# Improving the Adversarial Robustness of NLP Models by Information Bottleneck

**Cenyuan Zhang**[1,2]*, **Xiang Zhou**[1,2]*, **Yixin Wan**[3],
**Xiaoqing Zheng**[1,2], **Kai-Wei Chang**[3], **Cho-Jui Hsieh**[3]
[1]School of Computer Science, Fudan University, Shanghai, China
[2]Shanghai Key Laboratory of Intelligent Information Processing
[3]Department of Computer Science, University of California, Los Angeles, USA
{zhouxiang20,cenyuanzhang17,zhengxq}@fudan.edu.cn
elaine1wan@g.ucla.edu, {kwchang,chohsieh}@cs.ucla.edu

## Abstract

Existing studies have demonstrated that adversarial examples can be directly attributed to the presence of non-robust features, which are highly predictive, but can be easily manipulated by adversaries to fool NLP models. In this study, we explore the feasibility of capturing task-specific robust features, while eliminating the non-robust ones by using the information bottleneck theory. Through extensive experiments, we show that the models trained with our information bottleneck-based method are able to achieve a significant improvement in robust accuracy, exceeding performances of all the previously reported defense methods while suffering almost no performance drop in clean accuracy on SST-2, AGNEWS and IMDB datasets.

## 1 Introduction

Recently, a number of studies (Han et al., 2020; Jin et al., 2020; Shafahi et al., 2019) have revealed the fact that the performance of deep neural networks (DNNs) can be severely undermined by adversarial examples. In the text domain, these adversarial examples are crafted by semantic-preserving perturbations to inputs with word synonym substitution (Ebrahimi et al., 2018; Ren et al., 2019; Alzantot et al., 2018) and character-level transformations (Gao et al., 2018; Zang et al., 2020). The vulnerability of DNN models results in inferior performances under adversarial attacks in many NLP tasks including text classification, natural language inference (NLI), question answering (QA), etc. To resolve this problem, researchers have proposed various methods to defend against adversarial attacks (Goodfellow et al., 2014; Szegedy et al., 2013; Jia and Liang, 2017; Kang et al., 2018; Zhou et al., 2021).

In particular, Tsipras et al. (2019) and Ilyas et al. (2019) showed that the vulnerability of computer vision models can be attributed to "non-robust features," which are features in the representation space that are sensitive to adversarial attacks and can be easily manipulated by attackers. The presence of these features will weaken the robustness of deep learning models. Therefore, a potential defense strategy is to filter out such non-robust features in the inputs.

In this paper, we posit that the robustness of text classification models can be improved by filtering out the non-robust features. However, unlike the continuous input in the computer vision domain, input in the NLP domain is a sequence of words, which makes it difficult to model its distribution. We thus research into means to filter out the non-robust features in language model inputs.

Inspired by (Li and Eisner, 2019; Wang et al., 2021a), we propose to use the information bottleneck method (Tishby et al., 2000) in text classification tasks. Specifically, we plug in an information bottleneck layer (IB layer)[1] between BERT (Devlin et al., 2018) output layer and the text classifier to preserve only task-specific features.

Since the IB layer trades off between minimizing preserved information and model prediction performance, features that are not robust under the targeted task are filtered out. Therefore, our approach is able to focus more on the robust features and achieve an improvement in its robustness.

We conduct extensive experiments on three text classification benchmarks: SST-2 (Socher et al., 2013), AGNEWS (Gulli, 2004) and IMDB (Maas et al., 2011). Results have shown that our approach achieves a great improvement on model robustness compared with traditional defense methods, while only suffering little or even no performance drop on clean accuracy. We also provide a visualization to interpret how the information bottleneck layer works to keep robust features, in order to justify

*Equal contribution

[1]The source codes are available at https://github.com/zhangcen456/IB.



*Findings of the Association for Computational Linguistics: ACL 2022*, pages 3588 - 3598

our proposed approach.

In summary, we propose a new information bottleneck-based approach to improve the robustness of DNN language models. We demonstrate that our approach is effective in improving models' adversarial performance while maintaining their performance on clean data. In addition, experimental results show that our approach can also be combined with existing adversarial training methods like FreeLB (Chen et al., 2020) to further improve models' robustness under adversarial attacks.

## 2 Related Work

In response to the discovery of DNN's vulnerability to adversarial examples and the emergence of adversarial attacks, many defense methods have also been proposed to improve the robustness of DNN models.

Among these methods, adversarial training is one of the most effective and widely used to defend against adversarial examples. In adversarial training, the model is trained to correctly classify both adversarial examples and normal examples. Goodfellow et al. (2014) first propose a fast gradient sign method (FGSM) to generate adversarial examples for adversarial training in the image domain. In textual domain, many researchers tried to add perturbation to input word embedding to generate adversarial examples. Zhang and Yang (2018) applied several types of noises, such as Gaussian and Bernoulli, to perturb the input embeddings while Chen et al. (2020) proposed FreeLB, which minimizes the resultant adversarial loss inside different regions around input samples through adding adversarial perturbations to word embeddings. However, they all focus on the generalization of model, not the robustness.

Wang et al. (2019) and Wang and Wang (2020) proposed to replace certain words in the training dataset with their synonyms for the purpose of data augmentation. However, this kind of methods are specific for the defense against synonym substitution attack, and may be weak when facing other kind of adversarial attack methods, such as character-level attack.

Apart from empirical methods, a set of certified robustness training methods is introduced recently, which have proven to be effective in improving a model's robustness against a specific type of attacks. Huang et al. (2019) and Jia et al. (2019) used interval bound propagation (IBP) to propose certified robustness training methods that can limit the loss of the worst-case perturbations. These methods are provably robust to word substitution attacks. However, certified robustness training sacrifices the model's clean accuracy and is not generalized to all types of attacks.

The information bottleneck method was proposed by (Tishby et al., 2000), aiming to provide a quantitative notion of "relevant information". Although the method has been utilized in many NLP tasks such as rationale extraction (Paranjape et al., 2020), sentence summarization (West et al., 2019) and parsing (Wang et al., 2020), only few of them focus on combining information bottleneck with adversarial defense methods. Wang et al. (2021b) proposes infoBERT, which applied information bottleneck to the embedding layer of pre-trained language models to suppress noisy information contained in word embeddings. The implicit assumption of this approach is that the embedding layer contains enough information for the model to make predictions. However, different word combinations can have different meanings, so the semantics of a sentence can not be fully expressed without taking contextual information into consideration.

Therefore, we propose a completely different implementation of the information bottleneck. The information bottleneck is utilized to extract task-related features from the output of the last layer of BERT, which is pre-trained and thus be capable of generating a contextualized representation for the input sequence, and calculated by using the variational inference method.

Besides, InfoBERT uses the gradient information of each word to find local anchored features and aims at increasing the mutual information between the global representation and them, while in our approach, robust features are extracted from the global representation without additional steps.

## 3 Method

Derived from information theory, the information bottleneck method (Tishby et al., 2000) was proposed and has been used as a training objective as well as a theoretical framework (Tishby and Zaslavsky, 2015) in machine learning. The method of information bottleneck can be statistically formulated as follows: denote the input random variables as $\mathbf{X}$, which could be sentences or paragraphs, and the output as $\mathbf{Y}$. Denote the joint distribution of $\mathbf{X}$ and $\mathbf{Y}$ as $P(\mathbf{X}, \mathbf{Y})$. The purpose of information

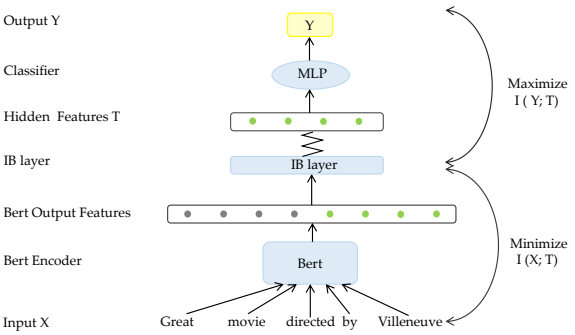


Figure 1: We use the IB layer to filter out non robust features, which are denoted by the gray circle dots in the figure, flow from Bert output. $I(\cdot;\cdot)$ denotes the mutual information and the jagged line denotes the compression process. By Maximizing the mutual information between the final prediction $\mathbf{Y}$ and the hidden features $\mathbf{T}$ while minimizing the mutual information between input $\mathbf{X}$ and $\mathbf{T}$ through IB layer, we are able to obtain a $\mathbf{T}$ that has all the non-robust features filtered out.

bottleneck is to learn a distribution $p_\theta(t|x)$ from $\mathbf{X}$ to a compressed hidden feature $\mathbf{T}$. To simplify the notation, we will omit $\theta$ in the subscript when we mention $p(t|x)$. The information bottleneck objective (IB objective) used for optimization is as follows:

$$\mathcal{L}_{\text{IB}} = -I(\mathbf{Y};\mathbf{T}) + \beta \cdot I(\mathbf{X};\mathbf{T}), \qquad (1)$$

where $I(\cdot;\cdot)$ denotes the mutual information. The intuitive explanation for optimizing the information bottleneck objective Eq.(1) is that we want to compress all information given by the input $\mathbf{X}$, while still maintaining enough knowledge for the model to give the correct prediction outcome $\mathbf{Y}$. This can be achieved through finding the minimum value of $\mathcal{L}_{\text{IB}}$. In the equation, parameter $\beta$ controls how much information we want to preserve among all the information extracted from the input $\mathbf{X}$. By increasing $\beta$, we can narrow the "neck", thus allowing less information from $\mathbf{X}$ to be transmitted to the hidden feature $\mathbf{T}$.

Inspired by (Ilyas et al., 2019)'s theory about robustness of features, we utilize the information bottleneck method to help the DNN models filter out "non-robust features" and only preserve "robust features" from model input. Since "Robust features" contribute to model's prediction, they contain semantic information of the input sequence. Taking this into account, our goal would be to filter out task-unrelated information while keeping the loss of task-related information to a minimum. This way, our method would be able to help improve model's robustness without diminishing its clean performance for the prediction task. By plugging in the IB layer right after BERT output, we can leverage the ability of pre-trained models on extracting contextualized features, preventing the possible loss of "robust features" after compression of information.

Specifically, given an input $\mathbf{X}$ and output $\mathbf{Y}$, we want to obtain specific hidden features $\mathbf{T}$ from $\mathbf{X}$ that only contain information which contributes to the final prediction $\mathbf{Y}$. By minimizing the IB objective in Eq.(1), the IB layer filters out the task-unrelated information in BERT output, which is extracted from input $\mathbf{X}$, and obtain the required $\mathbf{T}$.

To minimize the IB objective, we maximize the mutual information $I(\mathbf{Y};\mathbf{T})$. Since the purpose of maximizing $I(\mathbf{Y};\mathbf{T})$ is to enforce $\mathbf{T}$ contain enough information for the model's prediction, we choose to minimize the loss function of the original task to approximate the maximization of $I(\mathbf{Y};\mathbf{T})$. Taking classification tasks as an example, our method would be to minimize the cross entropy function $\mathcal{L}_{\text{CE}}$.

The mutual information $I(\mathbf{X};\mathbf{T})$ can be calculated by the Kullback-Leibler distance between the distributions of $P(\mathbf{T}|\mathbf{X})$ and $P(\mathbf{T})$ as follows:

$$\begin{aligned} I(\mathbf{X};\mathbf{T}) &= \mathbb{E}_X[D_{\text{KL}}[P(\mathbf{T}|\mathbf{X})||P(\mathbf{T})]] \\ &= \int p(x,t)\log\frac{p(t|x)}{p(t)}dxdt. \end{aligned} \qquad (2)$$

To calculate the Kullback-Leibler divergence between $P(\mathbf{T}|\mathbf{X})$ and $P(\mathbf{T})$, we need knowledge of their probability distributions. The $P(\mathbf{T}|\mathbf{X})$ term can be sampled empirically. However, the $P(\mathbf{T})$ term is difficult to be estimated. To resolve this challenge, we expand the Eq.2 to get the following equation:

$$\begin{aligned} I(\mathbf{X};\mathbf{T}) = &\int p(x,t)\log p(t|x)dxdt \\ &- \int p(t)\log p(t)dt, \end{aligned} \qquad (3)$$

where the marginal distribution of $\mathbf{T}$, $p(t) = \int p(t|x)p(x)dx$. Since the original Tishby et al. (2000) relied on the iterative Blahut Arimoto algorithm to opitimize the IB objective, which is infeasible to apply to deep neural networks, many researchers try to use variational inference to approximate this problem (Alemi et al., 2017; Chechik et al., 2005). Inspired by previous studies, we replace $p(t)$ with a variational approximation $q(t) =$

$\mathcal{N}(\mu_X, \sigma_X^2)$, which is a Gaussian distribution with the mean $\mu_X$ and standard deviation $\sigma_X^2$. Since the Kullback-Leibler divergence is defined to be non-negative, which means $\int p(t) \log p(t) dt \geq \int p(t) \log q(t) dt$, we derive the following upper bound:

$$I(\mathbf{X}; \mathbf{T}) \leq \int p(x) p(t|x) \log \frac{p(t|x)}{q(t)} dx dt \\ = \mathbb{E}_X[D_{\mathrm{KL}}[P(\mathbf{T}|\mathbf{X})||Q(\mathbf{T})]]. \quad (4)$$

We want to reduce the mutual information between $\mathbf{X}$ and $\mathbf{T}$ so that more task-unrelated information can be filtered out, which can help us retain more robust features for the final prediction. To achieve this goal in practice, we minimize the upper bound of $I(\mathbf{X}; \mathbf{T})$ derived in Eq.(4). We achieve this through adjusting the parameters in $Q(\mathbf{T})$ in order to minimize the Kullback-Leibler divergence between $P(\mathbf{T}|\mathbf{X})$ and $Q(\mathbf{T})$, which will lower the upper bound of $I(\mathbf{X}; \mathbf{T})$. Combined with the optimization goal of the term $I(\mathbf{Y}; \mathbf{T})$ we explained in the former chapter, the final loss function is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \beta \cdot D_{\mathrm{KL}}[P(\mathbf{T}|\mathbf{X})||Q(\mathbf{T})]. \quad (5)$$

By using the loss function Eq.(5) to optimize our model, our approach would be able to filter out the non-robust features for the classification task among all the inputs.

## 4 Experiments

In order to validate our assumption, we conduct several experiments to evaluate the effectiveness of our approach. We first compare our model and the baseline models both on their clean accuracy and accuracy under attack. Furthermore, in exploration of the ability of our model to work in conjunction with adversarial training methods- such as FreeLB- to achieve complementary effects, we also evaluate the performance of the combined model. In addition, we try to interpret and further analyze our approach through several additional experiments.

### 4.1 Dataset

We evaluate our approach on three widely-used classification benchmark datasets: IMDB dataset (Maas et al., 2011), SST-2 (Socher et al., 2013) dataset, and AGNEWS dataset (Gulli, 2004). Both IMDB and SST-2 are sentimental classification datasets with two classes, while AGNEWS is a topic classification dataset with four classes.

### 4.2 Baseline Models

Because our model can be viewed as an enhanced variant of the BERT models, we choose to use BERT base (Devlin et al., 2018) as one of the baseline models. We also establish a comparison with InfoBERT (Wang et al., 2021a), a method that is very similar to our approach, to verify the effectiveness of the proposed way of injecting an IB layer.

Apart from these two baseline models, we also compare our approach with four adversarial training methods: PGD (Madry et al., 2018), which is a classic and representative method, as well as FreeLB (Chen et al., 2020), SMART (Jiang et al., 2020) and TAVAT (Li and Qiu, 2021), which are three state-of-the-art defense methods.

### 4.3 Robustness Evaluation

We evaluate models' accuracy under four different attack algorithms, including both word-level attacks and character-level attacks.

**Textfooler** (Jin et al., 2019) Textfooler ranks the importance of words by the drop of true class probability after deleting words from the original text. By leveraging the similarity of word embeddings, it builds a candidate word set and selects the word that minimizes the predictive probability of the true class label.

**Textbugger** (Li et al., 2018) Textbugger contains both word-level and character-level perturbations by inserting, removing, swapping and substituting letters or replacing words.

**BERT-Attack** (Li et al., 2020) BERT-Attack uses the masked language model (MLM) of BERT to replace words with other words that fit the context. In addition to achieving high attack success rate, high perturbation percentage and relatively low calculation costs, BERT-Attack also ensures fluency and semanticality of adversarial samples.

**Deepwordbug** (Gao et al., 2018) Deepwordbug designs a score system to find the rank the importance of tokens to the prediction and perturb the top $k$ important tokens by swap, substitution, deletion and insertion.

### 4.4 Implementation Details

We train 10 epochs of the models on AGNEWS and SST-2 datasets, and 20 epochs on the IMDB dataset and provide experimental results averaged on three different random seeds: 0, 1, and 2.

For each attack, we take 1000 attack examples on

| Datasets | Methods | *Clean%* | TextFooler | | TextBugger | | BERT-Attack | | Deepwordbug | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | *Aua(Suc)%* | *#Query* | *Aua(Suc)%* | *#Query* | *Aua(Suc)%* | *#Query* | *Aua(Suc)%* | *#Query* |
| **SST-2** | BERT-base | 93.2 | 25.3(72.7) | 72.8 | 35.3(61.8) | 43.4 | 20.7(77.6) | 96.0 | 39.2(57.6) | 27.5 |
| | PGD | 93.5 | 27.9(70.2) | 74.6 | 37.0(60.3) | 43.6 | 22.0(76.3) | 96.6 | 40.3(56.7) | 27.2 |
| | SMART | **94.1** | 32.8(64.8) | 85.9 | 40.9(56.1) | 48.2 | 20.8(77.7) | 104.3 | 45.0(51.7) | 29.7 |
| | FreeLB | 93.9 | 29.5(68.5) | 73.4 | 40.0(57.3) | 44.6 | 23.7(74.7) | 97.0 | 42.5(54.6) | 28.0 |
| | InfoBERT | 93.9 | 31.5(66.2) | 74.1 | 40.9(56.1) | 44.4 | 25.4(72.7) | 99.4 | 42.9(53.9) | 28.3 |
| | TA-VAT | 93.6 | 34.6(62.6) | 75.4 | 43.3(53.2) | 44.4 | 26.4(71.5) | 99.5 | 45.8(50.5) | 28.0 |
| | Our Model | 93.3 | 37.6(59.9) | 104.8 | 46.5(50.3) | 61.0 | 32.9(64.9) | **147.0** | 48.4(48.0) | 34.2 |
| | + FreeLB | **94.1** | **40.4(56.8)** | **106.9** | **48.1(48.8)** | **62.7** | **33.3(64.5)** | 146.8 | **51.6(44.9)** | **35.0** |
| **AGNEWS** | BERT-base | 94.5 | 9.1(90.4) | 314.1 | 42.2(55.5) | 174.9 | 13.3(86.0) | 414.0 | 25.3(73.3) | 104.7 |
| | PGD | 94.9 | 59.0(37.6) | 261.7 | 58.8(37.8) | 287.3 | 62.7(34.0) | 264.2 | 65.7(30.6) | 254.8 |
| | SMART | 94.4 | 54.4(42.3) | 155.9 | 60.1(36.3) | 102.2 | 37.8(59.9) | 241.0 | 61.2(35.1) | 62.9 |
| | FreeLB | **94.7** | 13.6(85.7) | 343.4 | 47.5(49.8) | 175.2 | 15.9(83.2) | 435.6 | 22.3(76.4) | 106.3 |
| | InfoBERT | 93.6 | 65.0(29.8) | 173.0 | 68.0(26.6) | 106.5 | 55.0(40.7) | 261.3 | 67.0(28.0) | 180.3 |
| | TA-VAT | 94.5 | 56.7(40.1) | 264.2 | 56.3(40.5) | 290.1 | 63.1(33.5) | 270.6 | 62.3(34.2) | **260.0** |
| | Our Model | 94.2 | 68.6(27.2) | 516.4 | 70.8(24.9) | 319.9 | 60.7(35.7) | 827.2 | 70.0(26.0) | 130.8 |
| | + FreeLB | 94.4 | **70.8(25.0)** | **521.3** | **73.4(22.4)** | **326.8** | **64.0(32.4)** | **851.5** | **71.6(24.1)** | 132.0 |
| **IMDB** | BERT-base | 91.5 | 0.8(99.1) | 610.8 | 5.7(93.8) | 524.2 | 0.1(99.8) | 570.6 | 24.3(73.6) | 355.7 |
| | PGD | 92.6 | 35.7(61.4) | 1911.8 | 33.3(63.9) | 2261.9 | 36.8(60.1) | 1549.4 | 41.7(54.9) | 2082.7 |
| | SMART | **93.1** | 48.2(48.1) | 2035.4 | 53(43.2) | 1238.7 | 23.2(75.2) | 2053.1 | 58.8(36.8) | 571.8 |
| | FreeLB | 92.5 | 38.3(58.1) | 1843.1 | 49.7(45.7) | 1249.9 | 26.7(70.1) | 2453.5 | 57.5(37.1) | 550.5 |
| | InfoBERT | 91.9 | 32.2(65.3) | 1112.4 | 35.5(61.7) | 756.9 | 24.5(73.5) | 1394.3 | 44.7(51.8) | 465.0 |
| | TA-VAT | 92.5 | 38.3(58.7) | 2230.1 | 36.5(60.6) | **2864.5** | 40.3(56.5) | 1754.4 | 51.1(45.0) | **2258.17** |
| | Our Model | 91.5 | 52.2(42.4) | 2077.7 | 58.7(34.9) | 1366.0 | 38.7(57.0) | 2859.2 | 64.3(28.8) | 597.9 |
| | + FreeLB | 92.4 | **64.3(30.1)** | **2293.8** | **69.2(24.9)** | 1513.4 | **52.7(43.0)** | **3356.0** | **71.3(22.5)** | 621.8 |

Table 1: Accuracy achieved by our method and other competitive models both on clean data and under attacks. The number in bold denotes best performance on that dataset. *Clean%* denotes the prediction accuracy without attack. *Aua%* denotes accuracy under attack, *Suc%* denotes attack success rate and *#Query* denotes average query numbers. The average perturbed word percentage of all three attack methods are set to under 15%. All the attack methods used in the experiment are from the implementation of TextAttack (Morris et al., 2020). All other baseline models used in this study are based on our own implementation.

the SST-2 and AGNEWS datasets, and 200 attack examples on the IMDB dataset due to the excessive number of queries. We also set a default restriction of 15% on the maximum modify ratio for each attack algorithm. Further details for implementation would be discussed in the following sections.

### 4.5 Hyperparameter

There are two main hyperparameters in our experiments: hidden dimension (hd) and $\beta$. The hidden dimension controls the dimension of the IB layer and $\beta$ controls the trade-off between better prediction performance and restriction of the information flow. We experiment with different sizes of hd ranging from 100 to 700 and different values of $\beta$ from 0.05 to 0.3. Based on model performance, we finally choose to use hd = 100 on SST-2, AGNEWS, and hd = 200 on IMDB. We used $\beta$ = 0.1 on all three datasets. All hyperparameters are chosen based on the experimental results on the corresponding development dataset.

### 4.6 Results

Table 1 reports the detailed results of our experiment. As shown in Table 1, our model suffers little or even no performance drop in clean accuracy on all of the three datasets. On the IMDB and AGNEWS datasets, our model achieves around the same clean accuracy as the baseline model, while on the SST-2 dataset, clean accuracy of our model in fact outperforms the baseline model by a small margin. This implies that even with an added information bottleneck layer, the proposed method still ensures sufficient information flowing through the information bottleneck for the model to make accurate predictions.

Besides the clean performance, Table 1 also provides concrete experimental results on the robustness of all the models under four different types of attacks. The robustness accuracy result shows our model not just demonstrates significant improvement in adversarial robustness compared to BERT-base model, but is also very competitive with other defense methods under both word-level and character-level attacks. In particular, our model outperforms all baseline models by a great margin under the attack of TextFooler and TextBugger on IMDB dataset.

Furthermore, combining our method with FreeLB (a kind of adversarial training method) can further improve models' adversarial robustness. We achieve the highest adversarial accuracy under all circumstances in our experiments and only suffer a small drop in clean accuracy compared to the original FreeLB method. This implies that adversarial training methods and our approach are

| Datasets | Methods | *Clean%* | TextFooler | | TextBugger | | BERT-Attack | | Deepwordbug | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | *Aua(Suc)%* | *#Query* | *Aua(Suc)%* | *#Query* | *Aua(Suc)%* | *#Query* | *Aua(Suc)%* | *#Query* |
| **SST-2** | BERT-base | 93.2 | 5.6(94.0) | 89.2 | 27.9(69.8) | 47.5 | 6.2(93.3) | 111.7 | 27.4(70.3) | 29.0 |
| | PGD | 93.5 | 6.7(92.8) | 92.6 | 30.3(67.5) | 47.8 | 7.5(91.9) | 114.3 | 25.6(68.2) | 28.7 |
| | SMART | **94.1** | 12.0(87.1) | 107.8 | 33.0(64.6) | 53.7 | 8.6(90.8) | 123.3 | 35.1(62.4) | 31.2 |
| | FreeLB | 93.9 | 8.1(91.4) | 95.4 | 32.0(65.9) | 49.5 | 9.2(90.2) | 118.6 | 31.9(65.9) | 29.5 |
| | InfoBERT | 93.9 | 9.5(89.8) | 99.3 | 32.8(64.8) | 49.6 | 10.9(88.3) | 126.1 | 32.8(64.7) | 29.8 |
| | TA-VAT | 93.6 | 14.5(84.3) | 115.2 | 34.9(62.3) | 51.5 | 11.5(87.6) | 135.4 | 35.0(62.2) | 29.8 |
| | Our Model | 93.3 | 21.1(77.5) | 125.7 | 39.8(57.5) | 68.0 | 20.3(78.3) | 167.9 | 39.1(58.1) | 35.9 |
| | + FreeLB | **94.1** | **23.3(75.1)** | **129.8** | **42.7(54.4)** | **70.0** | **21.3(77.3)** | **169.0** | **41.7(55.7)** | **36.3** |

Table 2: Accuracy achieved by our method and other competitors on both clean data and adversarial examples. The average perturbed word percentage of all four attack methods are not constrained. For the DeepWordBug method, the edit distance is constrained to no more than five.

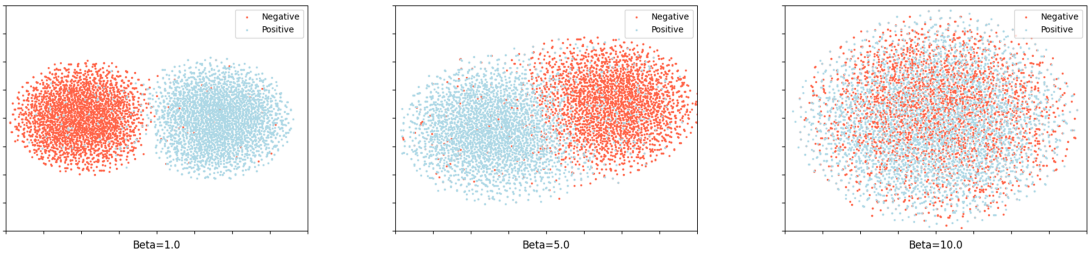


Figure 2: t-SNE visualization of our model under different $\beta$. Each marker in the figure denotes different sample. This series of figures (from left to right) shows a progression from moderate compression to too much compression. As the $\beta$ increases, the boundary between the two classes gradually disappears.

complementary to each other.

We also evaluate the accuracy of models under adversarial attack methods whose constraint is relaxed. Specifically, the maximum modify ratio is not constrained for all attack methods, which means that relatively stronger adversarial examples can be generated. The result in Table 2 shows that our method can still outperform all the baseline methods in this setting.

## 5 Discussion

In this section, we study how the implementation of IB layer affects the model's robustness. Furthermore, we seek to find a reasonable explanation for this effect.

### 5.1 Quantitative Analysis

First, we discuss the effect of the two hyperparameters in our model: hidden dimension (hd) and $\beta$. The size of the hidden dimension controls the dimension of the information bottleneck layer, thus limiting the amount of total information that flows through the information bottleneck. We test the impact of 10 different hidden dimension sizes ranging from 50 to 600 on the SST-2 dataset. As shown in Figure 3, a small hidden dimension size helps our model achieve better performance under adversarial attacks, while only suffering little drop in clean accuracy. This indicates that choosing small hidden dimension size works best in helping the model filter out task-unrelated information. However, when the size of the hidden dimension gets too small, we observe a significant drop in accuracy under attack. This may indicate that when the hidden dimension size is too small, the information bottleneck layer would compress too much such that task-related information is also filtered out.

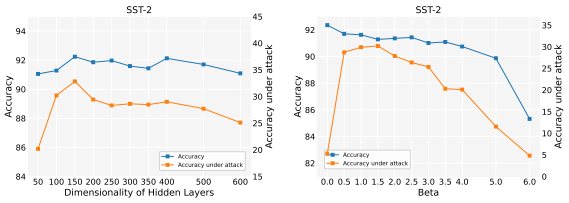


Figure 3: Performance change under different hyperparameter choices. The y-axis on the left denotes clean accuracy, while the y-axis on the right denotes accuracy under attack, using TextFooler as the attack method. The x-axis denotes different hyperparameters for our IB layer, from left to right, shows the performance of our model under different dimensionality of hidden layers and $\beta$. We fixed the hidden dimension of IB layer at 100 when choosing different values of $\beta$.

We also test the influence of different values of

$\beta$s from 0 to 6. Recall that the hyperparameter $\beta$ controls the trade off between better prediction performance and restriction of the information flow through the bottleneck. Therefore, adjusting the value of $\beta$ could also help with controlling the amount of information that flows through the information bottleneck layer. Specifically, a smaller $\beta$ will allow more information to flow from $\mathbf{X}$ to hidden representation $\mathbf{T}$, while a larger $\beta$ would "narrow" the bottleneck, allowing less information to flow through. When $\beta$ is set to 0, the information bottleneck layer is equal to a linear layer. Results of experiments with different values of $\beta$ on the SST-2 dataset are shown in Figure 3. At first, as the value of $\beta$ increases, less information are able to flow through the bottleneck, forcing the information bottleneck to filter out non-robust features. This results in an enhancement of model robustness performance. However, the robust accuracy decrease as $\beta$ increase further. This might indicates that if the information bottleneck is too "narrow", some robust features would also be filtered out by the IB layer.

We further visualize the influence of $\beta$ by using t-SNE. As shown in Figure 2, when $\beta$ is equal to 1.0, the samples can be clearly divided into two clusters corresponding to their labels. As the value of $\beta$ increases, more information are filtered out, including those useful for the sentiment classification task. It can be observed from the figure that samples from the two categories become close to each other when the value of $\beta$ is 5.0. Therefore, a small perturbation may cause the model to make a false prediction. The decision boundary becomes unclear when $\beta$ is set to 10.0 and the clean accuracy of the model decreases significantly in this case.

## 5.2 Interpretation

As we have discussed in former sections, researchers such as Tsipras et al. and Ilyas et al. have proposed the idea that features contributing to deep learning tasks can be divided into robust features and non-robust features. In our assumption, the information bottleneck layer that we plug in after BERT output works to filter out non-robust features while retaining the robust features from all the information that flows out from BERT. By only preserving task-specific robust features, our model is able to attain an improvement in adversarial robustness, while minimizing the drop in clean accuracy performance at the same time.

| Methods | Attack | | Attack (filtered) | |
|---|---|---|---|---|
| | Sig% | Acc% | Sig% | Acc% |
| Baseline | −7.9 | 2.7 | −3.6 | 7.3 |
| Our model | 4.6 | 37.1 | 23.7 | 92.0 |

Table 3: Accuracy achieved by our model and the baseline under attacks. $Sig\%$ denotes the sum of significance scores of words that are consistent with the whole sentence's sentiment tendency. $Acc\%$ denotes classification accuracy. "Attack" denotes the adversarial examples generated by the TextFooler algorithm, from which we choose two hundred sentences, indicated by "Attack (filter)".

### 5.2.1 Significance Score

In order to verify our assumption, we specifically conduct an experiment on the SST-2 dataset to see if our model is able to capture robust task-related features. In this experiment, a score $s_i$ is calculated for each word $x_i$ in the input sentence $X$ to measure the influence of $x_i$ when predicting the sentiment of the sentence. We denote the embedding of the input sentence $X$ as $E = \{e_1, ..., e_n\}$, and $X_{\backslash x_i} = \{e_1, ..., e_{i-1}, 0, e_{i+1}, ..., e_n\}$ denotes setting the embedding of word $x_i$ to zero. The normalized significance score $s_i$ is defined as follows.

$$s_i = \text{Rescale}(F_Y(X) - F_Y(X_{\backslash x_i})) \quad (6)$$

where $F_Y(X)$ denotes the probability that our model gives a prediction of the label $Y$, and $\text{Rescale}(\cdot)$ is defined by dividing the significance score by the sum of the absolute values of significance scores of all words in the sentence. Here, $s_i$ denotes the change in models' predictions before and after deleting word $x_i$. A positive value indicates that the word helps the model make correct predictions, and a negative value means that the word leads to incorrect predictions of the model.

We note that the SST-2 dataset provides sentiment labels for each word which are annotated manually. In order to identify sentiment words in sentences, we make use of these labels provided as the golden truth. Since the words that are consistent with the sentimental tendency of the sentence are important for sentiment classification, we sum the significance scores of these words and denote this sum as $Sig\%$.

As shown in Table 3, the baseline model is not able to correctly classify most sentences under textual adversarial attacks on the SST-2 dataset. Note that here $Sig\%$ is a negative value, meaning that the

words that the baseline model considers "harmful" to making classification predictions in fact have a counter-effect on the model. This means that the baseline model fails to correctly attention to important features in the input sentences. By definition, $Sig\%$ increases when the models' classification accuracy improves, indicating their correlation. Since our model achieves a higher $Sig\%$, it can be inferred that our model is better at extracting the robust features that are not likely to be perturbed under textual adversarial attacks.

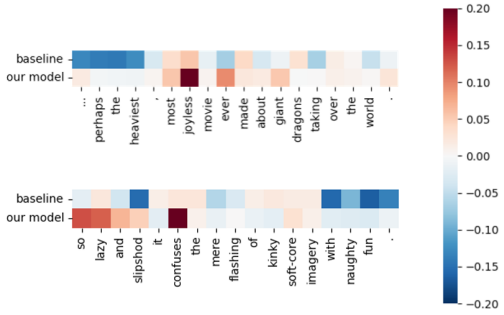


Figure 4: Illustration of each word's significance in the model's prediction process.

In order to better illustrate how our model succeeds in extracting robust features, specifically for classification tasks, we visualize two examples from the SST-2 dataset in Figure 4. In the first sentence, our model better attends to the word "joyless", which clearly expresses the emotional tendency in the sentence. In contrast, the attention of the baseline model is distracted by words such as "perhaps" and "the", which are almost irrelevant to the sentimental classification task. In the second sentence, "slipshod" is a sentiment word which is helpful for predicting the sentiment of the sentence. As demonstrated by the figure, our model succeeds in accurately capturing the importance of this sentiment word. The baseline model, however, failed to attend to this word and thus is unable to classify the sentence correctly. The figure shows that our model accurately captures robust features related to important sentimental words in inputs, while the baseline model fails to do the same thing. This further validates the assumption that our model is able to extract robust task-related features for classification.

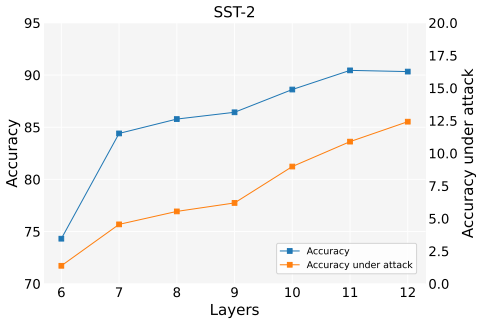


Figure 5: The performance change of our model when we add the Information Bottleneck layer after different layers of BERT model.The dimensionality of hidden layer and $\beta$ is fixed at 100 and 1.0, respectively.

#### 5.2.2 Plugging Information Bottleneck into Different Layers

Features in the deep layer model have been observed to transit from general to task-specific as the position of the layer gets higher (Yosinski et al., 2014; Howard and Ruder, 2018). More specifically, the first layers of deep neural networks may contain more general and static knowledge of the input sequences, while the higher layers contain more knowledge related to the task. Since the information bottleneck layer is used to extract robust, task-specific features, we assume that applying it to higher layers instead of the embedding layer, which is used in InfoBERT, would be a more reasonable implementation. To verify this, we add the Information Bottleneck layer to the output of different layers of BERT model and calculate an additional loss(Eq.(5)) as regularization. We train the models to minimize the total loss. Experimental results show that as the layer we choose to apply the information bottleneck layer becomes higher, model's classification accuracy and accuracy under attack both improve, which validates our assumption.

## 6 Conclusion

We propose a novel implementation of the information bottleneck method on BERT-base models to improve its adversarial robustness. Our method is proven to be successful in filtering out non-robust feature and keeping task-specific robust features, thus improving the adversarial robustness of models. Experimental results have shown that our method outperforms four widely used defense methods across three datasets with both sentence-level and character-level attack algorithms. We also validate our method through a comprehensive anal-

ysis on experimental results as well as quantitative interpretation of our model's performance under adversarial attacks.

## Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by National Science Foundation of China (No. 62076068), Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0103), and Zhangjiang Lab. Chang is supported in part by Cisco and Google. Hsieh is supported in part by NSF IIS-2008173 and IIS-2048280.

## References

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Gal Chechik, Amir Globerson, Naftali Tishby, and Yair Weiss. 2005. Information bottleneck for gaussian variables. *J. Mach. Learn. Res.*, 6:165–188.

Zhu Chen, Gan Yu, Cheng adn Zhe, Sun Siqi, Goldstein Tom, and Liu. Jingjing. 2020. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*, pages 50–56. IEEE Computer Society.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv 1412.6572*.

Antonio Gulli. 2004. Agnews. http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

Wenjuan Han, Liwen Zhang, Yong Jiang, and Kewei Tu. 2020. Adversarial attack and defense of structured prediction models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2327–2338, Online. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4081–4091. Association for Computational Linguistics.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. *arXiv:1905.02175*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142, Hong Kong, China. Association for Computational Linguistics.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational*

*Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

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

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 *AAAI (AAAI-20)*.

Dongyeop Kang, Tushar Khot, Ashish Sabharwal, and Eduard Hovy. 2018. AdvEntuRe: Adversarial training for textual entailment with knowledge-guided examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2418–2428, Melbourne, Australia. Association for Computational Linguistics.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *CoRR*, abs/1812.05271.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. pages 6193–6202.

Linyang Li and Xipeng Qiu. 2021. Token-aware virtual adversarial training in natural language understanding. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 8410–8418. AAAI Press.

Xiang Li and Jason Eisner. 2019. Specializing word embeddings (for parsing) by information bottleneck (extended abstract). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing.*, pages 4687–4691.

Andrew Maas, Raymond Daly, Peter Pham, Dan Huang, Andrew Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.*, pages 142–150.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.*

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1938–1952, Online. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. 2019. Are adversarial examples inevitable? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Richard Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, and C. Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, volume 1631, pages 1631–1642.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Naftali Tishby, Fernando C. Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv:physics/0004057*.

Naftali Tishby and Noga Zaslavsky. 2015. Deep Learning and the Information Bottleneck Principle. *arXiv:1503.02406*.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*.

Boxin Wang, Hengzhi Pei, boyuan Pan, Qian Chen, Shuohang Wang, and Bo Li. 2020. T3: Tree-autoencoder constrained adversarial text generation for targeted attack. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021a. Infobert: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021b. INFOBERT: Improving Robustness Of Language Models From An Information Theoretic Perspective. In *ICLR (ICLR-21)*.

Xiaosen Wang, Hao Jin, and Kun He. 2019. Natural language adversarial attacks and defenses in word level. *CoRR*, abs/1909.06723.

Zhaoyang Wang and Hongtao Wang. 2020. Defense of word-level adversarial attacks via random substitution encoding. In *Knowledge Science, Engineering and Management - 13th International Conference, KSEM 2020, Hangzhou, China, August 28-30, 2020, Proceedings, Part II*, volume 12275 of *Lecture Notes in Computer Science*, pages 312–324. Springer.

Peter West, Ari Holtzman, Jan Buys, and Yejin Choi. 2019. BottleSum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3752–3761, Hong Kong, China. Association for Computational Linguistics.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *NIPS*.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6066–6080. Association for Computational Linguistics.

Dongxu Zhang and Zhichao Yang. 2018. Word embedding perturbation for sentence classification. *CoRR*, abs/1804.08166.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2021. Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5482–5492, Online. Association for Computational Linguistics.