International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An Integrated DIET-BO Model for Intent Classification and Entity Extraction

Viet Minh Nhat Vo[1], Van Son Ngo*[2]

Submitted: 09/05/2023 **Revised**: 14/07/2023 **Accepted**: 06/08/2023

Abstract: The DIET (Dual Intent and Entity Transformer) architecture is known as an effective method of intent classification and entity extraction for chatbot systems. However, a challenge is how to determine the best set of hyperparameters in terms of the number of iterations, the number of transformer layers, the transformer size, etc. to achieve the best DIET architecture. With huge possible combinations of hyperparameter values, there are an explosive number of DIET architectures to be considered. One solution to this problem is to integrate a statistical analysis technique such as Bayesian Optimization (BO) into the process of determining the best DIET architecture. The article proposes an integrated DIET-BO model, in which each DIET architecture is a candidate solution in the search space, the DIET training process is considered as an objective function and BO is used to find the best DIET architecture in the space of candidate solutions. A hotel chatbot conversational dataset is used to evaluate the effectiveness of the integrated DIET-BO model. The experimental results show that the integrated DIET-BO model achieves the intent classification F1-score of 0.869 and the entity extraction F1-score of 0.913.

Keywords: *Chatbot, DIET, natural language processing, machine learning, pre-training*

1. Introduction

DIET is a new technology integrated into the RASA framework [1] to assist users in building chatbot systems. DIET can automatically classify intents and extract entities from user conversations. DIET is an architecture based on Transformer [2], which provides connectivity with pre-trained models such as BERT [3], GPT-3 [4], XLM [5], GloVe [6], RoBERTa [7], ConveRT [8], and XLNet [9]. According to Tanja Bunk et al. [10], an integrated DIET architecture is suitable for the chatbot development process, compatible with large pre-trained language models, and reduces training time. However, to improve efficiency, it is necessary to solve the problem of finding the most appropriate collection of hyperparameter values when training the DIET model. In practice, it is necessary to continuously adjust the hyperparameters and train the obtained DIET architecture to find the best combination. Therefore, how to find the best set of hyperparameter value is the most important issue.

There are two main approaches to find the most appropriate collection of hyperparameter values: Manual Search and Automatic Search. Manual search starts with trying different hyperparameter values, depending on expert intuition and experience, in order to identify the most appropriate collection of hyperparameter values [11]. This approach then determines the relationship between hyperparameter values and the final result displayed on a visual tool. Performing manual searches is challenging for individuals without a professional background and practical experience, posing difficulties for non-specialists. Furthermore, as the number of hyperparameters and the value range for each hyperparameter increase, the management becomes more challenging due to the complexity of processing multidimensional data and the risk of misunderstanding or overlooking trends and relationships among the hyperparameters.

In order to overcome the limitations of manual search, researchers have developed automated search methods, including grid search and random search. These approaches aim to streamline the search process and improve efficiency [11]. The principle of grid search is to search the entire search space. For each set of hyperparameter values found, it trains the corresponding DIET architecture with a machine learning model, which is considered the objective function. The training results are then evaluated to find the most appropriate collection of hyperparameter values. While this method allows for automated adjustment and has the potential to achieve the objective function's global best value, it comes with a significant trade-off in terms of execution time. As the number of hyperparameters and their corresponding value ranges increases, the effectiveness of the search decreases rapidly [12].

In order to mitigate the execution time problem encountered in grid search, a solution known as random search is introduced. This method selectively considers a subset of crucial hyperparameters, aiming to streamline the search process. The execution time is thus greatly improved, but

[1] Hue University, Hue - 530000, Viet Nam
ORCID ID : 0000-0002-0686-5529
[2] School of Hospitality and Tourism – Hue University, Hue - 530000, Viet Nam
ORCID ID : 0009-0007-3707-0669
** Corresponding Author Email: ngovanson@hueuni.edu.vn*

the obtained results are often only near-optimal. Random search offers improved efficiency compared to grid search when dealing with multidimensional data. However, its reliability decreases when dealing with complex training models. Therefore, how to automatically adjust to achieve high accuracy and high effectiveness is always an unsolved problem in machine learning.

Finding the optimal set of hyperparameter values for the DIET architecture poses an optimization challenge, as the objective function is often challenging to define. This characteristic has led to the metaphorical description of this problem as a "black box". Traditional optimization methods such as gradient descent are not suitable for this class of problems. Therefore, Bayesian optimization can be chosen because it combines the previous results with the current solution to obtain a next prediction. Based on predictable results, the position at which the function reaches the optimal value can be inferred. Experimental results of previous studies show that BO is often superior to other optimization methods [13],[14]. In this article, Bayesian optimization based on the Gaussian process [15] is used to determine the optimal set of hyperparameter values for the DIET architecture.

The main contributions of the paper include:

- Modeling the problem of intent classification and entity extraction into the problem of finding the optimal DIET architecture based on BO; an integrated DIET-BO model is then proposed;

- Analyzing, normalizing and converting hotel chatbot conversation data into input datasets for the integrated DIET-BO model; and

- Deploying and evaluating the effectiveness of the DIET-BO integrated model with the hotel chatbot conversation datasets.

The following contents of the article include: Section 2 summarizes and evaluates related studies in the past 5 years, which focus on the DIET architectures for chatbot. On the basis of the analysis, Section 3 describes in detail the integrated DIET-BO model. Experimental implementation and results analysis are presented in Section 4. Finally, the conclusion is presented in Section 5.

2. Related Works

To achieve high effectiveness in natural language processing, chatbot systems often use machine learning models in their predictive architecture. DIET is an architecture widely used in natural language understanding (NLU) systems. However, determining the optimal set of hyperparameter values for the DIET architecture is often a difficult task, requiring a deep understanding of the data and the training process. The article focuses on evaluating the DIET architecture used in chatbot systems with natural language understanding in the last 5 years.

The research of Astiti et al. [16] evaluates the performance of a chatbot system in natural language processing to answer questions about the COVID-19 epidemic. To build the chatbot system, the RASA framework and DIET architecture with 300 trained data samples are used. Experimental results on rasa.core.test and rasa.nlu.test show that DIET achieves about 85% accuracy of correct answers.

A chatbot was created in education to make it simpler to give better and more easily accessible information services. L. Fauzia et al. [17] proposed a chatbot design that is built with the Rasa framework and is based on the DIET with default hyperparameters, with the goal of answering inquiries concerning new student admissions. DIET was utilized in research [18] to create an intelligent system that can assist the admissions process by automatically answering questions. The experimental results reveal that the DIET pipeline chooses hyperparameter settings based on experience. Similarly, Vidhish Panchal et al. [19] offered 100 epochs in hyperparameter settings for a solution that is a voice-enabled and multilingual chatbot in Rasa and DIET. It can help both visually impaired persons and typical students learn by simply interacting with the speech bot.

In the research of Arevalillo-Herráez et al. [20], DIET is used to solve the sentiment analysis problem. According to the research, DIET can be used efficiently and seamlessly for NLU-related text classification, such as sentiment analysis. To evaluate the effectiveness, three movie review datasets, including Internet Movie Database (IMDb), Movie Review (MR) and Stanford Sentiment Treebank (SST2), are tested. The best result is obtained when DIET uses a pre-trained language model, surpassing other recent proposals on sentiment analysis. The accuracy rates of IMDb, MR, and SST2 are 0.907, 0.816, and 0.858, respectively. However, the research solely examines the default set of hyperparameter values without investigating alternative method.

Shen et al. [14] proposes the combination of Bidirectional Encoder Representations from Transformers (BERT) and Random Forests (RF), called the BERT-RF model, to classify sentiments based on a social media dataset. This proposal overcomes the disadvantages of social media texts that are normally characterized by features such as short, colloquial, difficult to extract specific information, and thus affect the accuracy of sentiment classification. The BERT-RF integration model uses BERT to derive feature extraction from the textual content on social networks and uses random forests as the classifier based on the features generated by BERT. To improve accuracy, this model has optimized the hyperparameters in the random forest model using the Bayesian algorithm. The research results show that the BERT-RF model has significantly increased sentiment

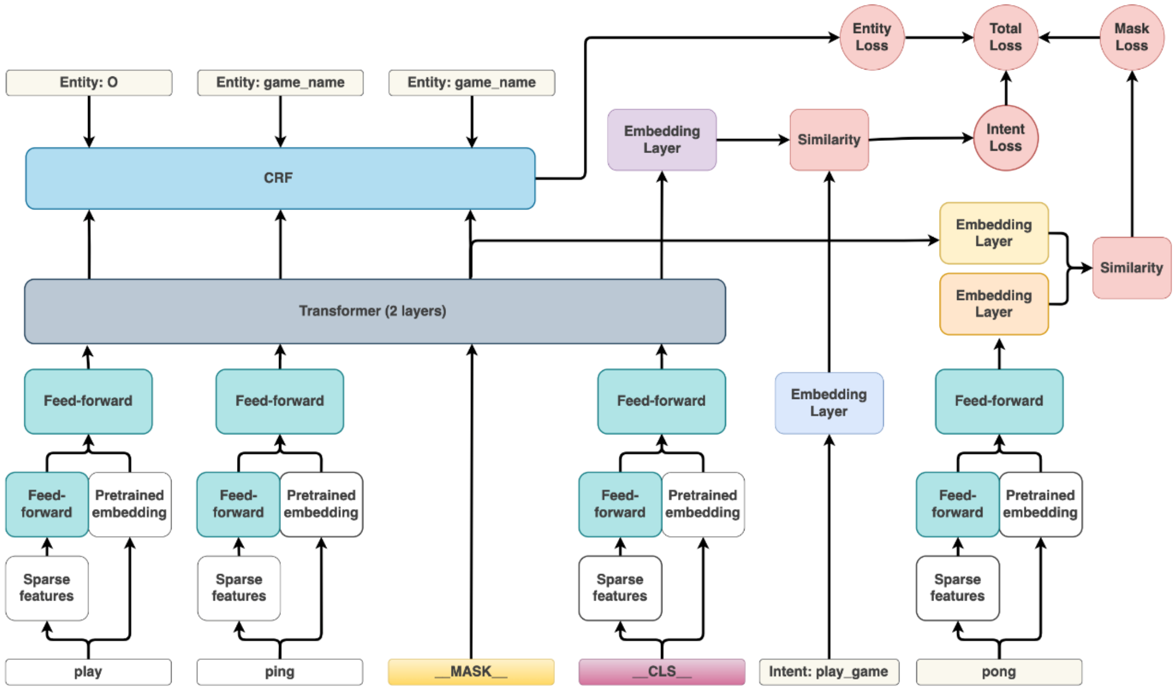

Fig 1. The DIET architecture [16]

classification accuracy. However, this study does not compare Bayesian optimization with other methods such as grid search, random search with two hyperparameters of *n_estimators* and *max_depth* that are used to optimize for random forests.

In summary, the above studies try to use the DIET architecture to increase the effectiveness of natural language processing. Depending on the specific case, each of the above studies shows how DIET is applied to each specific problem of intent classification, entity extraction in chatbots, and sentiment analysis. However, most of the above studies only use the default set of hyperparameter values and have not searched for the optimal set of hyperparameters. This article uses BO to find the most appropriate collection of hyperparameter values of the DIET architecture used in a chatbot system.

3. Integrated Diet-Bo Model

3.1. DIET: the architecture of intent classification and entity extraction

Large-scale pre-trained language models are not suitable for developing chatbot systems, because they tend to use a lot of computational resources, take a long time to train, and create many practical challenges when developing chatbots. Furthermore, when building a multilingual chatbot, it is important to achieve high performance without large-scale pre-training, while most pre-training models are done with English text.

To overcome this drawback, Tanja Bunk et al. [10] proposed DIET, an architecture based on the Transformer Neural Network with a centralized mechanism to handle both intent classification and entity extraction (Figure 1). DIET is built on a modular architecture that fits the software development process and improves training speed 6 times faster than the refined BERT model. DIET also provides integration with other pre-training models like BERT, GloVe, ConveRT, XLNet, etc. DIET is a flexible architecture for intent classification and entity extraction. Research results show that there is no single best embedding set for all datasets. This highlights the importance of a modular DIET architecture integrated into the RASA framework.

The DIET's mechanism of action is similar to Transformer [2] but has been redesigned to take into account word order, is more compact and offers better performance. DIET can be used to conduct both intent classification and entity extraction, or it can execute only one of these tasks.

3.2. Bayesian optimization

In building a machine learning model applied to an intelligent chatbot system, the adjustment of hyperparameter values is to optimize the system. This is the process of finding a combination of input values for a machine learning model to achieve the best performance for a specified problem. Hyperparameters affect the accuracy of the model, so the adjustment of hyperparameter values should be done reasonably. To find a set of hyperparameter values which satisfy some constraints during optimization, various approaches can be used.

- Manual: it is the way of selecting hyperparameter values based on experience, guessing before training the model; then repeats the selection process and train the model until

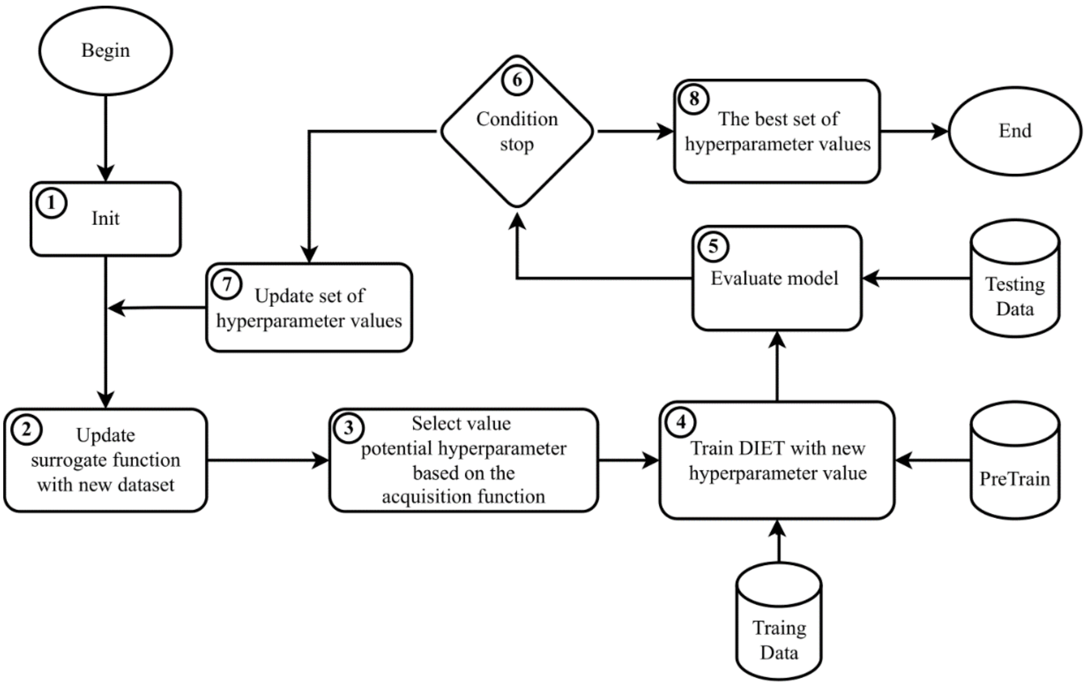

Fig 2. Implementation and analysis of results.

the most appropriate collection of hyperparameter values is found.

- Grid search: it is the method to find the most appropriate collection of hyperparameter values by creating a grid of values and in turn training the model according to each of those values to find the most appropriate collection of hyperparameter values.

- Random search: it is the approach of randomly selecting hyperparameter values for training the model and repeating this process for a limited number of iterations until the most appropriate collection of hyperparameter values is found.

- Automatic search: it is the use of some automatic methods to select the most appropriate collection of hyperparameter values.

For some problems, such as finding the best DIET architecture, the objective function f(x) is too complex to represent explicitly or cannot be analyzed, so f(x) can be considered as a black box. Applying grid search or random search may encounter the problem of unusable previous hyperparameter values. As for the manual method, the choice of hyperparameter values depends too much on the experience of the trainer. Therefore, BO is suitable for automatic search of hyperparameter values, reducing the number of evaluations per set of hyperparameter values, but consuming considerable time per evaluation. In fact, this helps BO jump out of local optimizations.

BO consists of two parts: the surrogate model and the acquisition function [21]. The surrogate model has the role of storing, updating, and extracting features of the relationship between hyperparameter values and corresponding training results. Based on the features provided by the surrogate model, the acquisition function performs the optimal value calculation. BO works primarily by constructing a surrogate model of f(x), which has a probability distribution of points that represents the properties of f(x). Using the surrogate function costs less than the objective function. So BO chooses the next values by optimizing the surrogate function based on past evaluation results [22]. The idea here is that, when the data is large enough, the surrogate function is asymptotic to the objective function, and the best found hyperparameters of the surrogate function is also the best for the objective function. Some popular choices used as surrogate functions are Gaussian process [15], Random Forest [23] and Tree Parzen Estimator [24].

Since we don't know anything about the optimal function, besides getting the results from the training, there are two factors that need to be fine-tuned:

- Exploration: prioritizes the selection of the most uncertain points, to help jump out the local optimal position during the search, but that is often time consuming.

- Exploitation: prioritizes the points near the optimal region based on the current optimal position by exploiting existing information.

Instead of falling into exploitation vs. exploration tradeoff, we can mix the two into one. The core idea of BO to solve this problem is to build an acquisition function that considers which option is the next best fit. The values contained in the representation model are used to create the acquisition function a(x). The next point x_t is determined by optimizing the function a(x), which is a function used to find out parameters in the BO process. It uses the mean and the predictive variance generated by the surrogate function. The effectiveness of the function f(x) is evaluated using the updated parameters x_t. The process goes on until the best parameters are reached. The Bayesian optimization algorithm [25], [26] is described as follows:

Algorithm 1 Pseudocode for Bayesian optimization algorithm

for t **do**=1,2,... do

update *surrogate model* on the observed dataset D_{t-1}

select new data point x_t by optimizing $a(x)$: $x_t = argmaxa(x)$

query $y_t = f(x_t)$

update dataset $D_t = D_{t-1} U(x_i, y_t)$

The components of BO include:

- Objective Function: takes hyperparameters and returns the measure to be minimized or maximized.

- Domain space: the value range of hyperparameters to calculate.

- Optimization algorithm: the way of employing the surrogate function and picking the model's next value.

- Results: measure and value pairs for the algorithm to build the surrogate function.

In this article, we propose to use the Gaussian Process for the approximation model and the expected improvement for the acquisition function [27].

3.3. Integration of DIET and BO

Searching hyperparameter values by grid or random search has limitations such as long training time and unexpected convergence. To overcome the drawbacks, the integrated DIET-BO model helps to find the most appropriate collection of hyperparameter values. BO uses an estimator to estimate the distance between the hyperparameter values and the actual result. Instead of deciding the model's hyperparameter values on their own, DIET-BO uses BO to find the optimal hyperparameter values for the DIET architecture. In the proposed DIET-BO model, the DIET architecture evaluation process is the objective function in the search space of BO. The integrated DIET-BO algorithm is implemented through 8 steps as shown in Figure 2.

Step 1: Initialize the search space of hyperparameter value sets. The initialization needs to be well defined so that the value of each hyperparameter is used correctly.

Step 1: Update the surrogate function with new data using the Gaussian normal distribution. Initially the data is randomly generated. After each loop, the hyperparameter values are updated and used for the surrogate function.

Step 3: Select the potential hyperparameter values based on the acquisition function. The acquisition function is used to weigh the choice between discovery and exploitation.

Step 4: Train the DIET architecture based on the set of hyperparameter values selected in Step 3.

Step 5: Evaluate the effectiveness based on test data with a new set of hyperparameter values.

Step 6: Check the stopping condition, which can be a limited number of loops or a predefined threshold.

Step 7: Update hyperparameter values and results of each loop through Bayesian cumulative calculation to create a more stable model.

Step 8: The returned result is the most appropriate collection of hyperparameter values.

In addition, the DIET-BO model has the flexibility to use new data or data from the pre-training model in combination with application domain data.

4. Implementation and Analysis of Results

We use RASA 3.0 to implement the integrated DIET-BO model, with the dataset collected from hotel conversation sites. The data was then edited according to the structure of RASA with 19 intents, 24 entities and more than 700 sentence patterns. The Python 3.8 language and related libraries are used to optimize hyperparameter values based on Bayesian optimization.

To avoid overfitting and underfitting when building the model, we used the Leave-One-Out technique (a case of k-fold cross-validation) to organize the training and testing dataset during training and evaluating.

To evaluate training models, the commonly used criterion is F1 score [14], [16]. In our implementation, the integrated DIET-BO model is compared to grid search and random search.

With the criterion for optimal evaluation being the F1 score of intent classification (intent f1-score) and entity extraction (entity f1-score), the hyperparameters used in implementation and search space are set up as follows:

embedding_dimension = (1, 1000)

epochs = (16, 1024)$

number_of_transformer_layers = (0, 8$

transformer_size = ['4', '16', '64', '256', '1024']

We perform 20 different sets of hyperparameter values for grid search, random search and DIET-BO integrated model. The results are shown in Figure 3, where each point is a corresponding optimal result: diamond points represent DIET-BO's results, square points represent random search's results, and triangle points represent grid search's results.

Table 1. The obtained optimal set of hyperparameter values of the integrated DIET-BO model.

Hyperparameter	*Value*	*F1 score on intent classification*	*F1 score on entity extraction*
embedding_dimension	905		
epochs	356	0.8693769998117824	0.9130434782608695
number_of_transformer_layers	8		
transformer_size	256		

Figure 3 shows that grid search's F1 scores have a low distribution: intent classification below 0.1 and entity extraction below 0.4. For the random search, the distribution of F1 scores on intent classification and entity extraction is more scattered. Especially with the DIET-BO model, F1 scores of both intent classification and entity extraction tend to distribute towards high. Also as described in Figure 3, with the DIET-BO model, initially the distribution points are low but gradually their positions move to higher. This represents an improvement in the effectiveness of the DIET-BO model by learning from previous hyperparameter values. The learning results in the higher distribution of F1 scores on both intent classification and entity extraction of the integrated DIET-BO model.

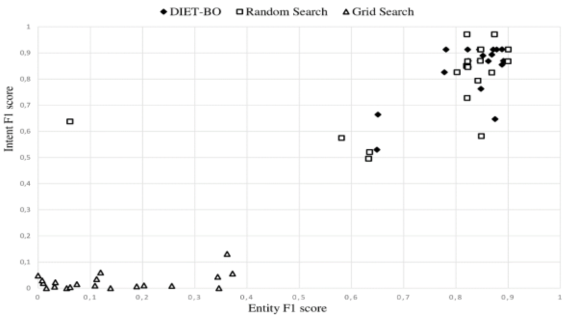

Figure 4. A comparison of F1 score on intent classification and entity extraction over 20 runtimes.

The obtained result of the integrated DIET-BO model is the optimal set of hyperparameter values as shown in Table 4.

The results are also shown through a Parallel Coordinates Plot (Figure 4), which analyzes information from the set of hyperparameter values compared to the optimization criteria thanks to the ability to display the criteria and difference between hyperparameters.

The Parallel Coordinates Plot in Figure 4 is shown over 20 runtimes to find the most appropriate collection of hyperparameter values. It shows the relationship between hyperparameters and criteria to be optimized (F1-scores of intent classification and entity extraction). Each connection line between the positions helps us to determine the optimal criterion compared to the selected hyperparameter values. In addition, the Parallel Coordinates Plot helps us to have an intuitive comparison between the optimal values and the changes of hyperparameters. Based on the Parallel Coordinates Plot, we see that the parameter *transform_size* has a major impact on the effectiveness of the model.

The experimental results also help to determine the role of hyperparameters, i.e. their impact on the evaluation score. As shown in Figure 5, *transform_size* has the greatest impact on both intent classification and entity extraction. Meanwhile, epochs had a second effect on intent

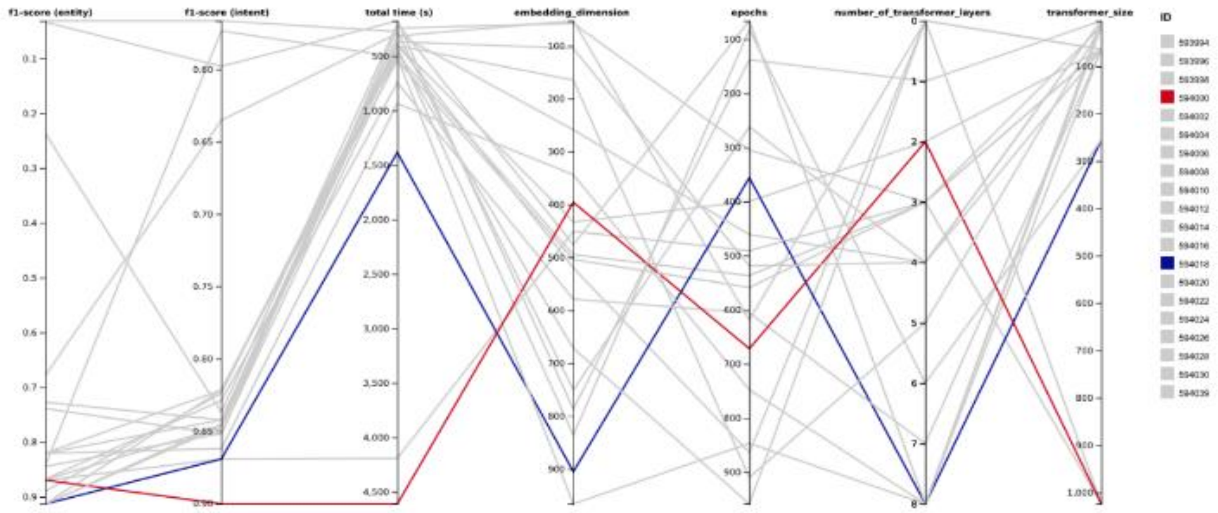

Fig 3. The Parallel Coordinates Plot of 20 runtimes shows the relationship between parameters and optimized.

classification, but no significant effect on entity extraction. *number_of_transformer_layers* has the least impact on both intent classification and entity extraction.

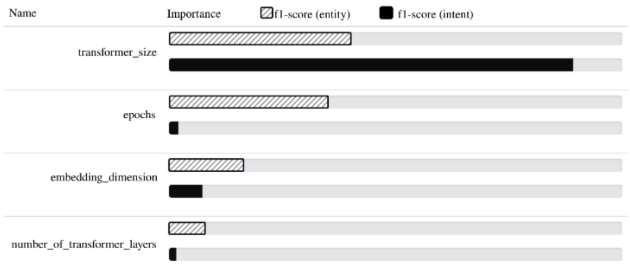

Fig 5. The Parallel Coordinates Plot of 20 runtimes shows the relationship between parameters and optimized.

5. Conclusions

The paper proposed an approach of integrating DIET with BO to find the most appropriate collection of hyperparameter values. The article has implemented the integrated DIET-BO model with the hotel chatbot conversation dataset, which are the conversation patterns between customers and chatbot. The preparation and normalization of the training dataset are described in detail and the evaluation indicators are analyzed. The implementation results show that the integrated DIET-BO model achieves the highest effectiveness when the most appropriate collection of hyperparameter values is found. This is shown by F1-scores of intent classification and entity extraction reaching 0.869 and 0.913, respectively. The results also show the impact of each hyperparameter on the DIET architecture.

References

[1] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open source language understanding and dialogue management," *arXiv Prepr. arXiv1712.05181*, 2017.

[2] A. Veltman, D. W. J. Pulle, and R. W. De Doncker, "The Transformer," in *Fundamentals of Electrical Drives*, Cham: Springer International Publishing, 2016, pp. 47–82. doi: 10.1007/978-3-319-29409-4_3.

[3] A. H. Mohammed and A. H. Ali, "Survey of BERT (Bidirectional Encoder Representation Transformer) types," *J. Phys. Conf. Ser.*, vol. 1963, no. 1, p. 12173, Jul. 2021, doi: 10.1088/1742-6596/1963/1/012173.

[4] M. Zhang and J. Li, "A commentary of GPT-3 in MIT Technology Review 2021," *Fundam. Res.*, vol. 1, no. 6, pp. 831–833, 2021, doi: https://doi.org/10.1016/j.fmre.2021.11.011.

[5] A. Conneau and G. Lample, "Cross-Lingual Language Model Pretraining," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., 2019.

[6] J. Pennington, R. Socher, and C. Manning, "{G}lo{V}e: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing ({EMNLP})*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.

[7] Y. Liu *et al.*, "Ro{\{}BERT{\}}a: A Robustly Optimized {\{}BERT{\}} Pretraining Approach." 2020. [Online]. Available: https://openreview.net/forum?id=SyxS0T4tvS

[8] M. Henderson, I. Casanueva, N. Mrkšić, P.-H. Su, T.-H. Wen, and I. Vulić, "{C}onve{RT}: Efficient and Accurate Conversational Representations from Transformers," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, Nov. 2020, pp. 2161–2174. doi: 10.18653/v1/2020.findings-emnlp.196.

[9] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V Le, "XLNet: Generalized autoregressive pretraining for language understanding," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, pp. 1–11, 2019.

[10] T. Bunk, D. Varshneya, V. Vlasov, and A. Nichol, "DIET: Lightweight Language Understanding for Dialogue Systems," 2020, [Online]. Available: http://arxiv.org/abs/2004.09936

[11] [11] M. H. M. Tarik, M. Omar, M. F. Abdullah, and R. Ibrahim, "Optimization of neural network hyperparameters for gas turbine modelling using bayesian optimization," *IET Conf. Publ.*, vol. 2018, no. CP749, 2018, doi: 10.1049/cp.2018.1308.

[12] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *J. Mach. Learn. Res.*, vol. 13, no. 10, pp. 281–305, 2012, [Online]. Available: http://jmlr.org/papers/v13/bergstra12a.html

[13] C. White, W. Neiswanger, and Y. Savani, "BANANAS: Bayesian Optimization with Neural Architectures for Neural Architecture Search," *35th AAAI Conf. Artif. Intell. AAAI 2021*, vol. 12A, pp. 10293–10301, 2021, doi: 10.1609/aaai.v35i12.17233.

[14] Y. Shen, M. Chen, S. Cai, S. Hu, and X. Chen, "BERT-RF Fusion Model Based on Bayesian optimization for Sentiment Classification," *2022 IEEE 2nd Int. Conf. Comput. Commun. Artif. Intell. CCAI 2022*, pp. 71–75, 2022, doi: 10.1109/CCAI55564.2022.9807729.

[15] F. Dernoncourt and J. Y. Lee, "Optimizing neural

network hyperparameters with Gaussian processes for dialog act classification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 406–413. doi: 10.1109/SLT.2016.7846296.

[16] W. Astuti, D. P. I. Putri, A. P. Wibawa, Y. Salim, Purnawansyah, and A. Ghosh, "Predicting Frequently Asked Questions (FAQs) on the COVID-19 Chatbot using the DIET Classifier," *3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021*, pp. 25–29, 2021, doi: 10.1109/EIConCIT50028.2021.9431913.

[17] L. Fauzia, R. B. Hadiprakoso, and Girinoto, "Implementation of Chatbot on University Website Using RASA Framework," in *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2021, pp. 373–378. doi: 10.1109/ISRITI54043.2021.9702821.

[18] M.-T. Nguyen, M. Tran-Tien, A. P. Viet, H.-T. Vu, and V.-H. Nguyen, "Building a Chatbot for Supporting the Admission of Universities," in *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*, 2021, pp. 1–6. doi: 10.1109/KSE53942.2021.9648677.

[19] V. Panchal and L. Kurup, "Educational chatbot on Data Structures and Algorithms for the visually impaired," in *2023 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, 2023, pp. 1–6. doi: 10.1109/ICNTE56631.2023.10146702.

[20] M. Arevalillo-Herraez, P. Arnau-Gonzalez, and N. Ramzan, "On Adapting the DIET Architecture and the Rasa Conversational Toolkit for the Sentiment Analysis Task," *IEEE Access*, vol. 10, no. September, pp. 107477–107487, 2022, doi: 10.1109/ACCESS.2022.3213061.

[21] Y. Kim and M. Chung, "An approach to hyperparameter optimization for the objective function in machine learning," *Electron.*, vol. 8, no. 11, 2019, doi: 10.3390/electronics8111267.

[22] M. Zulfiqar, K. A. A. Gamage, M. Kamran, and M. B. Rasheed, "Hyperparameter Optimization of Bayesian Neural Network Using Bayesian Optimization and Intelligent Feature Engineering for Load Forecasting," *Sensors*, vol. 22, no. 12, 2022, doi: 10.3390/s22124446.

[23] S. Wang, J. Zhuang, J. Zheng, H. Fan, J. Kong, and J. Zhan, "Application of Bayesian Hyperparameter Optimized Random Forest and XGBoost Model for Landslide Susceptibility Mapping," *Front. Earth Sci.*, vol. 9, 2021, doi: 10.3389/feart.2021.712240.

[24] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Adv. Neural Inf. Process. Syst. 24 25th Annu. Conf. Neural Inf. Process. Syst. 2011, NIPS 2011*, pp. 1–9, 2011.

[25] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019, doi: 10.11989/JEST.1674-862X.80904120.

[26] D. O. Oyewola, E. G. Dada, T. O. Omotehinwa, O. Emebo, and O. O. Oluwagbemi, "Application of Deep Learning Techniques and Bayesian Optimization with Tree Parzen Estimator in the Classification of Supply Chain Pricing Datasets of Health Medications," *Appl. Sci.*, vol. 12, no. 19, 2022, doi: 10.3390/app121910166.

[27] [27] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab869566

[28] Ms. Elena Rosemaro. (2014). An Experimental Analysis Of Dependency On Automation And Management Skills. International Journal of New Practices in Management and Engineering, 3(01), 01 - 06. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/25

[29] Stojanovic, N. . (2020). Deep Learning Technique-Based 3d Lung Image-Based Tumor Detection Using segmentation and Classification. Research Journal of Computer Systems and Engineering, 1(2), 13:19. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/6

[30] Sharma, R., Dhabliya, D. A review of automatic irrigation system through IoT (2019) International Journal of Control and Automation, 12 (6 Special Issue), pp. 24-29.