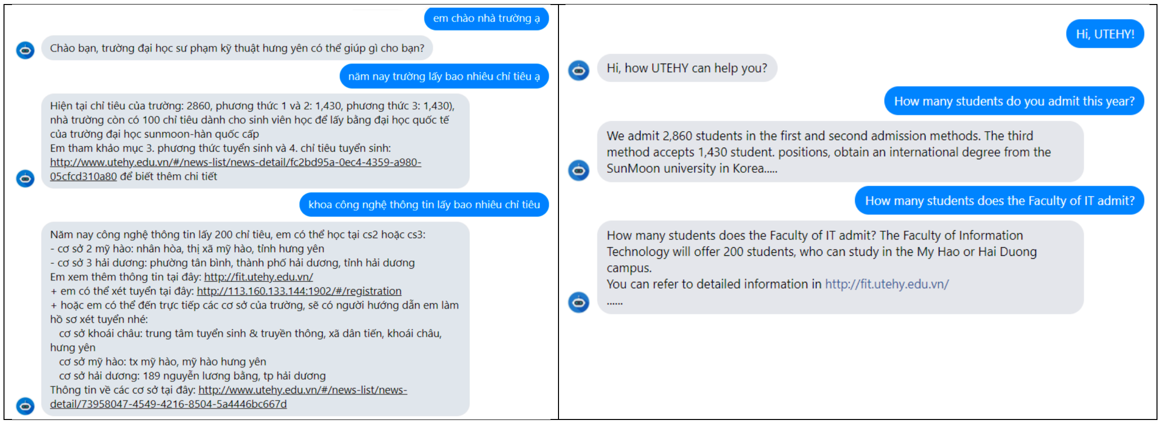

Figure 4. The output from the system. The text in the right was translated from Vietnamese in the left.

in Table VI. We argue that this problem can be addressed by adding more training samples for NLU.

VI. Conclusion

This paper introduces an intelligent chatbot that supports the admission process of universities. We built the bot based on the Rasa platform, in which different pipelines are investigated to find out the best combination of NLU components. Experimental results show three important points. First, the pipeline using DIET with language features shows competitive performance for NLU. Second, there are small margins among pipelines. Finally, SVM and CRF can output better results of DIET for NLU due to the small number of training data. We deploy the bot to support the admission process of Hung Yen University of Technology and Education as a case study.

The future work will refine the dataset (i.e. intents, entities, stories). We also would like to improve the quality of NLU and dialog management to generate more correct answers.

Acknowledgment

This research is funded by Hung Yen University of Technology and Education under the grant number UTEHY.L.2021.08.

References

[1] J. Gao, M. Galley, and L. Li, "Neural approaches to conversational ai," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 1371-1374*, 2018.

[2] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gasic, L. M. R. Barahona, P.-H. Su, S. Ultes, and S. Young, "A network-based end-to-end trainable task-oriented dialogue system," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 438-449*, 2017.

[3] H. D. Nguyen, V. T. Pham, D. A. Tran, and T. T. Le, "Intelligent tutoring chatbot for solving mathematical problems in high-school," in *2019 11th International Conference on Knowledge and Systems Engineering (KSE), pp. 1-6. IEEE*, 2019.

[4] M. Nuruzzaman and O. K. Hussain, "Intellibot: A dialogue-based chatbot for the insurance industry," *Knowledge-Based Systems 196: 105810*, 2020.

[5] T. Nguyen and M. Shcherbakov, "Enhancing rasa nlu model for vietnamese chatbot," *International Journal of Open Information Technologies 9, no. 1, pp. 31-36*, 2021.

[6] M. Zaib, W. E. Zhang, Q. Z. Sheng, A. Mahmood, and Y. Zhang, "Conversational question answering: A survey," in *arXiv preprint arXiv:2106.00874*, 2021.

[7] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Advances in neural information processing systems 28: 1693-1701*, 2015.

[8] H. Su, X. Shen, Z. Xiao, Z. Zhang, E. Chang, C. Zhang, C. Niu, and J. Zhou, "Moviechats: Chat like humans in a closed domain," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6605-6619*, 2020.

[9] M. Bates, "Health care chatbots are here to help," in *IEEE pulse 10, no. 3: 12-14*, 2019.

[10] A. Følstad, P. B. Brandtzaeg, T. Feltwell, E. L. Law, M. Tscheligi, and E. A. Luger, "Sig: chatbots for social good," in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1-4*, 2018.

[11] A. S. Miner, L. Laranjo, and A. B. Kocaballi, "Chatbots in the fight against the covid-19 pandemic," in *NPJ digital medicine 3, no. 1 : 1-4*, 2020.

[12] A. Quamar, F. Özcan, D. Miller, R. J. Moore, R. Niehus, and J. Kreulen, "Conversational bi: an ontology-driven conversation system for business intelligence applications," in *Proceedings of the VLDB Endowment 13, no. 12: 3369-3381*, 2020.

[13] C. Monz, "Machine learning for query formulation in question answering," *Natural Language Engineering*, vol. 17, pp. 425 – 454, 10 2011.

[14] T. Nguyen and M. Shcherbakov, "A neural network based vietnamese chatbot," in *2018 International Conference on System Modeling Advancement in Research Trends (SMART)*, 2018, pp. 147–149.

[15] T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol, "Diet: Lightweight language understanding for dialogue systems," in *arXiv preprint arXiv:2004.09936*, 2020.

[16] T. Vu, D. Q. Nguyen, M. Dras, and M. Johnson, "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, pp. 56-60*, 2018.

[17] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems, pp. 5754-5764*, 2019.

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog 1, no. 8: 9*, 2019.

[19] D. Q. Nguyen and A. T. Nguyen, "Phobert: Pre-trained language models for vietnamese," in *rXiv preprint arXiv:2003.00744*, 2020.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," in *arXiv:1907.11692*, 2019.

[21] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter," in *arXiv preprint arXiv:1910.01108*, 2019.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.