Digital Object Identifier

Friend-Guard Textfooler Attack on Text Classification System

HYUN KWON[1]
[1]Department of Electrical Engineering, Korea Military Academy, Seoul 01819, South Korea
Corresponding author: Hyun Kwon (e-mail: hkwon.cs@gmail.com or khkh@kaist.ac.kr).

This study was supported by National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (2021R1I1A1A01040308) and Hwarang-Dae Research Institute of Korea Military Academy.

ABSTRACT Deep neural networks provide good performance for image classification, text classification, speech classification, and pattern analysis. However, such neural networks are vulnerable to adversarial examples. An adversarial example is a sample created by adding a little noise to the original sample data and that, although presenting no change identifiable to human perception, will be misclassified by a deep neural network. Most studies on adversarial examples have focused on images, but research is expanding to include the field of text. Textual adversarial examples can be useful in certain situations, such as when models of both friend and enemy coexist, as in a military scenario. Here, a specific message may be generated as an adversarial example such that no grammatical or semantic problems are apparent to human perception and it will be correctly classified by the friend model but incorrectly classified by the enemy model. In this paper, I propose a “friend-guard” textual adversarial example for a text classification system. Unlike the existing methods for generating image adversarial examples, the proposed method creates adversarial examples designed to be misclassified by an enemy model and correctly classified by a friend model while retaining the meaning and grammar of the original sentence by replacing words of importance with substitutions. Experiments were conducted using a movie review dataset and the TensorFlow library. The experimental results show that the proposed method can generate an adversarial example that will be correctly classified with 88.2% accuracy by the friend model and 26.1% accuracy by the enemy model.

INDEX TERMS Machine learning, Text classification, Text adversarial example, Evasion Attack, Deep neural network (DNN).

I. INTRODUCTION

Deep neural networks [1] provide good performance in image recognition [2], text recognition [3], speech recognition [4], and pattern recognition [5]. However, these neural networks are vulnerable to adversarial examples [6]. An adversarial example is a sample created by applying a small perturbation to original sample data in such a way that it will be perceived as normal by humans but will be incorrectly classified by the target model. Adversarial examples pose a serious threat to self-driving vehicles and medical businesses.

The study of adversarial examples has focused mainly on their use in the field of computer vision. Recently, however, research on adversarial examples [7] has been expanding into the text domain. In the image domain, adversarial examples are typically generated using gradient-based methods, but in the field of text, adversarial examples are generated using the word-wise method or the generative adversarial net method.

Adversarial examples that attack only specific models in the text domain can be useful in certain situations. For example, when an enemy model and a friend model coexist, a scenario that an army may face, textual adversarial examples that will be correctly classified by the friend model and misclassified by the enemy model could be useful. In such an environment, if it is necessary to send espionage data with important secret content such as information related to nuclear armaments, and the messages will be received (or intercepted) by a deep learning model that automatically classifies espionage-related text content, “friend-guard” adversarial texts can be constructed, which are designed to be misclassified by the enemy model but correctly classified by the friend model. In this paper, I propose a friend-guard textfooler method designed for text classification systems. A textual adversarial example created using the proposed method is a sample that is designed to be correctly classified by a friend model and incorrectly classified by an enemy model while displaying no obvious grammatical or content abnormalities. By replacing important elements in sentences with other similar language elements using a word-wise method, the adversarial example is created without introducing any changes in terms of grammar or meaning. The contributions of this paper are as follows.

- The method developed in this study produces a friend-guard adversarial example in the text domain that can be correctly classified by a friend model and misclassified by an enemy model. In this paper, the principle and structure of the proposed method are systematically explained.
- The average difference between the original sample and the proposed textual adversarial example was analyzed in terms of accuracy of classification by the enemy and friend models.
- An experiment was conducted using the BERT model [8], the latest text classification system. The performance of the proposed method was evaluated using a movie review (MR) dataset [9].

The remainder of this paper is organized as follows. In Section II, studies related to the proposed method are described. In Section III, the problem addressed by the proposed method is defined. Section IV describes the proposed method, and Section V describes the experiment and evaluation. Section VI discusses the results. Finally, Section VII concludes the paper.

II. RELATED WORK

Adversarial examples were first proposed by Szegedy et al. [6]. This section discusses the BERT model, information available about the target model, the recognition target, the distortion metric, and the method of attack in the text domain.

A. BERT MODEL

Bidirectional encoder representations from transformers (BERT) [8] is a natural language processing neural network structure developed by Google. It is formed by stacking encoders in a structure called a transformer. With this structure, BERT trains on the task of predicting the next sentence and predicting hidden words in the sentence. A BERT model that has been trained in this way is characterized by high speed, making it well suited for solving other special natural language processing problems [10]. The core of the transformer in BERT is the self-attention algorithm. This algorithm obtains contextual information all at one time without processing tokens from one end of the sentence to the other, for bidirectional training. In addition, self-attention can process relationship information between one word and the next word.

The self-attention process is as follows. For each input token, a vector consisting of a query, key, and value, which can be trained independently, is allocated. First, the query of one token and the key value of the other tokens are multiplied. These products are divided by 8. After that, the score value divided by 8 allocated for each token is changed probabilistically through softmax so that all sums become equal to 1. The probability value obtained through softmax is multiplied by the value of each token. The result of attention to one token is the sum of all the multiplied values and the probability obtained from the result calculated above. It becomes self-attention with one head by repeating the above procedure for each token. BERT performs multi-headed self-attention to process contextual information in various dimensions, calculating repeatedly by increasing several queries, keys, and values.

BERT uses three techniques to process the input. The first, the wordpiece method, is used for tokenization, dividing each word into parts that appear frequently in each word. The second is segment embedding, which takes two sentences as input. To separate the two sentences, their order is classified using the special classification token (CLS) value. Third, positional embedding is necessary to inform BERT of the position of each token when BERT is being trained.

BERT's output is a prediction of the next sentence. BERT uses the CLS, the first token of the first sentence, to determine which sentence will come as the second sentence in order to predict the next sentence. In addition, BERT predicts hidden tokens. For this, each token of BERT except the CLS token at the front passes through a separate fully connected layer and a softmax layer, and a word is selected from approximately 30,000 vocabulary lists used for tokenization. For the hidden token, it outputs a vector with the appropriate number of dimensions and the token having the highest probability.

B. TARGET MODEL INFORMATION

Adversarial examples are classified as white box attacks or black box attacks according to the information available about the target model. A white box attack [11] [12] [13] is an attack that creates an adversarial example in a scenario in which all details of the various structures, parameters, and results of the model are known. A black box attack [14] [15] [16] is one that creates an adversarial example in a scenario in which only the result value for the input value is given, with no information available about the model itself.

C. TYPE OF RECOGNITION TARGET

Adversarial examples are divided into targeted attacks [17] [18] and untargeted attacks [19] according to the recognition target. A targeted attack is an adversarial example that is designed to be misclassified by the model as a specific class determined by the attacker. An untargeted attack is an adversarial example that is designed to be incorrectly classified by the model as any class other than the original class. A targeted attack is a more sophisticated attack than an untargeted one.

D. DISTORTION MEASURE

In the image domain, adversarial examples [20] [21] [22] [23] [24] [25] are created by adding minimal noise to the original data as a whole in pixel units so that no problem arises in terms of human detectability but they will be incorrectly classified by the model. In the text domain, however, adversarial examples are created by changing a specific word so that there is no resulting change in the meaning to a human but they will be misclassified by the text classification model. Therefore, the distortion of an adversarial example in the text domain is measured using the number of words changed between the adversarial example sentence and the original sentence.

E. TEXT DOMAIN METHOD OF ADVERSARIAL EXAMPLE ATTACK

Adversarial examples have been studied primarily in the image domain. Most adversarial example generation methods in the image domain use gradient descent to add noise to the input data. Zhao et al. [26] proposed a method for generating adversarial examples in the text domain using the generative adversarial network (GAN) method. This method regenerates adversarial sentences similar to their original sentences after changing the latent representation of the input data. Ebrahimi et al. [7] proposed a method for generating hostile samples by changing specific words at the word level with a white box attack. This method attacked the CharCNN-LSTM model [27] by changing one word, thereby generating an adversarial example. However, the method's attack success rate is somewhat low because it chooses the word to be replaced randomly rather than according to word importance, and grammar is not considered. Jin et al. [28] proposed a method for generating adversarial examples having the same grammatical meaning by using the word-wise method. In this method, after the importance of each word in the sentence is analyzed, one word is changed to a similar word so that it creates no problem in terms of human detectability but generates an adversarial example that will be incorrectly classified by the model. However, in the existing studies of adversarial examples in the text domain, only one model was considered as an attack target. In the case of multiple models, when the model to be protected and the model to be attacked coexist, it may be necessary to attack only a specific model. The proposed method applies the Jin method [28] to create a friend-guard adversarial example.

III. CONCEPTUAL BASIS FOR PROPOSED SCHEME

Figure 1 shows decision boundaries of the enemy model and the friend model for an original sample x and an adversarial example x^*. If a sample is within the decision boundary of the model, it will be correctly classified by the model, and if it is outside the boundary, it will be incorrectly classified. The proposed method creates textual adversarial examples that are within the decision boundary of the friend model and outside the decision boundary of the enemy model and minimizes the distortion from their corresponding original samples. In addition, in the proposed method, an adversarial example is generated using word-wise in the text domain; it is a sentence that is unchanged from the original version in terms of meaning and grammar. In the figure, original sample x is the sentence "The script is smart, not cloying", and adversarial example x^* is the sentence "The script is canny, not cloying". Both sentences have the same meaning to a human, and there are no grammatical flaws. However, the adversarial example, "The script is canny, not cloying", will be incorrectly classified as a negative sentence by the enemy model and will be properly classified as a positive sentence by the friend model.

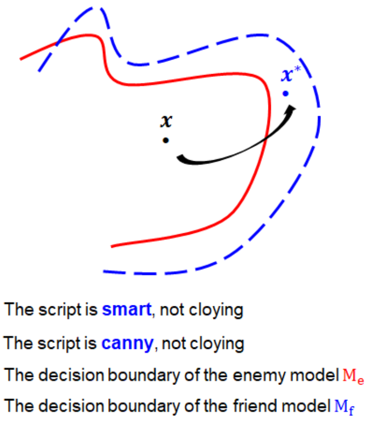
x^*
x
x The script is smart, not cloying
x^* The script is canny, not cloying
The decision boundary of the enemy model $\mathrm{M_e}$
The decision boundary of the friend model $\mathrm{M_f}$

FIGURE 1: Decision boundaries of the enemy model and friend model for an original sample x and adversarial example x^*.

IV. PROPOSED SCHEME

A. ASSUMPTION

For the proposed method to work, the generator needs to know the confidence score for the input data for the enemy and friend models. It is possible to generate adversarial examples without knowing the parameters and structures of the enemy and friend models. Under this assumption, the proposed method can generate textual adversarial examples that are not abnormal to human perception and will be incorrectly classified by the enemy model and correctly classified by the friend model.

B. PROPOSED METHOD

Figure 2 shows an overview of the proposed scheme. The proposed method has two main steps: word importance ranking and word transformation. First, through word importance ranking (WIR), the words that have a significant influence on the model's prediction are ranked in order. The second step, word transformation, consists of three parts as follows. Synonym extraction (SE) collects candidate words that can be substituted, in order by the priority of the words in the WIR. Next, the word candidates are subjected to a part-of-speech (POS) check [29], which ensures that certain words that affect the grammar of the sentence remain unchanged. Then, from among the candidate words, a candidate group capable of maintaining the highest similarity to the original sample is found through a semantic similarity check (SSC) [30]. After these two main steps, an adversarial example is created by replacing a word from among the remaining candidate words, and it is correctly classified by the friend model and incorrectly classified by the enemy model and has the highest similarity to its corresponding original sample. If

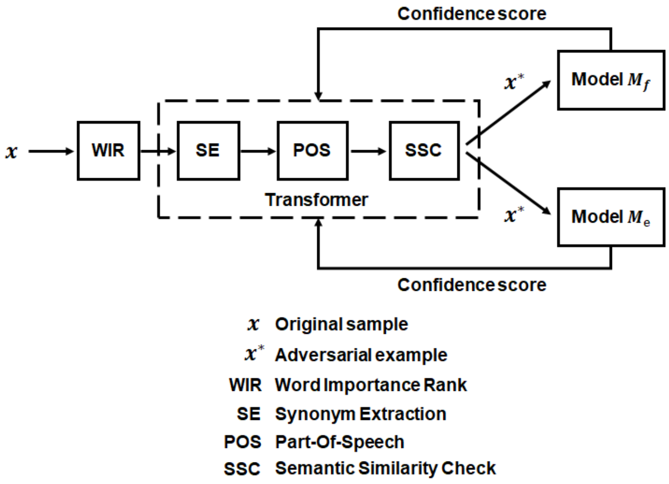

FIGURE 2: The overview of the proposed scheme.

the appropriate confidence score is low, the process moves to the next selected word and repeats the above steps.

The above procedure can be expressed mathematically as follows. First, in WIR, given a sentence $X = \{x_1, x_2, ..., x_n\}$ consisting of n words, the proposed method needs to find some key words that have the most influence on the prediction models, M_e and M_f. Therefore, the selection mechanism that most affects the change in the last predicted result is used. In addition, the semantic similarity should be kept as strong as possible, while changes due to the selection process are kept to a minimum. After the word x_i is deleted from $X = \{x_1, x_2, ..., x_n\}$, the confidence value is calculated by determining the difference between the prediction score and the result values entered into models M_e and M_f. The importance score S_{x_i} is calculated for the prediction change before and after the word change. When words are being ranked by importance, words such as "the" and "it" are filtered out so as not to disrupt the grammar.

Second, given a word of high importance I_{x_i} in the word transformer, a step is needed to apply a replacement mechanism for that word. To find the most suitable word substitution for generating friend-guard adversarial examples, three steps are required—SE, POS, and SSC—and conditions need to be identified so that the adversarial examples will be correctly classified by the friend model and misclassified by the enemy model.

In SE, all possible substitute candidate sets for the selected word x_i are collected. Let the candidates be the N synonyms having the closest cosine similarity to the word x_i ($\geq \epsilon$). Word embedding is used to express this word. The embedding vector is used to identify N synonyms having cosine similarity greater than the value of ϵ. In this study, N was set to 50 and ϵ was set to 0.7, to control diversity and semantic similarity.

The POS check of the candidates for the word x_i is performed because it is necessary to use a word with the same part of speech (POS) to maintain the grammatical character of the text.

Algorithm 1 Friend-guard textfooler generation

```
Input: Original sample $X = \{x_1, x_2, ..., x_n\}$, the original
    label $Y$, friend model $M_f$, enemy model $M_e$, sentence
    similarity function $s(\cdot)$, cosine similarity $s_c(\cdot)$, threshold
    $\epsilon$, word embedding $e$ over the vocabulary $V$, final candidate set $C_{final}$
Friend-guard textfooler:
    $C_{final} \leftarrow \{\}$ , $X_{adv} \leftarrow X$
    for each word $x_i$ in $X$ do
        Compute the importance score $I_{x_i}$
    end for
    Create a set $X_I$ of all words $x_i \in X$ using the importance
    score $I_{x_i}$.
    Filtering the stop words in $X_I$
    for each word $x_i$ in $X_I$ do
        Initiate the set of candidates $C$ by extracting the top
    $N$ synonyms using $s_c(e_{x_i}, e_{word})$ for each word in $V$.
        $C \leftarrow$ POS-checking($C$)
        for $c_j$ in $C$ do
            $X^* \leftarrow$ Replace $x_i$ with $c_j$ in $X_{adv}$
            if $s(X^*, X_{adv}) \geq \epsilon$ then
                $C_{final} \leftarrow$ append($c_j$)
                $Y_f \leftarrow M_f(X^*)$ and $Y_e \leftarrow M_e(X^*)$
            end if
        end for
        if there exist $c_j$ such that $Y_f = Y$ and $Y_e \neq Y$ then
            $c^* \leftarrow \mathrm{argmax}_{c \in C_f} S(X, X^*_{x_i \to c})$
            $X_{adv} \leftarrow$ Replace $x_i$ with $c^*$ in $X_{adv}$
            return $X_{adv}$
        end if
    end for
```

For the remaining candidates, SSC replaces the word x_i in the sentence and creates an adversarial example. The generated adversarial example is then provided to models

M_f and M_e to obtain a prediction score. Using the universal sentence encoder (USE), the semantic similarity is calculated using a high-dimensionality vector of sentence similarity and the cosine similarity score for the original sample and the adversarial example. If the semantic similarity is higher than the specified ϵ value, it is stored in the pool of final candidates.

In generating adversarial examples, the one with the highest similarity score among the final candidates is selected. If no final candidate exists, the SE, POS, and SSC steps are repeated as above for the next selected word. The details of this procedure for generating an adversarial example are given in Algorithm 1.

V. EXPERIMENT AND EVALUATION

Experiments were conducted to assess whether the proposed method can generate friend-guard adversarial examples for a text classification system. The experiments used the TensorFlow library [31], widely used for machine learning, and an Intel(R) i5-7100 3.90-GHz server.

A. EXPERIMENTAL SETUP

For the experiment, the MR dataset [9], which is a movie review file, was used. It is a dataset that categorizes emotions into positive and negative emotions at the sentence level. 9000 data were used for training and 1000 data for testing.

The friend and enemy text classification models consisted of the BERT model. There were 12 hidden layers, and the number of nodes was 768. The maximum number of position embeddings was 512, and the vocabulary size was 30,522. The intermediate size was 3072, and gelu [32] was used for hidden activation. To configure the friend and enemy models to be distinct, each model was trained using different parameters, shown in Table 1 of the appendix. The friend and enemy models had 85.4% and 85.7% accuracy, respectively, on the test data after training on the original training data.

For generating the proposed adversarial examples, the similarity score threshold was set to 0.7, the number of synonyms was set to 50, the batch size was set to 32, and the maximum sequence length was set to 128. The performance of the proposed method was evaluated by generating 500 adversarial examples as random test data.

B. EXPERIMENTAL RESULTS

Figure 3 shows three examples of sentence pairs, each pair consisting of an original sentence and a proposed friend-guard adversarial sentence for the friend model M_f and the enemy model M_e.

To human perception, each friend-guard adversarial sentence, formed by replacing a specific word in the original sentence with a different word, has the same meaning as the original sentence and has no grammatical errors. In terms of model classification, the friend-guard sentences are misclassified by the enemy model M_e and correctly classified by the friend model M_f. The classification results of the friend and enemy models for additional original and proposed sentence pairs are given in the appendix.

#1_Original sentence (M_f and M_e: negative): "no telegraphing is too obvious or **simplistic** for this movie"
#1_Proposed sentence (M_f: negative, M_e: positive): "no telegraphing is too obvious or **uncomplicated** for this movie"

#2_Original sentence (M_f and M_e: positive): "norton is **magnetic** as graham"
#2_Proposed sentence (M_f: positive, M_e: negative): "norton is **swipe** as graham"

#3_Original sentence (M_f and M_e: positive): "a **small** movie with a big impact"
#3_Proposed sentence (M_f: positive, M_e: negative): "a **fewer** movie with a big impact"

FIGURE 3: Three sentence pair examples: original sentence and proposed sentence for M_f and M_e.

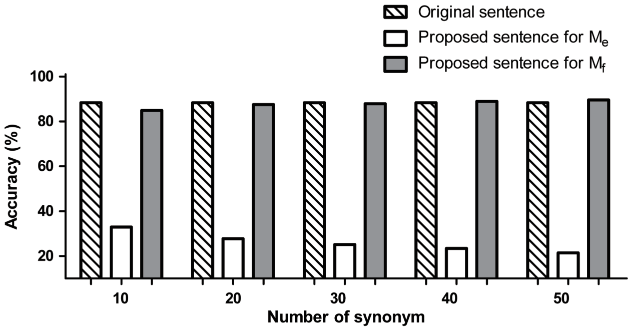

FIGURE 4: Accuracy for original samples and proposed adversarial examples by friend model M_f and enemy model M_e according to the number of synonyms.

Figure 4 shows the accuracy for the original samples and proposed adversarial examples of the friend model M_f and the enemy model M_e according to the number of synonyms. The accuracy shown for the original samples, 88.2%, is the average of the friend model M_f accuracy and the enemy model M_e accuracy. For the friend-guard adversarial examples, the proposed sentence, intended to be misclassified by the enemy model M_e, was classified correctly by this model with an average accuracy of 26.1%. Using the friend model, however, the proposed sentence was classified correctly with an average accuracy of 87.8%. As the number of synonyms increased, the accuracy of classification by the enemy model decreased and that of the friend model increased slightly. Therefore, the proposed adversarial sentence is generally classified correctly by the friend model and generally classified incorrectly by the enemy model.

Figure 5 shows the average change and the number of queries for the proposed adversarial example according to the number of synonyms. The figure shows that as the

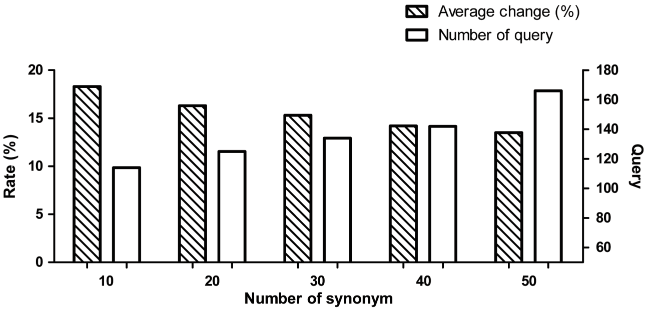

FIGURE 5: Average change and number of queries for the proposed adversarial example according to the number of synonyms.

number of synonyms increased, the average percent change decreased and the number of queries increased. A query of the model is required for each synonym; therefore, when the number of synonyms increases, the number of queries increases proportionally. Regarding the average change, as the number of synonyms increases, the number of cases that can be changed increases, and because various attacks are possible, the average change decreases slightly to maximize the similarity between the original sample and the proposed adversarial example.

VI. DISCUSSION

This section discusses the proposed method in terms of attack considerations, percent change, the number of synonyms, human perception, and limitations.

Attack consideration. The proposed method requires knowledge of the confidence scores for the friend and enemy models. Using these confidence scores for the input text sentence, an adversarial example (proposed sentence) is generated that is correctly classified by the friend model and incorrectly classified by the enemy model. Because the proposed method determines whether an alternative phrasing is suitable by examining the confidence score, access to the models' confidence scores is required. Although the friend-guard concept has been applied in the fields of image [33] [34] and audio [35], this paper is the first to propose it for the text domain, in which the construction of a method is more challenging owing to the need for a word-by-word approach.

Change The proposed method generates friend-guard adversarial examples by changing words that have a significant influence as determined by the word-wise method. It is characterized by its capability of generating proposed adversarial examples with few word changes, which is made possible by ranking the words by their degree of influence in the sentence and changing the words having the highest rank.

The number of Synonyms Because a proposed adversarial example is generated by replacing specific words word-wise, the performance of the proposed method varies according to the number of words to be replaced. The number of synonyms affects the number of queries required, the percent change, and the accuracy of the enemy and friend models on the proposed adversarial example, and hence the performance. As the number of synonyms increases, the number of replaceable words increases. Therefore, the accuracy of the friend model increases and the accuracy of the enemy model decreases for the proposed adversarial example. As the number of synonyms increases, the number of queries required increases, but the percent change decreases.

The human perception. The proposed adversarial example should have the same meaning as the original sample in terms of human recognition and should have no grammatical errors. Methods such as POS were used so as not to change words that contribute strongly to basic grammatical structure. To preserve the meaning of the sentence, a word is replaced with a noun having many synonyms; in addition, the number of words changed is kept low to minimize the difference in meaning between the proposed adversarial example and the original sample.

Limitations. Because the proposed method uses the word-wise method, generation of proposed adversarial examples may be limited if there is no appropriate alternative word that can be classified incorrectly by the enemy model and correctly by the friend model.

VII. CONCLUSIONS

In this paper, I have proposed a method for generating friend-guard adversarial examples in the text domain. The method creates friend-guard adversarial examples that will be correctly classified by the friend model and misclassified by the enemy model without introducing any changes in meaning or grammar that will be perceived by humans. It works by replacing words of high importance with synonyms, unlike the approach used in image studies. In the experiment, the proposed method generated friend-guard adversarial examples that were correctly classified with 88.2% accuracy by the friend model and 26.1% accuracy by the enemy model.

For future research, it might be interesting to develop a technique for generating an adversarial example based on a newer method, such as generative adversarial networks (GANs) [36]. Development of a defense against the proposed method would be another interesting topic for research.

REFERENCES

[1] Jürgen Schmidhuber. Deep learning in neural networks: An overview. Neural networks, 61:85–117, 2015.
[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations, 2015.
[3] Sasanka Potluri and Christian Diedrich. Accelerated deep neural networks for enhanced intrusion detection

system. In Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on, pages 1–8. IEEE, 2016.

[4] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29 (6):82–97, 2012.

[5] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. Nature, 529 (7587):484–489, 2016.

[6] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In International Conference on Learning Representations, 2014.

[7] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. arXiv preprint arXiv:1712.06751, 2017.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[9] Vuk Batanović, Boško Nikolić, and Milan Milosavljević. Reliable baselines for sentiment analysis in resource-limited languages: The serbian movie review dataset. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 2688–2696, 2016.

[10] Stéphane Meystre and Peter J Haug. Natural language processing to extract medical problems from electronic clinical documents: performance evaluation. Journal of biomedical informatics, 39(6):589–599, 2006.

[11] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In 2016 IEEE European symposium on security and privacy (EuroS&P), pages 372–387. IEEE, 2016.

[12] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In Security and Privacy (SP), 2017 IEEE Symposium on, pages 39–57. IEEE, 2017.

[13] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In International Conference on Learning Representations, 2015.

[14] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pages 506–519. ACM, 2017.

[15] Hyun Kwon, Yongchul Kim, Ki-Woong Park, Hyunsoo Yoon, and Daeseon Choi. Advanced ensemble adversarial example on unknown deep neural network classifiers. IEICE TRANSACTIONS on Information and Systems, 101(10):2485–2500, 2018.

[16] Jie Li, Rongrong Ji, Hong Liu, Xiaopeng Hong, Yue Gao, and Qi Tian. Universal perturbation attack against image retrieval. In Proceedings of the IEEE International Conference on Computer Vision, pages 4899–4908, 2019.

[17] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In 2018 IEEE Security and Privacy Workshops (SPW), pages 1–7. IEEE, 2018.

[18] Bo Luo, Yannan Liu, Lingxiao Wei, and Qiang Xu. Towards imperceptible and robust adversarial example attacks against neural networks. arXiv preprint arXiv:1801.04693, 2018.

[19] Aming Wu, Yahong Han, Quanxin Zhang, and Xiaohui Kuang. Untargeted adversarial attack via expanding the semantic gap. In 2019 IEEE International Conference on Multimedia and Expo (ICME), pages 514–519. IEEE, 2019.

[20] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. arXiv preprint arXiv:1709.04114, 2017.

[21] Yash Sharma and Pin-Yu Chen. Attacking the madry defense model with l_1-based adversarial examples. arXiv preprint arXiv:1710.10733, 2017.

[22] Tianyun Zhang, Sijia Liu, Yanzhi Wang, and Makan Fardad. Generation of low distortion adversarial attacks via convex programming. In 2019 IEEE International Conference on Data Mining (ICDM), pages 1486–1491. IEEE, 2019.

[23] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4322–4330, 2019.

[24] Pu Zhao, Kaidi Xu, Sijia Liu, Yanzhi Wang, and Xue Lin. Admm attack: an enhanced adversarial attack for deep neural networks with undetectable distortions. In Proceedings of the 24th Asia and South Pacific Design Automation Conference, pages 499–505, 2019.

[25] Deyan Petrov and Timothy M Hospedales. Measuring the transferability of adversarial examples. arXiv preprint arXiv:1907.06291, 2019.

[26] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. arXiv preprint arXiv:1710.11342, 2017.

[27] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. arXiv preprint arXiv:1508.06615, 2015.

[28] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. arXiv preprint arXiv:1907.11932, 2, 2019.

[29] Jerome R Bellegarda. Combined statistical and rule-based part-of-speech tagging for text-to-speech synthesis, May 6 2014. US Patent 8,719,006.

[30] James O'shea, Zuhair Bandar, and Keeley Crockett. A new benchmark dataset with production methodology for short text semantic similarity algorithms. ACM Transactions on Speech and Language Processing (TSLP), 10(4):1–63, 2014.

[31] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In OSDI, volume 16, pages 265–283, 2016.

[32] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. arXiv preprint arXiv:1710.05941, 2017.

[33] Hyun Kwon, Yongchul Kim, Ki-Woong Park, Hyunsoo Yoon, and Daeseon Choi. Friend-safe evasion attack: An adversarial example that is correctly recognized by a friendly classifier. computers & security, 78:380–397, 2018.

[34] Hyun Kwon, Hyunsoo Yoon, and Ki-Woong Park. Friendnet backdoor: Indentifying backdoor attack that is safe for friendly deep neural network. In Proceedings of the 3rd International Conference on Software Engineering and Information Management, pages 53–57, 2020.

[35] Hyun Kwon, Yongchul Kim, Hyunsoo Yoon, and Daeseon Choi. Selective audio adversarial example in evasion attack on speech recognition system. IEEE Transactions on Information Forensics and Security, 15: 526–538, 2019.

[36] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks. arXiv preprint arXiv:1703.01340, 2017.

APPENDIX

TABLE 1: The parameter for the friend model M_f and the enemy model M_e.

Description	Friend model	Enemy model
Attention dropout	0.1	0.15
Hidden dropout	0.01 s	0.02
Initial constant	0.02 s	0.01

The fifteen cases: the original sentence and the proposed sentence for the friend model M_f and the enemy model M_e.

#1_Original sentence (M_f and M_e: positive): intriguing and stylish

#1_Proposed sentence (M_f: positive, M_e: negative): confusing and stylish

#2_Original sentence (M_f and M_e: positive): an old fashioned scary movie , one that relies on lingering terror punctuated by sudden shocks and not constant bloodshed punctuated by flying guts

#2_Proposed sentence (M_f: positive, M_e: negative): an old fashioned scary movie , one that relies on being terror punctuated by sudden shocks and not constant bloodshed punctuated by flying guts

#3_Original sentence (M_f and M_e: positive): a solid , well formed satire

#3_Proposed sentence (M_f: positive, M_e: negative): a solid , well formed ridicule

#4_Original sentence (M_f and M_e: positive): if a horror movie 's primary goal is to frighten and disturb , then they works spectacularly well a shiver inducing , nerve rattling ride

#4_Proposed sentence (M_f: positive, M_e: negative): if a horror movie 's primary goal is to frighten and disturb , then they works ridiculously well a shiver inducing , nerve creaking ride

#5_Original sentence (M_f and M_e: positive): a delirious celebration of the female orgasm

#5_Proposed sentence (M_f: positive, M_e: negative): a wishful celebration of the female bulge

#6_Original sentence (M_f and M_e: positive): smart science fiction for grown ups , with only a few false steps along the way

#6_Proposed sentence (M_f: positive, M_e: negative): lustrous science fiction for grown ups , with only a few false steps along the way

#7_Original sentence (M_f and M_e: negative): a puzzling experience

#7_Proposed sentence (M_f: negative, M_e: positive): a shocking experience

#8_Original sentence (M_f and M_e: negative): most of the dialogue made me want to pack raw dough in my ears

#8_Proposed sentence (M_f: negative, M_e: positive): most of the dialogue made me liked to posse raw dough in my anklets

#9_Original sentence (M_f and M_e: positive): a burst of color , music , and dance that only the most practiced curmudgeon could fail to crack a smile at

#9_Proposed sentence (M_f: positive, M_e: negative): a implosion of color , music , and dance that only the most practiced curmudgeon could fail to crack a smile at

#10_Original sentence (M_f and M_e: positive): hashiguchi vividly captures the way young japanese live now , chafing against their culture 's manic mix of millennial brusqueness and undying , traditional politesse

#10_Proposed sentence (M_f: positive, M_e: negative): hashiguchi ridiculously captures the way young japanese live now , chafing against their culture 's manic mix of millennial brusqueness and undying , traditional politesse

#11_Original sentence (M_f and M_e: positive): a portrait of hell so shattering it 's impossible to shake

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3080680, IEEE Access

#11_Proposed sentence (M_f: positive, M_e: negative): a spitting of hell so implosion it 's impossible to shake

#12_Original sentence (M_f and M_e: negative): a farce of a parody of a comedy of a premise , it is n't a comparison to reality so much as it is a commentary about our knowledge of films

#12_Proposed sentence (M_f: negative, M_e: positive): a farce of a parody of a comedy of a scenario , it is n't a comparison to reality so importantly as it is a commentary about our expertise of films

#13_Original sentence (M_f and M_e: negative): yes they can swim , the title is merely anne sophie birot 's off handed way of saying girls find adolescence difficult to wade through

#13_Proposed sentence (M_f: negative, M_e: positive): yes they can swim , the title is scarcely anne sophie birot 's off handed way of saying girls find adolescence difficult to wade through

#14_Original sentence (M_f and M_e: negative): while the film shuns the glamour or glitz that an american movie might demand , scherfig tosses us a romantic scenario that is just as simplistic as a hollywood production

#14_Proposed sentence (M_f: negative, M_e: positive): while the film shuns the glamour or glitz that an american movie might demand , scherfig tosses us a romantic scenario that is just as uncomplicated as a hollywood production

#15_Original sentence (M_f and M_e: negative): the story is predictable , the jokes are typical sandler fare , and the romance with ryder is puzzling

#15_Proposed sentence (M_f: negative, M_e: positive): the story is predictable , the jokes are typical sandler fare , and the romance with ryder is heartrending

•••

HYUN KWON received the B.S degree in mathematics from Korea Military Academy, South Korea, in 2010. He also received the M.S. degree in School of Computing from Korea Advanced Institute of Science and Technology (KAIST) in 2015, and the Ph.D. degree at School of Computing, KAIST in 2020. He is currently an assistant professor in Korea Military Academy. His research interests include information security, computer security, and intrusion tolerant system.