Attacking the Dialogue System at Smart Home

Erqiang Deng[1,2(✉)], Zhen Qin[1,2,3], Meng Li[1,2,3], Yi Ding[1,2,3,4], and Zhiguang Qin[1,2,3]

[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China
dylandeq@outlook.com
[2] Network and Data Security Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China, Chengdu, China
[3] Institute of Electronic and Information Engineering of UESTC, Chengdu, China
[4] Institute of Electronic and Information Engineering of UESTC in Guangdong, Guangdong 523808, China

Abstract. Intelligent dialogue systems are widely applied in smart home systems, and the security of such systems deserves concern [1, 2]. In this paper, we design a threatening scenario of dialogue systems at a smart home. A trojan robot is disguised as one part of the whole system but generates dialogue adversarial examples to attack the normal robots according to the information of users. To achieve the goal in such a scenario, the responding speed, the correctness of the grammar, and the consistency of semantic is necessary. Based on these requirements, we propose a novel method named Attention weight Probability Estimation Attack (APE) to allocate the keys words in dialogue and substitute these words with synonyms in real-time. We perform our experiments on popular classification datasets in the DNN model, and the result shows that APE effectively attacks the system with low responding time and a high success rate.

Keywords: Smart home · Security · Dialog system · Adversarial example

1 Introduction

In the scenario of smart home, voice interaction has been the optimal choice, because of the natures of short responding time, fluency of interaction and the convenience of operation [3]. These voice interaction systems are composed of speech signal processing system (SSP) and natural language processing system (NLP). In the front end, the SSP system samples the original information and convert the frequency signals to text information [4], and in the back end, the NLP system recognizes the text information and generates corresponding feedback to users, with comprehending the semantic information of input. In both SSP and NLP parts, deep neural network (DNN) is mainstream technology.

However, previous works have shown the vulnerability of DNN. Szegedy et al. [5] firstly added small perturbations to the input images, which led to the misjudgment of the DNN model with high classification confidence. Jia and Liang [6] deceived the reading

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2021
Published by Springer Nature Switzerland AG 2021. All Rights Reserved
H. Gao et al. (Eds.): CollaborateCom 2020, LNICST 349, pp. 148–158, 2021.
https://doi.org/10.1007/978-3-030-67537-0_10

comprehension model by changing a few words in a paragraph. With the deployment of the dialogue system at smart homes, the latent risk of security is inevitable and deserves concern. Based on this consideration, we design this paper to research the security issue and try to expose the vulnerability of such systems from the perspective of an attacker.

To achieve this goal, we design a Trojan scenario [7] at a smart home. As shown in Fig. 1, robot No.1 is a Trojan invader. It receives the input of the user and generates an adversarial example to attack the normal robot No.2, leading it misclassify the information, although only a few words are changed and the semantic information seems consistent with the original input. The existence of the Trojan invader is possible if the number of robots at a smart home is large.

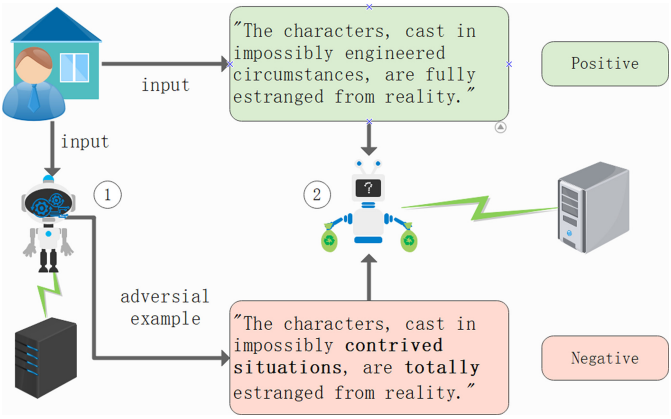

Fig. 1. The Trojan scenario at smart home. Robot No. 1 is a invader, it receives the information of user and tries to attack the normal robot No. 2 by crafting adversarial examples.

In this paper, only the NLP part of the dialogue system is involved, because the complexity and variability of language are contained in text, which makes the NLP system more vulnerable to be attacked. We performed our experiments on IMDB sentiment classification tasks and generated adversarial examples to attack the DNN models. In such a Trojan scenario, the responding speed is critical to perform an attack, and the grammar correctness and the semantic consistency are also important to disguise itself from being perceived by humans. To satisfy these requirements, we propose a novel method based on the attention mechanism, which indicates the distribution of importance of the words in a paragraph. We estimate the probability of a successful attack by attention distribution and select the words with high attention weights to be substituted. Our main contributions are:

Proposing a novel method to estimate the probability of successful attack in text.

Proposing an algorithm to search possible adversarial words in short time.

Designing a possible scenario of adversarial attack in physical world and craft adversarial examples in real-time.

2 Related Work

Szegedy et al. [5] proposed the notion of adversarial attack in the area of computer vision (CV). They added perturbation into images which were negligible to human but misguided the DNN. Szegedy developed this branch of security in DNN, and many works were engaged in this area. The attack methods of FGSM [8], JSMA [9], and C&W [10] were proposed and the defensive methods of adversarial training and distillation followed in CV.

In text area, Papernot et al. [11] calculated the forward derivative, i.e. Jacobian to generate adversarial text sequences on RNN. TextFool [11] used the concept of FGSM and USES backpropagation to calculate the cost gradient of each training sample. The classifier error was made by inserting, modifying, and deleting the character containing the largest gradient by identifying it and naming it hot character. Xue et al. [13] proposed a method to generate a dataset for robustness evaluation of the Q&A system in the black-box scenarios. Ren et al. [1] proposed PWWS to attack text classification models. They calculated the word saliency and the classification probability, and performed a greedy search to generate adversarial examples.

These works succeeded in the area of robustness and security, while they are not suitable for our Trojan scenario. The methods using greedy search may not satisfy the requirement of real-time, and in most embedding equipments, only inference is supported and the methods based on gradient backpropagation cannot be applied. To perform an in time attack at smart home, a fast response method must be developed.

3 Attention Probability Estimation

The attention mechanism [14] and its upgraded technologies [15, 16] have boosted performance on a range of NLP tasks. The attention weights explicitly reflect representations of input components, and these weights can be used to identify the importance of the input from the perspective of statistical distribution. Based on this feature of the attention mechanism, we approximate the attention weights distribution as the saliency map of input when we deploy a real-time attack, to reduce the expense of gradient calculation of backpropagation.

3.1 Adversarial Attack in Text

Suppose a text sequence $x = \{\omega_1, \omega_2 \ldots \omega_n\}$, where $\omega_i \in D$, and D is the space of dictionary. A DNN model receives the input x and outputs the prediction of classification of x, $F(x) = y$.

An adversarial example is defined as:

$$F(x) = y \text{ and } F(x') \neq y \tag{1}$$

where $x \approx x'$. It indicates that the adversarial example x' is similar to x and the difference between x and x' is negligible for humans, but it makes model F misjudge the information.

3.2 Attention Based Classification

Attention Mechanism

The attention mechanism is designed to focus on the key parts of a whole sequence, to strengthen the ability to abstract the most important information. Mathematically, the attention mechanism is a weights-refreshed affine transformation. It can be expressed as:

$$f(x) = \tilde{A}x + b = \sum_{\mathrm{i}} \alpha_{\mathrm{i}} x_i + b \tag{2}$$

Different from the parameters of full connection layer in DNN, the α weights are not directly updated by backpropagation, but are determined by query-key calculation:

$$\alpha_{\mathrm{i}}(Q, K) = \mathrm{softmax}(q_i^T k_i) \tag{3}$$

The query-key pair with a higher product of multiplication will be allocated a higher weight in affine transformation. The attention mechanism shows its effectiveness in the scenario of long sequences to solve the gradient vanishing problem (Fig. 2).

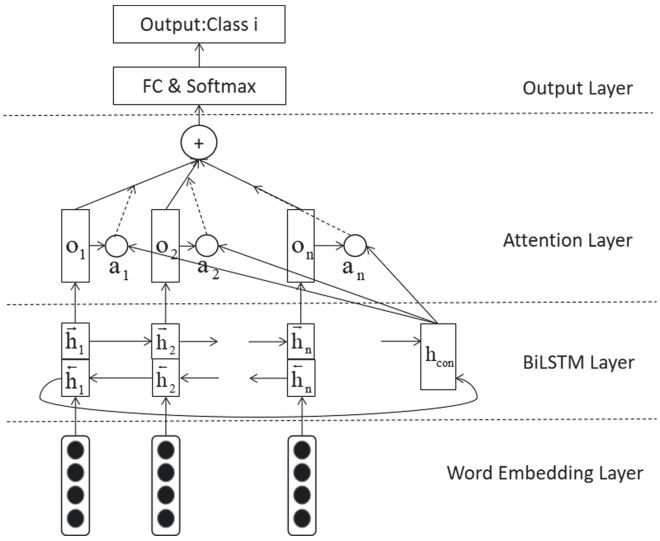

Fig. 2. Attention BiLSTM model.

Self-attention in LSTM Model

A self-attention BiLSTM model is applied to extract features. $\vec{h}_{\mathrm{i}}$ and $\overleftarrow{h}_{\mathrm{i}}$ is the hidden vectors of step i, o_i is the output of step i. h_{con} is the concatenated vector of $\vec{h}_{\mathrm{n}}$ and $\overleftarrow{h}_1$,

where n means the numbers of the input sequence. a_i is set as the weight of vector o_i, and is calculated by the function:

$$\alpha_i = \frac{\exp(o_i^T h_{con})}{\sum_{j=1}^{n} \exp(o_j^T h_{con})} \tag{4}$$

The hidden layer has the information of whole input, the output contributes more can be allocated a higher attention weight, because of the larger product of the vector multiplication. The final output of LSTM layer is:

$$output = \sum_{i=1}^{n} \alpha_i o_i \tag{5}$$

3.3 Word Substitution

In this paper, we perform the attack based on the synonym substitution and estimating the replacement order by the weight of attention. Our method is named Attention Weight Probability Estimation Attack (APE).

Attention Weight Estimation
We define a text sequence with n words $x = \{\omega_1, \omega_2, \omega_3, \ldots, \omega_n\}$. We input the text sequence to a pretrained attention-BiLSTM model, and the output of the model contains the predicted label y and attention distribution:

$$y, a = F(x) \tag{6}$$

where $a = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ such that $\sum \alpha_i = 1$ and $\alpha_i \in [0, 1]$.

We select the top k words with the highest attention weights as the substitution objects, and k is a hyper-parameter to control the substitution rate.

The attention distribution reflects the importance order of the words in the sequence. It is the refreshed parameters of the output of LSTM in each step. The attention distribution doesn't equal the saliency map, the contribution of each word to the result, but it is positively correlated between the two concepts. The words with high attention coefficient has high probability to contribute more to the prediction.

Meanwhile, the attention distribution is calculated by one forward propagation, and the adversarial example can be easily crafted without backpropagation or greedy search. The requirement of real-time is the reason why we use attention distribution to determine the substitution order.

Word Substitution Strategy. Assuming that ω_i is the substitution word in x, we use the WordNet tool to build a set of synonyms of ω_i, and we define the set as:

$$S_i = \{\omega'_{i1}, \omega'_{i2}, \ldots, \omega'_{im}\} \tag{7}$$

Where m is the number of synonyms. A substitution candidate could be a single word or a short phrase and S_{ω_i} could be empty if no synonym is found. We use two strategies to perform the substitution, the first one is Optimal Greedy Substitution and the second one is probability estimation based on attention distribution and beam search.

Optimal Greedy Substitution.
When k substitution words in a text x are determined by attention distribution, we replace the words in the order of attention weights. For one word ω_i, we try to find the synonym which makes the text change most when substitution is performed. It can be expressed as:

$$\omega_i* = \underset{\omega_i' \in S_i, y' \neq y}{\arg\max} (P(y'|x')) \tag{8}$$

Where:

$$x' = \{\omega_1, \omega_2, \ldots \omega_i' \ldots \omega_n\} \tag{9}$$

When ω_i is replaced by ω_i*, the same strategy is repeated on the next candidate word. After all the k words are substituted, an adversarial example is crafted:

$$x* = \{\omega_1, \omega_2, ..\omega_i * ..\omega_j * ..\omega_k * ..\omega_n\} \tag{10}$$

This new text sequence has a relatively high probability of a successful attack, but it's not time efficient enough because of the greedy search.

Algorithm 1

Require: Generating adversarial text x' in real time.
1. Input the original text x to the Trojan model F(self-attention BiLSTM)
2. Get the predicted label of original input: F(x)=y
3. Get the attention weight distribution of input of each word.
4. Select k words with high attention weight
5. **for** i=1 to k:
6. Choose the word with i-th highest attention weight, which is ω_i.
7. Get the substitution set $S_i = \{\forall \omega'_{ij}\}$, where ω'_{ij} are m synonyms of ω_i.
8. Combine the synonym words in as a string ss_i and input it into F.
9. Get the attention weight distribution of ss_i.
10. Select q words with high attention weight ω^*_i
11. Generate $x'(i)$ by replacing ω_i in $x'(i-1)$ with ω^*_i
12. **if** $F(x'(i)) \neq y$:
13. $x' = x'(i)$
14. **end for**
15. **end for**

Attention Based Substitution Search
This method is also based on attention distribution. When a synonym set of a word is built, we connect these synonym words into a new string and put it into the attention BiLSTM model. Then the output of substitution combination is:

$$y_s, A_s = F(S_{\omega_i}) \tag{11}$$

where $S_{\omega_i} = \{\omega'_{i1}, \omega'_{i2}, \ldots, \omega'_{im}\}$, y_s is the prediction of the model, and A_s is the attention distribution of these words. The combination of these words is meaningless but the attention distribution probably reflects the priority of their contribution to the predicted result y_s. If $y_s = y = F(x)$, the words with low attention weight are selected first to perform an attack, and if $y_s \neq y$, the words with high attention weight are prior choices.

Then we use a heuristic search to find a locally optimal solution. Every circular is a synonym of the word ω_i, and the darker color means a better attack expectation. In each first step, we select the top q synonym words for substitution. Except for the first and last steps, there are q^2 choices to replace the relative words in original input x. We select q best choice to calculate the next step. At last, one best sequence is determined and the adversarial examples are crafted. An example with q = 2 is shown in Fig. 3, and the procedure is shown in Algorithm 1.

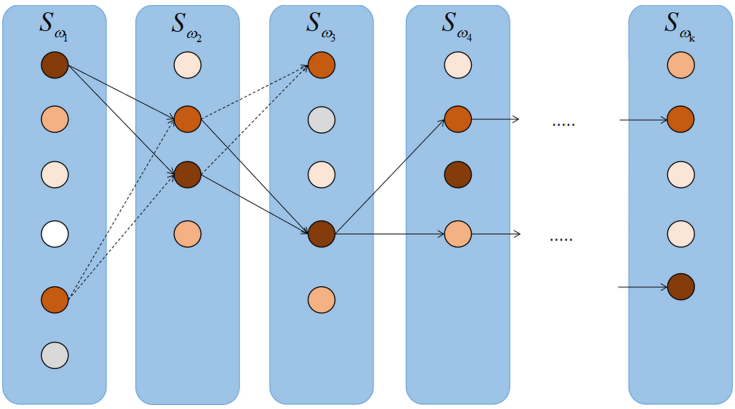

Fig. 3. Synonym Words Attention based heuristic search.

4 Experiment

4.1 DataSets and Deep Neural Models

IMDB. IMDB sentiment is a binary sentiment(positive and negative) classification dataset. It collects highly polar movie reviews from the IMDB, the world's most popular and authoritative website for movies. In this dataset, a set of 25,000 comments is provided for training, and 25,000 for testing.

Attn Bi-LSTM. The word embedding dimension is set as 200, and 128 units in BiLSTM layer. An attention layer is added to refresh the weights of output. A full connection layer with softmax is used to get the result. The framework of Attn-BiLSTM is shown in Fig. 1, and the original accuracy of Attn-BiLSTM is 90.6%.

4.2 Attention Weight V.S. Saliency Map

The saliency map is the contribution of the words to the predicted class. Previous work WS and PWWS estimated the order of importance by saliency map. Attention weight is an inner parameter of the BiLSTM, and the feature with high attention weight will contribute more to the prediction result. Figure 4 and Fig. 5 show that the saliency map and attention weight have a positive correlation.

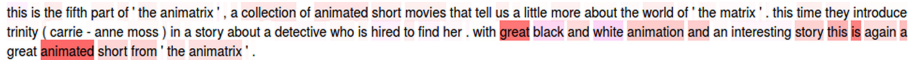

Fig. 4. The effect of Saliency Map.

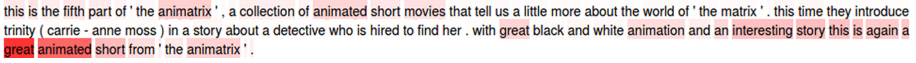

Fig. 5. The effect of Attention weight distribution.

4.3 Attacking Result

We experimented by controlling the number of substitution words. In some cases, even a text is long, a modification of one word changed the result. Examples are shown in Table 1. In 'Text information' column of this table, the back words are the original material of the dataset. The green ones are the selected words by our method and the red ones in brackets are the substitution counterparts. When the adversarial examples are input to the same model, the polarities of these texts are changed, as shown in the first two columns.

Table 1. APE attack results.

Original Result	Adversarial Result	Text information
Negative	Positive	i was very excited when paranormal state first came on a&e. i thought that it may bring some more interesting ghostly evidence . the production value looked good and i really love the logo . then , after about few episodes in , i started to feel that this show may not be looking for evidence but had a strong religious agenda.it seems like every case they investigate has some big powerful evil demon that can't even make a teacup move on camera , yet everyone is terrified . then comes some power of christ ritual that saves everyone.also , there is very little focus on other members of the team . the entire show focuses on ryan and he feels like one of those people that hands you pamphlets about his church on the street.has paranormal phenomenon and demons become the new missionaries of christianity , scaring people to convert ? really , this should be on a christian network . i was very disappointed(defeated).
Negative	Positive	Primary(chief) plot!primary direction! poor interpretation
Positive	Negative	i saw this movie(pic) last night and thought it was decent . it has it 's moments(bit) i guess you would say . some of the scenes with the special ops forces were cool , and some of the location shots were very authentic . i wo n't be putting this movie in my dvd collection but it is fair enough to recommend for renting . i guess nothing set the movie at another level compared to others of the same genre . the action is good , the acting is decent , the women are extremely seductive and exotic in my opinion , and the story is pretty(middling) interesting . 7 out of ten "

We compare four substitution method. The first one is the Random method which repla**ces** K words in a text randomly. PWWS [1] method calculate the saliency map and replace the word in the order of saliency value. APE + GS is our method attention probability estimation, and the synonyms are tested by a greedy search. APE + ABS means the Attention-based Substitution Search mentioned in Sect. 3.3. K is the max number of substitution words in a text. The results are shown in Table 2. It can be concluded that the attack effect of APE + GE is significant especially when the K is large.

Table 2. Substitution attack comparison.

Prediction accuracy	K = 1	K = 5	K = 10	K = 15	K = 20
Self-attn BiLSTM	90.6%				
Random	88.1%	84.2%	82.4%	78.8%	75.5%
PWWS	**83.4%**	72.7%	57.9%	45.0%	42.0%
APE + GS	83.8%	**63.1%**	**38.8%**	**37.1%**	**15.9%**
APE + ABS	85.3%	72.8%	65.1%	55.1%	52.3%

And the curves of more tested data are shown in Fig. 6.

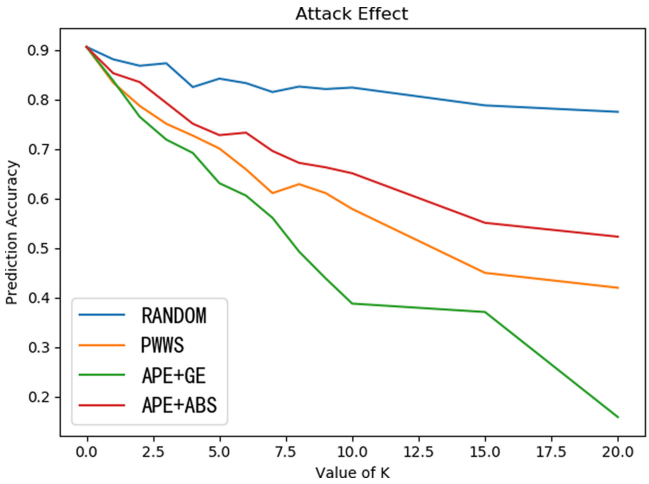

Fig. 6. The comparison of attack effect of four methods.

5 Conclusion

In this paper, we study the security issue of smart home. We design a possible trojan scenario of attacking the dialogue devices or robots at smart home, and we propose a novel method based on attention distribution to craft textual adversarial examples and an algorithm to improve the responding speed. The result shows that it works and exceeds the previous substitution method. However, the transferibility of the method should be considered and the substitution distribution could be predicted in a more precise way, these issues should be concerned in our future work.

Acknowledgement. We thank the anonymous reviewers for their insightful comments on the preliminary version of this paper. This work is supported by the Natural Science Foundation of Guangdong Province (Grant No. 2018A030313354). Any findings, opinions, or conclusions in this paper are those of the authors and do not reflect the views of the funding agency.

References

1. Ren, S., Yihe, D., Kun, H., Che, W.: Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. ACL (2019)
2. Copos, B., Levitt, K., Bishop, M., Rowe, J.: Is Anybody Home?. Inferring Activity From Smart Home Network Traffic. Security & Privacy Workshops, IEEE (2016)
3. Kwabena, O.-A., Qin, Z., Zhuang, T., Qin, Z.: MSCryptoNet: Multi-Scheme privacy-preserving deep learning in cloud computing. IEEE Access **7**, 29344–29354 (2019)
4. Awni, H., Case, C., Casper, J., Catanzaro, B., Diamos, G.: Deep speech: scaling up end-to-end speech recognition. Computer Science (2014)
5. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J.: Intriguing properties of neural networks. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014) (2014)
6. Jia, R., Liang. P.: Adversarial examples for evaluating reading comprehension systems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). Copenhagen, Denmark, pp. 2021–2031 (2017)
7. Jin, Y., Kupp, N., Makris, Y.: Experiences in Hardware Trojan Design and Implementation. In: IEEE International Workshop on Hardware-oriented Security & Trust. IEEE (2009)
8. Ian, J.: Goodfellow, Jonathon Shlens, Christian Szegedy. Explaining and Harnessing Adversarial Examples (2015). https://arxiv.org/abs/1412.6572
9. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: Proceedings of IEEE European Symposium on Security and Privacy (2016)
10. Carlini, N., Wagner, D.: Towards Evaluating the Robustness of Neural Networks (2016)
11. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP) (2016)
12. Liang, B., Li, H., Su, M., Bian, P., Li, X., Shi, W.: Deep Text Classification Can be Fooled. arXiv preprint arXiv:1704.08006 (2017)
13. Xue, M., Yuan, C., Wang, J., Liu, W.: DPAEG: a dependency parse-based adversarial examples generation method for intelligent Q&A robots. In: Security and Communication Networks 2020 (2020): 5890820:1-5890820:15
14. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
15. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
16. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)