EcoDialTest: Adaptive Mutation Schedule for Automated Dialogue Systems Testing

Xiangchen Shen
School of Computer Science & Communication Engineer
Jiangsu University
Zhenjiang,China
3210608008@stmail.ujs.edu.cn

Haibo Chen
School of Computer Science & Communication Engineer
Jiangsu University
Zhenjiang,China
hbchen@stmail.ujs.edu.cn

Jinfu Chen[*]
School of Computer Science & Communication Engineer
Jiangsu University
Zhenjiang,China
jinfuchen@ujs.edu.cn

Jiawei Zhang
School of Computer Science & Communication Engineer
Jiangsu University
Zhenjiang,China
3210608050@stmail.ujs.edu.cn

Shuhui Wang
School of Computer Science & Communication Engineer
Jiangsu University
Zhenjiang,China
3200615001@stmail.ujs.edu.cn

Abstract*—With the rapid growth of Artificial Intelligence, dialogue systems have become increasingly powerful. Though** **Recurrent Neural Network power the dialogue systems, it also bring challenges to the systems' testing.** ***In order to ensure the safety of these systems, which we must pay attention to, DialTest showed up. DialTest broke the traditional test methods, it made innovation at many levels. We have to acknowledge this great contribution. However, DialTest has a smattering of shortcomings. It treats all seeds as equal, implying that it cannot adjust the energy assignment quickly, resulting in energy waste. Moreover, **DialTest's mutant sentences generated by a few original seed sentences in the late stage of variation.** ***This paper presents an improved DialTest with an adaptive mutation schedule, we called it EcoDialTest. EcoDialTest divides all the seed into three states, different states have different energy distribution strategies. We devise a new mutation strategy to improve the effectiveness and dependability of the seeds in the transformed seed set. All of these were implemented based on DialTest, we still adopt DeepGini impurity as the main guidance to guide the test generation process and utilize the three mutation operators as it does. Through ATIS, Snips and Facebook datasets, EcoDialTest was evaluated by two state-of-the-art models in the experiment. According to the result, we found that EcoDialTest attained lower values in both intent accuracy and slot accuracy than DialTest.***

***Keywords*—dialogue system testing, fuzzing, deep learning testing, natural language processing**

I. INTRODUCTION

With the continuous development of Artificial Intelligence (AI), Natural Language Processing has also made significant progress. In order to enable machines to understand natural language texts and react accordingly [1], researchers utilized the advantages of Recurrent Neural Networks (RNN) models. In recent years, since an increasing number of companies have vigorously developed RNN-driven dialogue systems [2], addressing the functionality of these systems has become essential. We can say that RNN-driven dialogue systems (like Apple ' s Siri) have brought convenience to our lives to a certain degree. However, they are not as perfect as people think. RNN-driven dialogue systems also have some potential problems. Sometimes a casual conversation may lead the robot to make excessive actions.

Due to some special causes, testing these systems becomes more difficult. For example, unlike traditional software testing, the RNN model often consists of millions of parameters and multiple layers. Moreover, it also requires large-scale data to train its logic [3]. These characteristics increase the difficulty of analyzing their behavior. As a result, Zixi Liu et al. designed DialTest [4], which is an effective testing method for RNN-driven dialogue systems with a fuzzing framework. DialTest integrates Synonym Replacement (SR), Back Translation (BT) and Word Insert (WI) transformed operators to make slight modifications to original seed data, and it adopts DeepGini [22] impurity to guide the seed selection process [5]. Test cases with a high DeepGini coefficient are more likely to find potential vulnerabilities because of the uncertainty about the prediction result.

However, DialTest performs certain shortcomings sometimes. The main problem is that it treats all the seeds as equal, all the test cases are generated randomly by the aforementioned three transformation operators and then be added to the bottom of the set, when DialTest needs a seed for the next fuzzing operation, it just chooses the next seed from the seed sentence set. The irregular arrangement of seeds in the set causes the selected seed may not an interesting seed, further letting a significant amount of energy be wasted on some seeds that cannot reach a high level of failure detection. Moreover, in traditional fuzzing, some researchers tried to solve the challenge that most of the test cases exercise some high-frequency paths [6] repeatedly. Here, we also found that DialTest may have a similar problem. In the later stages of fuzzing, DialTest may begin to mutate some sentences that are generated by a few original seed sentences. This reduces the diversity of generated testing inputs to a certain extent. In addition, we have found that the long sentence always shows its difficult to make a breakthrough in the Gini level through a single mutation. As a result, DialTest may abandon these sentences inadvertently.

This paper proposes a new mutation strategy. We modeled each seed sentence as a "bandit" and tried to trade-off between exploration (the seed in this stage has a low Gini coefficient but can be transformed to a higher Gini level easily) and exploitation (the seed in this stage has reached the

[*] Corresponding author

978-1-6654-5278-6/23/$31.00 ©2023 IEEE
DOI 10.1109/SANER 56733.2023.00113

predefined threshold of Gini impurity) during fuzzing [7]. Different stages correspond to different mutation strategies. Before the exploitation and exploration phases, we add an initial phase. In addition to the exclusive allocation strategy, EcoDialTest designs a determined mutation process when transforming a seed for the first time in order to avoid dropping seeds with long lengths easily. In the exploitation phase, we applied a power schedule to avoid the overexploitation of a few seeds. When all the seeds in the exploitation seed set exhaust all the allocated energy, EcoDialTest will transfer into the exploration period. In the exploration phase, EcoDialTest focuses on transforming seeds with a low Gini coefficient, but the seeds have great potential to excavate. Finally, we implemented all of these improvements on DialTest and developed a new test tool named EcoDialTest, an adaptive mutation schedule dialogue system test tool. In the final experiment, we found that, EcoDialTest had lower intent accuracy and lower slot accuracy in most cases compared with DialTest. This indicates that EcoDialTest may detect the RNN-driven NLU model’ s hidden vulnerabilities more effectively.

In summary, the following are the contributions we made in this paper.

- We proposed an automatic adjustment model to improve the existing test tool for the RNN-driven dialogue system. When it is exploring, it calculates the frequency that a seed can convert to a good seed, and adjust the energy cost according to the actual situation constantly. Meanwhile, when it is exploiting, it also ensures that a sentence is not mutated frequently, preserving the diversity of the transformed data set.
- Breakthrough of tenacious seeds. We designed a determinate mutation process. When an original seed cannot be transformed easily into a good seed or a potential seed. We will make two different variations in succession.
- We implement our improvements on DialTest and develop EcoDialTest, an automated dialogue system test tool with an adaptive mutation schedule. Results showed that EcoDialTest can achieve lower intent accuracy and lower slot accuracy when compared with DialTest.

The remainder of this paper is structured as follows: Section 2 introduces background on the fuzzing, dialogue system, Dialogue Test and Multi-Armed Bandit problem. Section 3 describes the implementation method of our mutation strategies. Followed by Section 4 Section 5, which presents our experimental design scheme and results analysis. Section 6 discusses some related work for Dialogue Test. Finally, we summarize this research and organize the future direction in Section 7.

II. Background

This section presents an outline of fuzzing, the workflow of the dialogue system, and describes the NLU model in detail. We subsequently introduce some research on the dialogue test. Finally, we briefly explain the Multi-Armed Bandit problem.

A. The brief introduction of fuzzing

Fuzzing is an active topic both in research and practice [21]. Fuzzy testing is a method to discover software vulnerabilities by providing unexpected inputs to the target systems and monitoring any abnormal results. The test cases generated automatically or semi-automatically are input into the program to detect whether the target program runs abnormally [8]. It analyzes the input cases that cause the crashes to test whether the program will be abnormal, so as to find possible security vulnerabilities in the application. Fuzzy testing has benefited from important algorithmic innovations and increased computational power [19]. As a result, fuzzing tools can be used to detect various software’s hidden security vulnerabilities, including integer overflow vulnerabilities, buffer overflow vulnerabilities, etc. [9]. For any fuzzing test tool, executing more test cases in a certain amount of time enhances the function of the fuzzing campaign by either achieving the same coverage faster or attaining better coverage with the same resources [23]. Nowadays, an increasing number of approaches model fuzzing as an optimization problem and try to mutate program seed inputs to address this problem [20].

B. The workflow of dialogue system

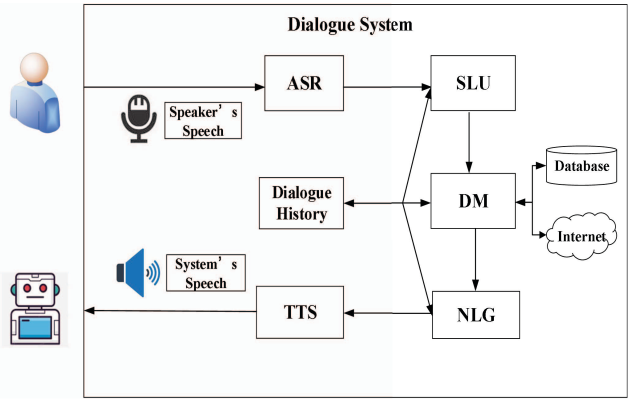

Fig. 1. Workflow of Dialogue System

The general workflow of dialogue systems is mainly built with five technologies. First, they need a module called the speech recognizer to implement Automatic Speech Recognition (ASR). And the outcome of the speech recognizer is the input to the Nature Language Understanding (NLU) module. NLU is a critical component of conversational dialogue systems [24]. The purpose of NLU is to gain a semantic representation of the input. Then the output of the NLU will flow into the Dialogue Management (DM) module. Typically, Dialogue Management inquires about a native database and searches the Internet for data. Its goal is to decide what the system must do next in response to users ’ input. The dialogue manager’ s decision about what the system must do next is the input to the Natural Language Generation (NLG) module, whose purpose is to transform the decision into one or more sentences in text format that must be grammatically correct. Finally, the NLG module's sentences are fed into the Text-to-Speech Synthesis (TTS) module, which converts the sentence into the dialogue system's voice [10].

sentence	tell	me	the	trains	from	Shanghai	to	Beijing
slots	o	o	o	o	o	B-FromCity	o	B-ToCity
intent	Search_Trains							

Fig. 2. An example of slot and intent analysis

In our research, we focused on the NLU model. The model mainly consists of two parts: slot filling and intent detection. Slot filling’s primary mission is to search for keywords in the input sentence and fill them into a predefined semantic slot. It is typically treated as a sequence labeling problem. And intent detection is the classification of the input sentence based on sentence semantics. As a result, intent detection can be seen as a sentence classification problem. Due to their important status in the dialogue system, we utilize them to evacuate EcoDialTest in our research. An example of slot and intent analysis is provided in Fig.2.

C. Dialogue Test

Before the DialTest, a few studies had tried to detect RNN-driven dialogue systems’ abnormal behavior. For example, Bozic et al. [17] proposed a testing method based on AI planning to verify the communication competence of chatbots. They designed a test tool to monitor multi-dimensional indicators.

As the first RNN-driven dialogue system testing tool, DialTest is greatly different from conventional fuzzing test tools, like AFL [11]. Firstly, DialTest integrated three mutation operators together to ensure the efficiency of testing and better improvement. They are Synonym Replacement (SR), Back Translation (BT) and Word Insert (WI). Test cases generated by these specialize in slot filling and intent detection. Second, in order to better adapt to dialogue systems, DialTest experiments with intent detection and slot filling using the DeepGini [22] impurity as guidance. Test cases with a high DeepGini coefficient are more likely to find potential vulnerabilities, because of the uncertainty about the prediction result. After every variation, DialTest works out the degree of the change and then determines whether a sentence can become an interesting seed. The applied DeepGini [22] function was applied to make the generated test data more likely to find potential bugs. Thirdly, DialTest has high practicality, it has been evaluated on four state-of-art NLU models and three widely-used datasets, and this has demonstrated its powerful capability. More importantly, it can not only identify erroneous behaviors in the system but also improve the dialogue systems’ effectiveness after retraining.

D. Multi-Armed Bandit Problem

Multi-Armed Bandit (MAB) is a classic exploration and exploitation problem [12]. A gambler did not know the real profit of each slot machine in advance. He needs to think about choosing which slot machine to play next time or whether to stop gambling according to the results. Finally, he achieves the purpose of maximizing his income. This classic question focuses on a core trade-off in reinforcement learning: should we explore new possibilities or should we keep to the best choice we have known? During the process of exploration, we should know that the more attempts on an arm, the more information we will get. And while we can infer rewarding expectations from this data, it is possible that we will spend too much time and money on a machine with low profits. If we have sufficient information to make a decision, we will select arms with the highest expectations, but this may make us miss the chance to find a better machine. We call this process “exploitation”. Therefore, achieving a trade-off between exploration and exploitation is a key step to achieving higher rewards [13].

In this paper, we modeled each seed sentence as a "bandit". During the exploration and exploitation period, we designed the exploitation state, when EcoDialTest tries to transform the seeds that have already exceeded a predefined threshold of Gini impurity and strives to reach a higher value. The exploration state is when EcoDialTest runs out of all the energy in the exploitation state and starts to mutate the seed with a lower Gini coefficient in the interesting seed set because the seed have the potential to be transformed to a higher Gini level easily.

III. Improvement of Test Tool

In this section, we introduce the three transformation operators and mutation schedule approaches we designed. Contents include how we control the high frequency of variation for a few seeds and make a balance between exploration and exploitation. Also, include how we attempt to break the stability of tenacious seeds.

A. Framework

As shown in Fig.3, EcoDialTest calculates the Gini impurity after every mutation. When it meets an indomitable seed at the initial stage, it will transform it by using fixed mutation operators (Word Insertion first and then Synonym Replacement). With this method, EcoDialTest can effectively select more data and trigger potential vulnerabilities in the model. According to the change, EcoDialTest divides seeds into an Exploration or Exploitation state. During exploration and exploitation states, each state has its own energy distribution strategies. And EcoDialTest has a method for transitioning between the two states. Finally, the generated seeds are stored in the transformed data set. The model input data is in the set and harvest evaluation result (Slot Filling Accuracy and Intent Detection Accuracy).

B. Sentence Transformation

(1)**Synonym Replacement (SR):** This variation method changes the sentence by replacing a single word with a synonym, thus keeping the meaning of the sentence unchanged while changing the structure of the sentence. We employ WordNet [25] ——a large-scale lexical database for English, it can locate and replace synonyms. WordNet [25] provides a more effective combination of traditional

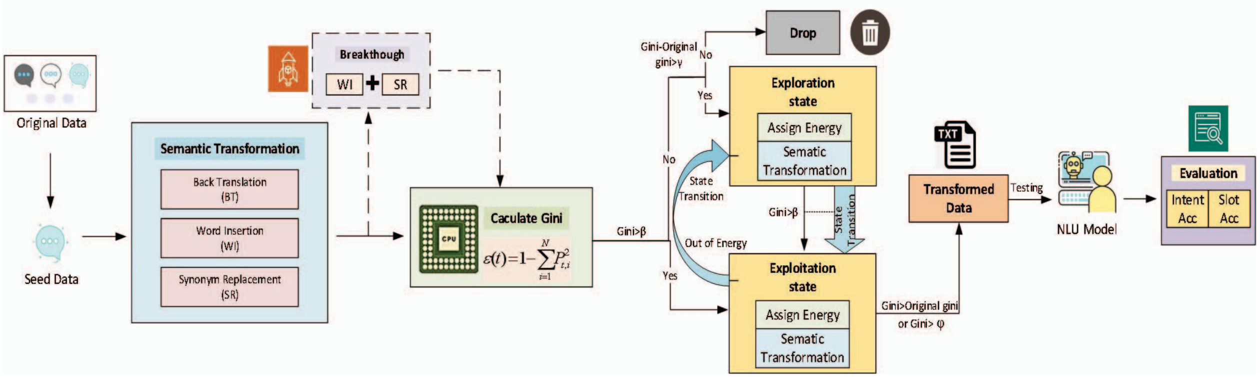

Fig. 3. Overview of EcoDialTest

```
Algorithm 1: The algorithm of EcoDialTest
input  : The tested model $\mathbb{R}$,Transformed seed set $\mathbb{T}$,Interest seed set
         $\mathbb{I}$,Seed sentence $\mathbb{S}$,Exploitation seed set $\mathbb{E}$
output: Transformed test sentences

numAug=0;
$\mathbb{E}$=$\mathbb{S}$;
while $\mathbb{I}$ and $\mathbb{E}$ is not empty do
    if $\mathbb{E}$ is empty and $\mathbb{I}$ is not empty then
        $\mathbb{E}$.add($\mathbb{I}$.pop());
    end
    s=$\mathbb{E}$.pop();
    OriGini=CalcuateGini(s);
    FailTime=0;
    MaxFailTime=EnergyAssign();
    while FailTime<=MaxFailTime do
        if s.IsBreakthrough() then
            newSentence=BreakthroughTransform(s);
        else newSentence=Transform(s);
        Gini=CalculateGini(newSentence);
        if Gini $\geq\beta$ then
            if Gini>$\psi$ or Gini>OriGini then
                $\mathbb{E}$.add(newSentence) ;
                $\mathbb{T}$.add(newSentence) ;
                numAug=numAug+1;
            end
        if Gini-OriGini>$\gamma$ then
            $\mathbb{I}$.add(newSentence);
        else
            FailTime=FailTime+1;
        end
    end
end
return numAug,$\mathbb{T}$
```

lexicographic information and modern computing. WordNet [25] is an online vocabulary database designed for use under program control. English nouns, verbs, adjectives, and adverbs are organized into sets of synonyms, each of which stands for a different concept [25].

(2) **Back Translation (BT):** This mutation method translates the target sentence into the transitional language and then translates it back to the initial language. The translation tool can improve the diversity of sentences while keeping the meaning unchanged. We choose Chinese, Spanish and French as DialTest does. Because they have completely different alphabets and grammar. We adopted the Baidu translation API to undertake the BT mission.

(3)**Word Insertion (WI):** This variation method transforms the sentence by inserting words. It adds some words unrelated to the intention to the sentence while keeping the grammar correct and the meaning unchanged. We follow DialTest to implement the approach based on the BERT language model and select three pre-trained language representation models with different parameter configurations in order to insert new words into the original sentences.

C. Exploration, Exploitation and Breakthrough

MAB's application in traditional fuzzing tests needs the test tool to calculate the reward probability and choose the seed with the highest reward probability [7]. However, in the dialogue systems test, there are no so-called "new paths". As a result, we can't employ it as before. During RNN-driven dialogue system testing, the primary mission we need to accomplish is to search for more seeds with a high Gini coefficient and ensure seed diversity as much as possible at the same time.

Moreover, all the seeds are classified into three periods, and different periods have different energy distribution strategies. And we use fail times ($fail\_num_{id}$) to control the energy like DialTest. The interesting seed set is defined as the input of mutation during the exploration period. The seeds in the exploitation seed set are prepared to execute the variation in the exploitation period. All seeds that can be filtered and finally inputted into the RNN model are stored in the transformed seed set. We ensure that the transformed seed set size keeps the same as the original set size.

The whole process is shown in Algorithm 1. And the three periods are listed below:

(1)**Initial Period:** In this period, all seeds are unfuzzed, we add every initial seed to the transformed data set above all. EcoDialTest initializes each seed's $fail\_num$ to α. In order to ensure it doesn't skip the seeds that are not preferred easily, we designed a directed mutation process in this state. After an initial seed uses up the distributed energy, EcoDialTest will try to take measures to break through its stability if the variant seed isn't added to the exploitation seed set or the interesting seed set. Because most tenacious seeds cannot mutate to a high level of Gini due to their length, EcoDialTest tries WI first, followed by SR, after performing these mutations in the specified order. The transformed sentence will be added to the interesting seed set, waiting for subsequent mutation. Certainly, we ensure both SR and WI processes should get an increased Gini. We think this can break the sentence's structure in part, experiments also proved this.

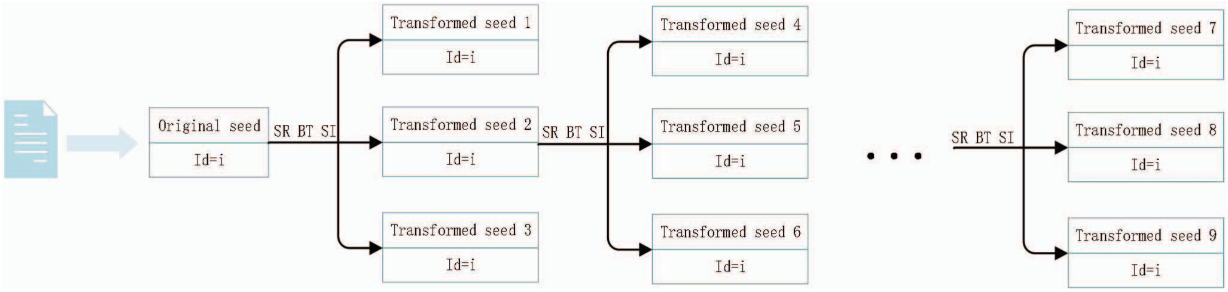

Fig. 4. Overview of numbering rules

(2)**Exploitation Period:** In this period, all the seeds' Gini impurities are over β (one of the pre-defined thresholds). EcoDialTest number the transformed seed has the same id as its initial seed, as shown in Fig.4. It records the quantity of the seeds with the same id in the transformed seed set num_{id}. We defined $fail\_num_{id}$ as the max fail times when the $seed_{id}$ execute transformation. To ensure diversities of the model input, we design the energy strategy as follow:

$$fail\_num_{id} = \lceil \alpha \cdot (\max\_num - num_{id}) / \max\_num \rceil \quad (1)$$

Here max_num is a fixed constant, which means the max quantity all the seeds with the same id can reach.

In this state, if the generated seed's Gini (NewGini) satisfy following condition:

$$\begin{cases} NewGini > OriginalGini \\ \quad or \\ NewGini > \varphi \end{cases} \quad (2)$$

EcoDialTest will add it to the transformed seed set. φ here is another pre-defined threshold, which is used to avoid semantic change caused by overexploitation. If all the seeds in the exploitation seed set run out of energy, EcoDialTest transfers into the exploration period.

(3)**Exploration Period:** In this period, EcoDialTest focuses on mutating seeds from the interesting seed set. We compute the frequency f_{id} of interesting $seed_{id}$ can be mutated into an exploitation seed. The following is a concrete calculating method:

$$f_{id} = good_{id} / sum_{id} \quad (3)$$

Here $good_{id}$ represents the number of seeds that can be transformed from $seed_{id}$ to achieve victory over β gini and added to the exploitation seed set. The sum_{id} means the total times the $seed_{id}$ has mutated in all.

EcoDialTest in this period assigns energy according to the frequency as:

$$fail\_num_{id} = \min(fail\_num_{id}, \lfloor \alpha \cdot (1 - f_{id}) \rfloor) \quad (4)$$

In this state, if the NewGini satisfy the following condition:

$$NewGini - OriginalGini > \gamma \quad (5)$$

EcoDialTest will add it to the bottom of the interesting seed set. However, if

$$NewGini > \beta \quad (6)$$

EcoDialTest will add it to the Exploitation seed set, then it will transfer back to the exploitation period and start to exploit the new additional seed.

IV. Experimental Design

In this section, we describe the experimental design. It mainly consists of two parts: we introduce datasets and NLU models that we will test first. Then we discuss the experimental metric.

A. The Datasets and RNN Models

In our experiment, we evaluated three popular datasets, i.e. ATIS [14] Snips [15] and Facebook's multilingual dialogue corpus[2].

The Airline Travel Information System (ATIS) data set, an open corpus in the air travel field. It contains over 5,000 in total, which consists of over 4,400 training data, 500 valid data, and over 800 test data.

The Snips data set is a data set that comes from personal voice assistants. Its diverse intent and a larger vocabulary make it more complex than the ATIS data set. There are over 13,000 training data points, 700 valid data points, and 700 test data points in total.

Facebook's multilingual dialogue corpus contains tag dialogue data in three languages (English/Spanish/Thai). The corpus comes from intelligent assistants and involves weather queries, music playing and other fields. In this paper, we only consider the English corpus. There are more than 30,000 training data, more than 4,000 valid data and more than 8,000 test data.

At preliminary phase, we use two NLU models to evaluate EcoDialTest's performance. They are listed below:

LSTM$_{\text{ELMo}}$ [16], a LSTM model which is pre-trained with ElMo for word embedding. The embedded size of

[2] https://fb.me/multilingual_task_oriented_data

words generated by ELMo is 1024, and the number of LSTM hidden units is 200.

BLSTM$_{\text{ELMo}}$ [16], a BLSTM model which is also pre-trained with ElMo for word embedding. The parameter configurations of ELMo and LSTM$_{\text{ELMo}}$ hidden unit are identical.

B. Evaluation Metric

In order to evaluate the performance of EcoDialTest, we use two evaluation indicators, intent accuracy and slot accuracy. For every input, the model can make a classification. We measure EcoDialTest's classification accuracy according to the proportion of correctly classified sentences:

$$Intent\_Acc = \frac{|Correct\_classification|}{|Total\_sentence|}$$

For slot filling, the model can label every word in a sentence. Here, we do not directly make a comparison between the predicted label and the actual label of each word. Instead, we used the ratio of the correct number of entities to the total number of entities:

$$Slot\_Acc = \frac{|Correct\_Entity|}{|Total\_Entity|}$$

V. Analysis of Experimental Result

In order to validate all the sentences in the transformed seed set can more effectively detect RNN-driven NLU model's hidden vulnerabilities. We compare it with DialTest's Gini guide generation and record the three aforementioned test sets' intent accuracy and slot accuracy. In the experiment, we ensure that the transformed seed set size is the same as the original set size.

We recorded the slot filling accuracy and intent detection accuracy on LSTM$_{\text{ELMo}}$ and BLSTM$_{\text{ELMo}}$ separately. The result shows that compared with the experimental result of DialTest, the accuracy of transformed seed sets generated by EcoDialTest decreased in most cases. This outcome reflects that EcoDialTest has a better effect on RNN-driven NLU model testing in general.

TABLE I. Testing Results on Different Models and Datasets Compared with DialTest

Model	**Dataset**	**Intent Acc**			**Slot Acc**		
		Ori.	**Dial.**	**Eco.**	**Ori.**	**Dial.**	**Eco.**
LSTM$_{\text{ELMo}}$	ATIS	0.98	0.65	**0.41**	0.93	0.63	**0.52**
	Snips	0.99	0.74	**0.40**	0.91	0.71	**0.51**
	Facebook	0.99	0.59	**0.32**	0.85	**0.36**	0.45
BLSTM$_{\text{ELMo}}$	ATIS	0.98	0.65	**0.21**	0.94	0.57	**0.36**
	Snips	0.99	0.73	**0.42**	0.95	0.57	**0.41**
	Facebook	0.99	0.69	**0.42**	0.89	0.49	**0.48**

VI. Related Work

A few researches [17][18] attempted to detect abnormal behavior in RNN-driven dialogue systems. Bozic et al. proposed a metamorphic testing approach for the testing of chatbots relies on metamorphic relations. The research concentrated on creating test samples for slot filling and intent detection tasks are even more limited. Zixi Liu et al.[4] made innovative attempts.

The main differences between EcoDialTest and DialTest are listed here:

(i) All the cases in the seed sentence set are generated randomly by SR, BT and WI, this means that there is no order between all seed permutations. However, DialTest only chooses the next seed from the seed sentence set before the fuzzing operation. In our research, we classified all the seeds into three periods, different periods have different energy distribution strategies. We tried to strike a balance between the exploration and exploitation periods.

(ii) EcoDialTest ensures diversity of the transformed data set due to the aforementioned energy distribution strategies, rather than mutating the sentences generated by a few original seed sentences at a later stage. The purpose is to make the transformed seed set be filled with more sentences with different meanings and to make the sentences have a more balanced proportional distribution.

(iii) EcoDialTest designs a directed mutation method that can break through more tenacious seeds. In order to ensure it doesn't skip the seeds that are not preferred easily. The experiment performed that most tenacious seeds can't mutate to a high level of Gini because of their long length, As a result, we devised a method for performing WI first, followed by SR, after performing the two mutations in the prescribed order. The transformed sentence will be added to the interesting seed set, waiting for the following mutation.

VII. Conclusion and Future Work

In this paper, we propose an automatic adjustment model and use it for mutation scheduling. In order to achieve improvement of the existing test tool for the RNN-driven dialogue system. Different periods have different distribution strategies for addressing overexploitation problems and so on. Based on these, we also design methods to try to break more indomitable seeds because a sentence's length may make its structure stable. Finally, we implemented these algorithms on an adaptive test tool for an RNN-driven dialogue system called EcoDialTest. The experimental results show that EcoDialTest has decreased the accuracy of transformed seed sets in most cases, compared with DialTest.

In the future, we will conduct further experimental verification, try to use EcoDialTest to test more models, and constantly improve our algorithm. The reduction of time consumption is one of the aspects we should think about. Moreover, we may think about combining machine learning with our distribution strategies to achieve higher energy efficiency.

Acknowledgments

The project was supported by the National Natural Science Foundation of China (NSFC) (Grant no.62172194 and U1836116), the National Key R&D Program of China (Grant no.2020YFB1005500), and the Leading-edge Technology Program of Jiangsu Natural Science Foundation (Grant no. BK20202001).

REFERENCES

[1] Chowdhary, K. (2020). Natural language processing. Fundamentals of artificial intelligence, 603-649.

[2] Deriu, J., Rodrigo, A., Otegi, A., Echegoyen, G., Rosset, S., Agirre, E., & Cieliebak, M. (2021). Survey on evaluation methods for dialogue systems. Artificial Intelligence Review, 54(1), 755-810.

[3] Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). Exploring strategies for training deep neural networks. Journal of machine learning research, 10(1).

[4] Liu, Z., Feng, Y., & Chen, Z. (2021, July). DialTest: automated testing for recurrent-neural-network-driven dialogue systems. In Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, 115-126.

[5] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

[6] Böhme, M., Pham, V. T., & Roychoudhury, A. (2016, October). Coverage-based greybox fuzzing as markov chain. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 1032-1043.

[7] Yue, T., Wang, P., Tang, Y., Wang, E., Yu, B., Lu, K., & Zhou, X. (2020). EcoFuzz: Adaptive Energy-Saving Greybox Fuzzing as a Variant of the Adversarial Multi-Armed Bandit. In 29th USENIX Security Symposium (USENIX Security 20), 2307-2324.

[8] Wang, C., Tok, Y. C., Poolat, R., Chattopadhyay, S., & Elara, M. R. (2021). How to secure autonomous mobile robots? An approach with fuzzing, detection and mitigation. Journal of Systems Architecture, 112, 101838.

[9] Oehlert, P. (2005). Violating assumptions with fuzzing. IEEE Security & Privacy, 3(2), 58-62.

[10] López-Cózar, R., Callejas, Z., Griol, D., & Quesada, J. F. (2014). Review of spoken dialogue systems. Loquens, 1(2), 012.

[11] Gan, S., Zhang, C., Qin, X., Tu, X., Li, K., Pei, Z., & Chen, Z. (2018, May). Collafl: Path sensitive fuzzing. In 2018 IEEE Symposium on Security and Privacy (SP),679-696.

[12] Bubeck, S., & Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends® in Machine Learning, 5(1), 1-122.

[13] Patil, K., & Kanade, A. (2018). Greybox fuzzing as a contextual bandits problem. arXiv preprint arXiv: 1806.03806.

[14] Hemphill, C. T., Godfrey, J. J., & Doddington, G. R. (1990). The ATIS spoken language systems pilot corpus. In Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990.

[15] Coucke, A., et al.(2018). Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint arXiv: 1805.10190.

[16] Siddhant, A., Goyal, A., & Metallinou, A. (2019). Unsupervised transfer learning for spoken language understanding in intelligent agents. In Proceedings of the AAAI conference on artificial intelligence 33(01), 4959-4966.

[17] Bozic, J., & Wotawa, F. (2019, October). Testing chatbots using metamorphic relations. In IFIP International Conference on Testing Software and Systems. Springer, Cham. 41-55.

[18] Huang, W., Sun, Y., Zhao, X., Sharp, J., Ruan, W., Meng, J., & Huang, X. (2021). Coverage-guided testing for recurrent neural networks. IEEE Transactions on Reliability. 1191-1206.

[19] Andronidis, A., & Cadar, C. (2022). SnapFuzz: High-Throughput Fuzzing of Network Applications. In Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2022). 340–351.

[20] Wu, M., Jiang, L., Xiang, J., Zhang, Y., Yang, G., Ma, H., ... & Zhang, L. (2022, May). Evaluating and improving neural program-smoothing-based fuzzing. In Proceedings of the 44th International Conference on Software Engineering , 847-858.

[21] Molina, F., d'Amorim, M., & Aguirre, N. (2022). Fuzzing class specifications. In Proceedings of the 44th International Conference on Software Engineering (ICSE '22). 1008–1020..

[22] Feng, Y., Shi, Q., Gao, X., Wan, J., Fang, C., & Chen, Z. (2020, July). Deepgini: prioritizing massive tests to enhance the robustness of deep neural networks. In Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis , 177-188.

[23] Giuffrida, E. G. C., & van der Kouwe, H. B. E. (2022). Snappy: Efficient Fuzzing with Adaptive and Mutable Snapshots. In Proceedings of the 38th Annual Computer Security Applications Conference (ACSAC '22). 375–387.

[24] Zhu, S., Chen, L., Cao, R., Chen, Z., Miao, Q., & Yu, K. (2021, October). Few-Shot NLU with Vector Projection Distance and Abstract Triangular CRF. In CCF International Conference on Natural Language Processing and Chinese Computing 505-516.

[25] Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41.