Generalization to Mitigate Synonym Substitution Attacks

Basemah Alshemali
College of Computer Science and Engineering
Taibah University
Almadinah, KSA
College of Engineering and Applied Science
University of Colorado at Colorado Springs
Colorado Springs, USA
balshema@uccs.edu

Jugal Kalita
College of Engineering and Applied Science
University of Colorado at Colorado Springs
Colorado Springs, USA
jkalita@uccs.edu

Abstract

Studies have shown that deep neural networks are vulnerable to adversarial examples – perturbed inputs that cause DNN-based models to produce incorrect results. One robust adversarial attack in the NLP domain is the synonym substitution. In attacks of this variety, the adversary substitutes words with synonyms. Since synonym substitution perturbations aim to satisfy all lexical, grammatical, and semantic constraints, they are difficult to detect with automatic syntax check as well as by humans. In this work, we propose the first defensive method to mitigate synonym substitution perturbations that can improve the robustness of DNNs with both clean and adversarial data. We improve the generalization of DNN-based classifiers by replacing the embeddings of the important words in the input samples with the average of their synonyms' embeddings. By doing so, we reduce model sensitivity to particular words in the input samples. Our algorithm is generic enough to be applied in any NLP domain and to any model trained on any natural language.

1 Introduction

Deep Neural Networks (DNNs) have achieved remarkable success in various machine learning tasks, including computer vision (Krizhevsky et al., 2012; He et al., 2019), speech recognition (Hinton et al., 2012; Chen et al., 2019), and natural language processing (NLP) (Kim, 2014; Pirinen, 2019; Kambhatla et al., 2018). However, studies have found that DNNs are vulnerable to adversarial examples – artificially modified input samples that lead DNNs to produce incorrect results, while not being detectable by humans (Szegedy et al., 2014). These vulnerabilities have been exposed in the domains of computer vision (Goodfellow et al., 2015; Papernot et al., 2016; Carlini and Wagner, 2017), speech (Alzantot et al., 2017; Carlini and Wagner, 2018), and NLP (Ebrahimi et al., 2018; Jin et al., 2020).

Based on the adversary's level of perturbation, three categories of adversarial attacks in NLP systems have been proposed: Character-level, token-level, and sentence-level adversarial attacks (Alshemali and Kalita, 2020; Zhang et al., 2020). One robust existing token-level adversarial attack in NLP is black-box synonym substitution (Alzantot et al., 2018; Ren et al., 2019; Zhang et al., 2019; Jin et al., 2020). In attacks of this variety, the adversary substitutes tokens with synonyms. Since synonym substitution perturbations aim to satisfy all lexical, grammatical, and semantic constraints, they are difficult to detect with automatic syntax check as well as by humans.

In this work, we propose a defensive method to mitigate synonym substitution perturbations. We propose to improve the generalization of DNN-based models by replacing the embeddings of the important tokens in the input samples with the average of their synonyms' embeddings. By doing so, we reduce model sensitivity to particular tokens in the input samples. Experimenting on two popular datasets, for two types of text classification tasks, demonstrates that the proposed defense is not only capable of defending against these adversarial attacks, but is also capable of improving the performance of DNN-based models when tested on benign data. To our knowledge, our defense is the first proposed method that can effectively (1) Improve the robustness of DNN-based models against synonym substitution adversarial attacks and (2) Improve the generalization of DNN-based models with both clean and adversarial data.

2 Related Work

Alzantot et al. (2018) developed a black-box synonym substitution attack to generate adversarial

Proceedings of Deep Learning Inside Out (DeeLIO):
The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 20–28

samples for sentiment analysis. They first computed the nearest neighbors of a token based on the Euclidean distance in the embedding space. Then, they picked the token that maximizes the target label prediction when replacing the original token. Their adversarial examples successfully fooled their LSTM model's output with a 100% success rate, using the IMDB dataset (Maas et al., 2011).

Ren et al. (2019) proposed a black-box synonym substitution attack for text classification tasks. They employed word saliency to select the token to be replaced. For each token, they selected the synonym that causes the most significant change in the classification probability after replacement. They experimented with three datasets: IMDB, AG's News (Zhang et al., 2015), and Yahoo! Answers[1] using the word-level CNN of Kim (2014), the character-level CNN of Zhang et al. (2015), a Bi-directional LSTM, and an LSTM. Their results showed that, under their attack, the classification accuracies on the three datasets IMDB, AG's News, and Yahoo! Answers were reduced by an average of 81.05%, 33.62%, and 38.65% respectively.

Zhang et al. (2019) adopted the Metropolis-Hastings (M-H) sampling approach (Metropolis et al., 1953; Hastings, 1970) to generate black-box synonym substitution perturbations against text classification and textual entailment tasks. They used the M-H approach to replace targeted words with synonyms, followed by a language model to enforce the fluency of the sentence after replacing the words. Their attack successfully changed the output of their Bi-LSTM model and the Bi-DAF model (Seo et al., 2017) with 98.7% and 86.6% success rates, respectively, using the IMDB dataset, and the SNLI dataset (Bowman et al., 2015).

Jin et al. (2020) also proposed a black-box synonym substitution attack to evaluate text classification systems. They first identified important tokens for the target model, then gathered the top tokens whose cosine similarity with the selected tokens are greater than a threshold. They kept the candidates that altered the prediction of the target model. Using their attack, they evaluated the word-level CNN and a word-level LSTM, using the AG's News and IMDB datasets. Their results suggested that their attack reduced the accuracy of all target models by at least 64.2%.

[1]https://webscope.sandbox.yahoo.com/catalog.php?

3 Methodology

This paper proposes improving the generalization of DNN-based models by reducing a model's sensitivity to particular tokens in the input samples. This effectively mitigates black-box synonym substitution perturbations. We propose a method that combines word importance ranking, synonym extraction, word embedding averaging, and majority voting techniques to mitigate adversarial perturbations. Figure 1 illustrates the overall schema of the proposed approach. The proposed approach for mitigating adversarial text consists of four main steps:

- **Step 1:** Determine the N important tokens in the input sequence.
- **Step 2:** Build a synonym set for each important token.
- **Step 3:** Replace the embedding of each important token by the average of its synonyms' embeddings.
- **Step 4:** Perform a majority voting for the N replacements based on their predictions.

3.1 Scoring Function

Given a sequence of tokens, only some key tokens act as influential signals for the model's prediction. Therefore, we use a selection mechanism to choose the tokens that most significantly influence the final prediction results. We use the Replace-1 scoring function $R1S()$ of Gao et al. (2018) to score the importance of tokens in an input sequence according to the observed results from the targeted model.

By assuming the input sequence $x = x_1x_2...x_n$, where x_i is the token at the i^{th} position, we measure the effect of the x_i token on the output of the targeted model (F). The scoring function $R1S()$ measures the effect of x_i on the model by replacing x_i with x'_i. More formally:

$$R1S(x_i) = F(x_1, x_2, ..., x_{i-1}, x_i, ..., x_n) - F(x_1, x_2, ..., x_{i-1}, x'_i, ..., x_n), \quad (1)$$

where x'_i is chosen to be out-of-vocabulary (OOV) and it is obtained by inserting, deleting, or substituting a letter in x_i for a random letter. $R1S()$ measures the importance of a token by calculating the effect of replacing it with an OOV token, while observing the model's prediction. The token's importance is thus calculated as the prediction change

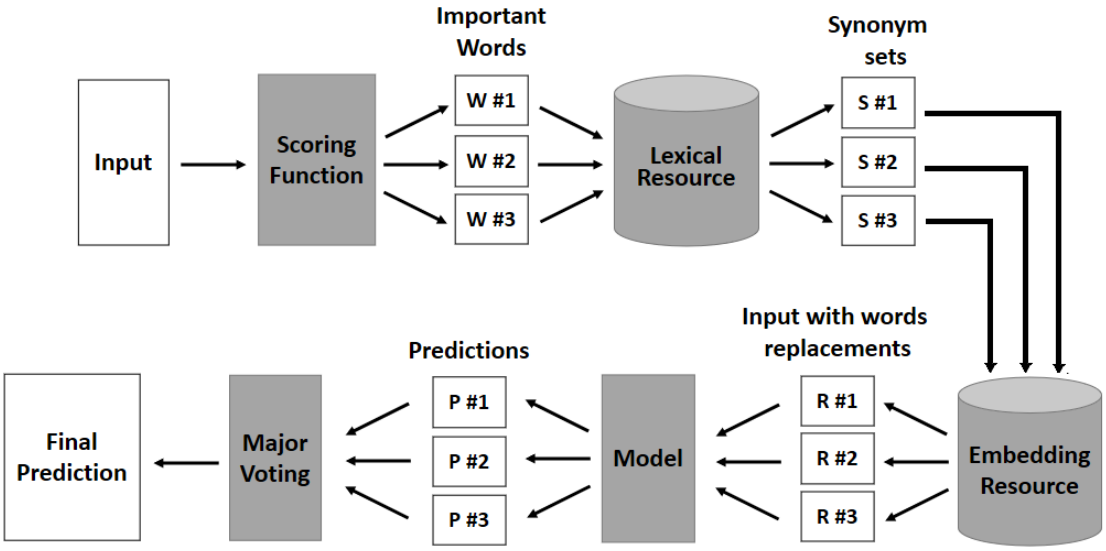

Figure 1: Schema of the proposed defensive method. The proposed defense involves the following steps: Step 1: Extract the important tokens in the input sample (here, we extract the three most important tokens). Step 2: Build a synonym set for each important token. Step 3: Replace the embedding of each important token by the average of its synonyms' embeddings. Step 4: Perform a majority voting for the replacements based on their predictions.

before and after replacing it with an OOV. By calculating the effect of replacing x_i with OOV, the importance of all tokens in the input sample can be measured and ranked. This step is employed to report the N most important tokens in an input sample. In our experiments, setting N to be 5 produces the best results.

3.2 Synonym Extraction

For a given token with a high importance score obtained in Step 1, we build a synonym set ($Synset$) for the selected token. Synonyms can be found in WordNet[2] (Miller, 1995), a large lexical resource for the English language. For each token, we use WordNet to build a synonym set that contains all possible synonyms of the token. More formally,

$$Synset(token) = \{syn_1,\ syn_2,\ ...,\ syn_m\}, \quad (2)$$

where m is the quantity of the token's synonyms that exist in the lexical resource (WordNet). If a token does not have any synonyms in the lexical resource, the processing moves to the next important token. In this step, we use WordNet as a lexical resource, but the proposed defense can use any other lexical resource (e.g. Wiktionary[3]).

[2]https://wordnet.princeton.edu/
[3]https://www.wiktionary.org/

3.3 Embedding Averaging

In the previous steps, we determine the N important tokens in an input sample (Step 1), and then extract a synonym set for each one of the important tokens (Step 2). In the third step, for each important token, we replace its embedding by the average of its synonyms' embeddings. More formally,

$$E(token) = \frac{1}{m}\sum_{i=1}^{m} E(syn_i), \quad (3)$$

where $E()$ represents the word embeddings resource, and m is the count of synonyms in the synonym set of the token.

3.4 Majority Voting

In the previous step, for each important token, we replace the embedding of the token by the average of its synonyms' embeddings. In this step, the model makes a prediction after each replacement, and assigns each replacement a vote based on its prediction. The model's final prediction will be the prediction with the majority of the votes. An example of this step is illustrated in Figure 2. In this figure, the model made three predictions and the final classification is positive, based on the votes. The proposed approach with all steps is shown in Algorithm 1.

Figure 2: Step 4: The model makes a prediction after each replacement, and assigns each replacement a vote based on its prediction. The model's final prediction is the prediction with the majority of the votes.

Algorithm 1: The overall procedure of the proposed defensive method.

input : Input sample X, classifier $F()$, Replace-1 scoring function to extract important tokens in an input sample $R1S()$, lexical resource to extract synonyms $Synset()$, word embeddings resource to represent tokens $E()$, prediction set P, majority voting method $V()$.

output : $F(X)$

$R1S(X) = \{token_1, token_2, ..., token_n\}$

for $c \leftarrow 1$ **to** n **do**

  $Synset(token_c) = \{syn_1, syn_2, ..., syn_m\}$

  $E(token_c) = \frac{1}{m}\sum_{i=1}^{m} E(syn_i)$

  $S = X$

  $S \leftarrow E(token_c)$

  $P \leftarrow F(S)$

end

$F(X) = V(P)$

Return $F(X)$

In this paper, we proposed a simple and structure-free defensive strategy which can be successful in hardening DNNs against synonym substitution based adversarial attacks. As shown in Section 5, the proposed defense yielded great performance. The advantage of our approach is that it can use any embeddings and lexical resources. It does not require any additional data to train, or modify the architecture of the models. Our implementation is generic enough to be applied in any domain and to models trained on any natural language.

4 Experiments

We implemented the proposed defensive method using Python, Numpy, Tensorflow, Scikit-learn, and Pandas libraries.

4.1 Corpus

To study the efficiency of our defense, we used the Internet Movie Database (Maas et al., 2011). IMDB is a sentiment classification dataset which involves binary labels annotating the sentiment of sentences in movie reviews. IMDB consists of 25,000 training samples and 25,000 test samples, labeled as positive or negative. The average length of samples in IMDB is 262 words.

4.2 Targeted Classification Models

To evaluate our proposed approach, several experiments on the word-level CNN model of Kim (2014) and the Bi-directional LSTM model of Ren et al. (2019) were conducted. We replicated Kim's CNN architecture, which contains three convolutional layers, a max-pooling layer, and a fully-connected layer. The Bi-directional LSTM model involves a Bi-directional LSTM layer and a fully connected layer.

4.3 Adversarial Attacks

We evaluated our defensive method with two black-box synonym substitution attacks: The attack of Alzantot et al. (2018) and the attack of Ren et al. (2019), explained in Section 2.

4.4 Word Embeddings

We used the Global Vectors for Word Representation (GloVe) embedding space (Pennington et al., 2014) to generate word vectors of 300 dimensions.

4.5 Performance Evaluation

Classification accuracy is used as the metric to evaluate the performance of the proposed defensive model. Higher accuracy denotes a more effective approach.

5 Results

The CNN and Bi-LSTM models were trained on the IMDB training set, and achieved training accuracy scores similar to the original implementations.

Model	Model w/o defense	Model w/defense	Percent Increase
CNN	76.50	80.00	3.50
Bi-LSTM	73.44	78.90	5.46

Table 1: The accuracy of the classification models on the original benign data with and without our defensive method. No adversarial perturbations were used. "w/o defense" denotes using the model with no defense. "w/defense" denotes using the model with our defense. Percent Increase is the percent increase of the classification accuracy after using the defense.

Model	Attack	Model w/o defense	Model w/defense	Percent Increase
CNN	Alzantot et al.	35.00	74.20	39.20
CNN	Ren et al.	24.60	68.00	43.40
Bi-LSTM	Alzantot et al.	23.50	72.70	49.20
Bi-LSTM	Ren et al.	5.07	67.20	62.13

Table 2: The accuracy of the classifiers under adversarial attacks, with and without the defense applied. The accuracies of the models with the original data were 76.50% and 73.44% for the CNN and the Bi-LSTM, respectively.

Following the practices of previous studies that have explored adversarial examples (Alzantot et al., 2018; Ren et al., 2019; Zhang et al., 2019; Jin et al., 2020), and because the process of generating adversarial examples to evaluate the defense is time and resource-consuming, we randomly sampled 1280 examples from the IMDB testing set to evaluate the efficiency of the proposed defensive method. As shown in Section 3, for each sample, our defensive method first extracts the five important tokens. It then extracts their synonyms from the lexical resource. Overall, there were 2.15 synonyms per important token on average, as the majority of important tokens had 2 or 3 synonyms.

We first present how the defensive method behaves on benign data with no adversarial attacks. In Table 1, we report the accuracy of the targeted models on the original test samples, with and without the defense applied. Table 1 shows that the defense is capable of improving the performance of the models even when they are not under attack. The classification accuracy of the CNN increases by 3.50%, and that for the Bi-LSTM is also increased by 5.46%. This indicates that the defense is beneficial not only in adversarial situations, but also in secure situations with no adversarial attacks.

5.1 Effectiveness of the Defense

To evaluate the efficiency of our defense in adversarial situations, we used the adversarial attacks of Alzantot et al. (2018) and Ren et al. (2019) to perturb the 1280 benign samples and convert them to adversarial examples. A more effective defensive method should cause a smaller drop in model classification accuracy when said model is under attack. Table 2 shows the efficacy of various adversarial attacks and the defensive method.

Under the adversarial attacks of Alzantot et al. and Ren et al., the classification accuracy of the models dropped significantly. For the CNN, the accuracy degraded more than 41.50% and 51.90%, under the Alzantot et al. and Ren et al. attacks, respectively. Similarly, the accuracy of the Bi-LSTM model reduced more than 49.94% and 68.37%, under the same attacks. Our results suggest that (1) DNN-based models with higher original accuracy (with clean data) are more difficult to be attacked. For instance, as shown in Tables 1 and 2, the under-attack accuracy is higher for the CNN model compared with the Bi-LSTM model under all attacks. This agrees with the observation from previous research that, in general, models with higher original accuracy have higher under-attack accuracy (Jin et al., 2020). (2) The Bi-LSTM model is more vulnerable to the two attacks than the CNN model by a 12.45% accuracy difference on average. This supports the conclusion from previous research that, in the NLP domain, deep CNNs tend to be more robust than RNN models (Ren et al., 2019; Alshemali and Kalita, 2019). (3) While Alzantot et al. randomly selected the tokens to be replaced, Ren et al. employed the word saliency technique to determine the tokens to be replaced. This makes the attack of Ren et al. more effective than the attack of Alzantot et al. on both models by an average margin of 10.40% for the CNN and 18.43% for the Bi-LSTM.

After employing our defensive method, the ro-

Model	Attack	Model w/o defense	Model w/defense	Percent Increase
SVM	No-attack	88.28	92.35	4.06
SVM	Alzantot et al.	60.00	76.10	16.10
SVM	Ren et al.	55.00	74.15	19.15
XGBoost	No-attack	85.15	89.93	4.77
XGBoost	Alzantot et al.	49.61	70.00	20.39
XGBoost	Ren et al.	40.94	66.41	25.47

Table 3: The accuracy of the nonneural classification models under adversarial attacks, with and without the defense applied. Percent Increase is the percent increase of the classification accuracy with the defense applied.

bustness of the models significantly improved under all attacks. The effectiveness of the proposed defense is evaluated under the two attacks and the results are presented in Table 2. Our results show that the proposed defense effectively mitigated most of the adversarial examples generated by the two attacks. Under the Alzantot et al. attack, the defense increased the accuracies of the models by 39.20% and 49.20% for the CNN and Bi-LSTM, respectively. Under the Ren et al. attack, the accuracies of the models were improved by an average of 43.40% and 62.13% for the CNN and Bi-LSTM, respectively. Our results highlight that (1) Under the same attack, the proposed defense performs better with the Bi-LSTM model than with the CNN by an average difference of 14.36%; and (2) Under the same model, the proposed defense performs better in mitigating Ren et al.'s adversarial examples than in mitigating the adversarial examples generated by the attack of Alzantot et al., with an average difference of 8.56%. This is likely because Ren et al. used WordNet to obtain their synonyms, while Alzantot et al. considered the nearest neighbors of a token's embedding vector as its synonyms.

5.2 Nonneural Models

In this section, we evaluated the defense using two nonneural machine learning classification algorithms, that were selected due to their high performance on a variety of text classification tasks: (1) Support Vector Machine (SVM) (Cortes and Vapnik, 1995); and (2) Extreme Gradient Boosting (XGBoost) (Chen and Guestrin, 2016). We examined the performance of our defense with the SVM and XGBoost models, trained on the IMDB dataset, and using the GloVe embedding space.

To evaluate the defense with the SVM and XGBoost models, we used the adversarial attacks of Alzantot et al. (2018) and Ren et al. (2019) to perturb the same 1280 benign samples of IMDB reviews (used in Section 5.1) and convert them to adversarial examples. Table 3 shows how the defense behaves with nonneural models on benign and adversarial data. Table 3 shows that the SVM model has more than 28.00% and 33.00% accuracy degradation under the Alzantot et al. and Ren et al. attacks, respectively. Similarly, the accuracy of the XGBoost model was reduced by 35.54% and 44.21%, under the same attacks, respectively.

By utilizing our defense, the robustness of the nonneural models improved under all attacks. Our results illustrate that the proposed defense is effectively able to mitigate most of the adversarial examples generated by the two attacks. Under the Alzantot et al. attack, the defense increased the accuracies of the models by 16.10% and 20.39% for SVM and XGBoost, respectively. Under the Ren et al. attack, the accuracies of the models were improved by 19.15% and 25.47% for SVM and XGBoost, respectively. Table 3 also shows that the defense improved the performance of the models with benign data. The classification accuracy of the SVM model increases by 4.06%, and that for the XGBoost is also increased by 4.77%.

5.3 News Categorization Task

In Sections 5.1 and 5.2, we evaluated the effectiveness of the discussed defense on the sentiment analysis task. Here, we evaluated it on the news categorization task, using the Bidirectional Encoder Representations from Transformers (BERT) embedding space and the BERT model (Devlin et al., 2019). This model was trained on the AG's News categorization dataset (Zhang et al., 2015). We used the 12-layer BERT model, also called the base-uncased version[4].

AG's News is a news categorization dataset which contains news articles categorized into four classes: World, Sports, Business and Sci/Tech.

[4]https://github.com/huggingface/transformers

Attack	Model w/o defense	Model w/defense	Percent Increase
No-attack	65.56	68.00	2.44
Alzantot et al.	35.00	58.60	23.60
Ren et al.	30.00	59.61	29.61

Table 4: The classification accuracy of the BERT model under adversarial attacks, with and without the defense applied. Percent Increase is the percent increase of the classification accuracy with the defense applied.

The total number of training samples is 120,000 and testing 7,600. The average number of words per sample is 278.6. We randomly selected 1280 samples from the AG's News testing set to evaluate the effectiveness of the proposed defensive method. We used the adversarial attacks of Alzantot et al. and Ren et al. to perturb the 1280 benign samples and convert them to adversarial examples. Table 4 shows the efficacy of the defensive method with various adversarial attacks.

Even for the powerful BERT, which has achieved great performance in various NLP tasks, adversarial attacks can still reduce its classification accuracy by about 30.56% with the attack of Alzantot et al. and by 35.56% with the attack of Ren et al.. These accuracy drops are unprecedented, however, employing our defense boosted the robustness of the BERT model under all attacks. Table 4 shows that, under the Alzantot et al. attack, the defense improved the accuracy of the model by 23.60%. Similarly, under the Ren et al. attack, the accuracy of the model was increased by 29.61%.

5.4 Statistical Analysis

While the defended classifiers had higher accuracy scores than the undefended classifiers across all tasks, adversarial attacks, and datasets, it is important to determine whether the difference in performance of the defended models is statistically significant. Many researchers recommend McNemar's test (McNemar, 1947) for comparing the performance of two classifiers (Salzberg, 1997; Dietterich, 1998; Japkowicz and Shah, 2011; Costa et al., 2018) as it has a lower probability of Type I error. McNemar's is a non-parametric pairwise test designed for comparing two populations, or in this case, the predictions from two different classifiers on the same test dataset. In this paper, McNemar's test was applied to compare the performance of the defended models with their undefended counterparts (studied in Sections 5.1, 5.2, and 5.3). Here, we wish to compare the performance of the defended CNN with the undefended CNN, the defended SVM with the undefended SVM, etc.

We performed McNemar's test to determine if there was a significant difference between the accuracy of the defended models and that of the undefended ones. We tested the null hypothesis, which states that there is no significant difference in the accuracy of the models studied, and the alternative hypothesis, which states that there is a difference in the accuracy of the models studied. Several comparisons were performed, and the significance threshold for each individual pairwise test was adjusted to 0.05. In all cases, the difference between the defended models and the undefended models (the p-value) was significant (< 0.05). Thus, we reject the null hypothesis which assumed there was no difference between the classifiers, in favor of the alternative. The results show that there was a statistically significant difference in the accuracy of all models, which indicates that the defended models had significantly better performance.

6 Conclusion

In this paper, we proposed a structure-free defensive method that is capable of improving the performance of DNN-based models with both clean and adversarial data. Our findings show that replacing the embeddings of the important words in the input samples with the average of their synonyms' embeddings can significantly improve the generalization of DNN-based models. Our results indicate that the proposed defense is not only capable of defending against adversarial attacks, but is also capable of improving the performance of DNN-based models when tested on benign data. On average, the proposed defense improved the classification accuracy of the CNN and Bi-LSTM models by 41.30% and 55.66%, respectively, when tested under adversarial attacks. Extended investigation shows that our defensive method can improve the robustness of nonneural models, achieving an average of 17.62% and 22.93% classification accuracy increase on the SVM and XGBoost models, respectively. The proposed defensive method has also

shown an average of 26.60% classification accuracy improvement when tested with the infamous BERT model. In further work, we plan to generalize our approach to achieve robustness against other types of adversarial attacks in NLP. We also hope to evaluate the defense with a variety of NLP systems, such as textual entailment systems.

References

Basemah Alshemali and Jugal Kalita. 2019. Toward mitigating adversarial texts. *International Journal of Computer Applications*, 178(50):1–7.

Basemah Alshemali and Jugal Kalita. 2020. Improving the reliability of deep neural networks in NLP: A review. *Knowledge-Based Systems*, 191(105210):1–19.

Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. 2017. Did you hear that? adversarial examples against automatic speech recognition. In *Proceedings of the 31st Conference on Neural Information Processing Systems*.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

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

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

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *International Conference on knowledge discovery and data mining*, pages 785–794. ACM.

Xie Chen, Xunying Liu, Yu Wang, Anton Ragni, Jeremy HM Wong, and Mark JF Gales. 2019. Exploiting future word contexts in neural network language models for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(9):1444–1454.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Joana Costa, Catarina Silva, Mario Antunes, and Bernardete Ribeiro. 2018. Adaptive learning models evaluation in Twitter's timelines. In *International Joint Conference on Neural Networks*, pages 1–8. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Thomas G Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for NLP. In *The Annual Meeting of the Association for Computational Linguistics*, pages 31–36.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *IEEE Security and Privacy Workshops*, pages 50–56.

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

W Keith Hastings. 1970. Monte carlo sampling methods using markov chains and their applications. *Oxford University Press*.

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. 2019. Bag of tricks for image classification with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*.

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. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Nathalie Japkowicz and Mohak Shah. 2011. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? a strong baseline for natural language attack on text classification and entailment. In *the Association for the Advancement of Artificial Intelligence*.

Nishant Kambhatla, Anahita Mansouri Bigvand, and Anoop Sarkar. 2018. Decipherment of substitution

ciphers with neural language models. In *the Conference on Empirical Methods in Natural Language Processing*, pages 869–874.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

George A Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *IEEE European Symposium, Security and Privacy*, pages 372–387.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Tommi A Pirinen. 2019. Neural and rule-based Finnish NLP models—expectations, experiments and experiences. In *the International Workshop on Computational Linguistics for Uralic Languages*, pages 104–114.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *The Annual Meeting of the Association for Computational Linguistics*, pages 1085 –1097.

Steven L Salzberg. 1997. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3):317–328.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

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

Huangzhao Zhang, Hao Zhou, Ning Miao, and Lei Li. 2019. Generating fluent adversarial examples for natural languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5569.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.